EMPRA: Embedding Perturbation Rank Attack against Neural Ranking Models

AMIN BIGDELI, University of Waterloo, Canada
NEGAR ARABZADEH, University of California, Berkeley, USA
EBRAHIM BAGHERI, University of Toronto, Canada
CHARLES L. A. CLARKE, University of Waterloo, Canada

Recent research has shown that neural information retrieval techniques may be susceptible to adversarial attacks. Adversarial attacks seek to manipulate the ranking of documents, with the intention of exposing users to targeted content. In this paper, we introduce the *Embedding Perturbation Rank Attack* (EMPRA) method, a novel approach designed to perform adversarial attacks on black-box Neural Ranking Models (NRMs). EMPRA manipulates sentence-level embeddings, guiding them towards pertinent context related to the query while preserving semantic integrity. This process generates adversarial texts that seamlessly integrate with the original content and remain imperceptible to humans. Our extensive evaluation conducted on the widely-used MS MARCO V1 passage collection as well as the TREC DL 2019 and TREC DL 2020 benchmarks, demonstrate the effectiveness of EMPRA against a wide range of state-of-the-art baselines in promoting a specific set of target documents within a given ranked results. Specifically, on MS MARCO Dev set queries, EMPRA successfully achieves a re-ranking of almost 96% of target documents originally ranked between 51-100 to rank within the top 10. Furthermore, EMPRA does not rely on surrogate models for generating adversarial documents, enhancing its robustness against various victim NRMs in realistic settings.

CCS Concepts: • **Information systems → Retrieval models and ranking**; **Adversarial retrieval**.

Additional Key Words and Phrases: Information Retrieval, Neural Ranking Models, Adversarial Attacks, Black-box Attacks, Embedding Perturbation, Robustness Evaluation

ACM Reference Format:
Amin Bigdeli, Negar Arabzadeh, Ebrahim Bagheri, and Charles L. A. Clarke. 2026. EMPRA: Embedding Perturbation Rank Attack against Neural Ranking Models. 1, 1 (February 2026), 41 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Despite significant advancements in Neural Ranking Models (NRMs), recent research highlights vulnerabilities and a possible lack of resilience to adversarial attacks and perturbations, within both queries and documents [54, 76, 77]. These attacks are crafted to either elevate or diminish the ranking of a target document, thereby amplifying or reducing the likelihood of users encountering the information it contains. Consequently, the presence of such attacks and the fragility of neural information retrieval systems may negatively impact the integrity and dependability of the results. In particular, adversarial documents operationalize a practical threat model in which an attacker can strategically manipulate information exposure by steering users toward targeted documents, suppressing competing content, or

Authors' Contact Information: Amin Bigdeli, University of Waterloo, Waterloo, ON, Canada, abigdeli@uwaterloo.ca; Negar Arabzadeh, University of California, Berkeley, Berkeley, California, USA, negara@berkeley.edu; Ebrahim Bagheri, University of Toronto, Toronto, ON, Canada, ebrahim.bagheri@utoronto.ca; Charles L. A. Clarke, University of Waterloo, Waterloo, ON, Canada, claclark@gmail.com.

Manuscript submitted to ACM

amplifying misleading and low-quality results. This raises fundamental concerns regarding ranking integrity and user trust, making document-injection attacks a critical object of study. We argue that the systematic creation and evaluation of such adversarial documents are essential for identifying the limits of neural rankers robustness, as uncovering high-impact and imperceptible vulnerabilities is a prerequisite for building the next generation of robust neural ranking models.

In the early days of the web search, adversarial attacks might take the form of term spamming, wherein query-related terms were repetitively inserted into a target document to enhance its ranking in the retrieved results [7, 28, 63]. These attacks were undertaken with the aim of engaging in black-hat Search Engine Optimization (SEO), wherein specific documents are targeted and their content manipulated to secure higher rankings in search results. This manipulation sought to increase the visibility of the content, exposing it to a larger audience [27]. However, given their relative simplicity, such term spamming tactics were susceptible to detection by spam filters [12, 80]. In contrast, more recent research in this space has been inspired by broader work in adversarial attacks on deep neural networks, which are often designed to manipulate the classification outcomes of these models [19, 46, 66, 81]. Most notably, various authors have focused on assessing the resilience of neural-based ranking models against adversarial attacks, including word substitution rank attacks, trigger generation-based attacks and prompt-based attacks [9, 39, 44, 71, 77]. This work is based on the fact that neural ranking models learn the semantic mapping between the query and document during the training process. As such, adding/replacing terms and sentences that are semantically similar to the original text and are capable of deceiving the model can enhance the ranking position of the perturbed document.

It is noteworthy that unlike term spamming techniques, these attack strategies can subtly manipulate document content in ways that are more imperceptible to both humans and machines, rendering them challenging to detect. For example, Chen et al. [9] propose a method that generates a pool of connection sentences by prompting a generative Language Model (LM) with a target pair of document and query. Then the most effective sentence — which promotes the ranking of the target document while maintaining coherence within the original document text — is selected and injected into the target document to increase its chance of exposure. In other work, Wu et al. [77] introduced a word-level substitution method for attacking neural ranking models. Their proposed method pinpoints the key tokens within a document through the gradient of a neural ranking model which are then substituted with their nearest neighbors, selectively enhancing the document's ranking if the substitution proves beneficial.

While these state-of-the-art methods have established a strong foundation and achieve significant attack success rate, many current methods encounter three key challenges, namely: (1) In the context of an adversarial attack, the target neural ranking model that the attacker seeks to mislead is referred to as the victim model. To manipulate its outputs, existing methods typically utilize a surrogate model, which is a proxy model specifically trained to mimic the behavior of the victim model. Specifically, these approaches depend heavily on surrogate models during adversarial text generation, a process that requires a substantial amount of in-distribution training data obtained by querying the victim model. As a result, the attacking method often exhibit a lack of robustness and a significant drop in attack performance metrics when models trained on easily accessible out-of-domain data are employed for crafting the adversarial documents. Moreover, this dependency not only reduces robustness across various victim models, but also limits their practicality in real-world scenarios where querying the victim model is not feasible or accessible. (2) They can generate adversarial documents that exhibit grammatical errors, nonsensical word sequences, and incoherent text fragments, rendering a considerable portion perceptible to both humans and machines that the document has been manipulated. (3) They can compromise content fidelity to achieve ranking boosts. Approaches that rely on token substitution or generative rewriting can inadvertently alter the document's core semantic meaning or introduce logical inconsistencies. This sacrifice of content integrity undermines the ultimate objective of the attack even if the document achieves a high rank.

In response to these challenges, we present the Embedding Perturbation Rank Attack (EMPRA) method, which is a surrogate-agnostic method designed to execute adversarial black-box attacks on NRMs. EMPRA strategically manipulates *sentence-level* embeddings to enhance the ranking of specific target documents. For sentence-level perturbations, EMPRA iteratively operates on the embedding representation of a document's sentences. This iterative process involves two key functions: (1) a *transporter function*, which shifts sentence representations closer to the query context, and (2) a *transformer function*, which converts the perturbed embedding representations into lexical form. The objective is to guide the sentences embeddings towards the context of the query while maintaining certain constraints that prevent substantial semantic deviation from the original sentences. After a set number of iterations, EMPRA generates adversarial text that not only encapsulate information from the original document's sentences but also exhibit semantic proximity to query-related information. Followed by this, the generated adversarial text is injected into the original version of the target document while preserving coherence and relevance, thus culminating in the production of a final adversarial document that is imperceptible to humans and machines. Unlike many baselines, EMPRA generates adversarial documents without relying on any specific surrogate NRM. This independence from surrogate model selection enhances its robustness, enabling it to adapt seamlessly to both In-Distribution (ID) and Out-of-Distribution (OOD) scenarios while ensuring appropriate attack performance against a diverse range of victim models.

To evaluate the efficacy of EMPRA, we conduct experiments utilizing the MS MARCO V1 passage collection [48] as well as the TREC DL 2019 [14] and TREC DL 2020 [13] benchmarks used by all prior works to attack NRMs. When targeting documents from the ranked list for diverse queries, EMPRA consistently outperforms state-of-the-art baselines, notably improving the ranking positions of target documents. Specifically, on MS MARCO Dev set queries, EMPRA outperforms the baselines by re-ranking almost 96% of attacked documents that originally ranked 51-100 and 65% of documents that ranked 996-1000 into the top 10, with comparable performance gains observed across TREC DL benchmarks. Furthermore, our experimental findings highlight the robustness of EMPRA across various victim NRMs and datasets, underscoring its performance reliability in real-world scenario attacks. Notably, EMPRA demonstrates an ability to generate documents of high grammatical quality that remain imperceptible to human observations and automated detection systems.

The key contributions of our work can be enumerated as follows:

- **C1:** We propose a black-box adversarial attack method against neural ranking models that applies embedding perturbations on sentences within documents to generate adversarial documents that can outperform state-of-the-art baselines in terms of attack performance.
- **C2:** We report extensive attack experiments across three datasets demonstrating that our method effectively ranks the adversarial documents in high positions. Since it is surrogate-agnostic and does not rely on surrogate models for adversarial document generation, our attack method is the most robust and effective against various victim models.
- **C3:** We demonstrate that EMPRA generates low-perplexity and fluent adversarial documents that can remain imperceptible under both human and automatic evaluations.

2 Related Work

In this section, we first provide an overview of adversarial attack research across various other domains and then introduce existing adversarial attack work specifically designed for text ranking models.

2.1 Adversarial Attacks Across other Domains

Since the emergence of Deep Neural Network (DNN) models, the research community has extensively studied their robustness to adversarial attacks across various fields such as computer vision [1, 2, 45, 65], recommender systems [16, 20, 36, 73, 75], and natural language processing [19, 25, 35, 46, 56, 66, 81].

Within the field of computer vision, adversarial attacks typically involve perturbing the representation of image inputs in imperceptible ways with the objective of misleading the prediction output of the victim model. These attacks have been explored in depth, and significant studies have highlighted the vulnerabilities of image classifiers to such adversarial attacks [1, 2, 45, 65]. Various attacks such as the Fast Gradient Sign Method (FGSM) [24] have demonstrated how gradient-based perturbations can drastically manipulate a model's classification output and significantly reduce its accuracy. Building on FGSM, Carlini and Wagner [6] introduced the C&W attack, which employs an optimization framework that generates adversarial examples by minimizing perceptual perturbations while ensuring misclassification. This attacking strategy effectively exposes the limitations of defense mechanisms like defensive distillation [52] and often succeeds where simpler methods like FGSM fail by focusing on reducing the perceptual impact of the perturbation while still achieving the desired attack outcome. In addition, there are works that go beyond classification and focus on adversarial attacks on object detection systems and face recognition [2]. For instance, adversarial patch attacks introduce visible perturbations strategically placed within an image to mislead the object detection systems in identifying objects [5]. Similarly, adversarial attacks on face recognition systems aim to alter facial features imperceptibly in a way that misleads the model into making incorrect identifications [64].

In recommender systems, adversarial attacks exploit vulnerabilities by introducing fake user interaction data such as fake reviews, manipulated ratings, or fake user profiles to manipulate recommendation results of the system. These manipulations are often carried out through data poisoning attacks, where the goal of adversaries is to inject malicious data to promote or demote specific items within the recommendation system output [16, 20]. For instance, Chen et al. [8] proposed a novel framework that leverages a knowledge graph to integrate auxiliary item attribute knowledge into a hierarchical policy network to enable the generation of high-quality fake user profiles designed to manipulate the recommendation outputs of the system effectively. Fan et al. [21] proposed a black-box attack framework that leverages transferable cross-domain user profiles from a source for crafting adversarial attack on the target domain through a locally trained surrogate model.

In natural language processing, adversarial attacks can be categorized based on the granularity of the perturbations. These attacks include character-level, word-level, sentence-level, and multi-level approaches, with techniques ranging from simple character swaps and word substitutions to more complex paraphrasing or generative methods [25, 56]. For instance, HotFlip [19] introduced a white-box gradient-based character-level attack that identifies optimal perturbations in text by leveraging model gradients to craft minimal yet highly effective adversarial examples. TextFooler [29] employs word-level substitutions using semantically similar and grammatically correct replacements to mislead NLP models output while preserving the original input's fluency and coherence. BERT-Attack [35] is a contextualized adversarial attack method that leverages the masked language modeling capabilities of BERT [17] to craft adversarial examples. It identifies and replaces vulnerable words with semantically consistent and grammatically correct substitutions and is capable of effectively misleading the victim model's prediction outputs while maintaining input fluency and coherence. It is important to note that these attack methods are specifically designed for NLP tasks such as sentiment analysis and text classification, where the output consists of classification scores. In contrast, our work and related studies focus on adversarial attacks targeting text ranking models, which fundamentally rely on ranking relevance criteria between queries and documents, resulting in different outputs.

At a high level, adversarial attacks can be categorized into three types:

- White-box Attacks: The attacker has full access to the training data, model architecture, hyperparameters, and other details of the victim model [19, 22]. This type of attack is primarily used for research and development to build resilient models and is not practical in real-world scenarios, as attackers typically do not have extensive access to model settings.
- Gray-box Attacks: The attacker has partial access to the model's information, such as user reviews and ratings on platforms like Amazon [23, 72]. This makes gray-box attacks more practical and dangerous than white-box attacks.
- Black-box Attacks: In many real-world contexts, the attack happens under a black-box scenario where the attacker has no information about the target model's settings or internal details. They can typically only query the target model to refine their attack strategies [21, 67]. This type of attack is the most realistic scenario for attackers targeting the neural network models used in commercial contexts.

2.2 Adversarial Attacks on Text Ranking Models

The advancement of neural models in information retrieval has led to a paradigm shift from traditional term-frequency-based approaches, such as BM25 [61], which rely on exact term matching between queries and documents, to neural models capable of capturing the semantic relationships between queries and documents. For instance, various studies have shown that fine-tuning transformer-based models such as BERT [17] for information retrieval tasks significantly outperforms traditional methods [38, 50, 51, 55]. This is primarily due to their ability to capture the semantic context of words within a query and document, along with their capacity to learn relevance by constructing an embedding space that brings the representations of queries and their relevant documents closer together while increasing the distance to irrelevant documents. Despite their impressive retrieval effectiveness, the core language models underlying these neural-based systems are vulnerable to adversarial attacks [29, 35]. These vulnerabilities can be exploited in different ways within the context of text ranking models, allowing attackers to manipulate the rankings by promoting or demoting specific documents in the list presented to users.

Recently, there has been growing attention towards assessing the robustness of the neural-based retrieval systems against adversarial attacks, particularly in the context of black-hat SEO and web spamming attacks [27, 53]. Adversarial attacks in the search domain aim to manipulate a target document to deceive the model into ranking the perturbed document higher, thereby increasing its exposure to user search queries [7]. These attacks pose significant threats to the fairness and integrity of search engines by exploiting vulnerabilities in ranking algorithms to undermine relevance criteria, distort fair ranking practices, and promote biased or misleading content.

Adversarial attack methods in search systems can be broadly categorized based on the type of target model: (1) attacks targeting retrieval models [41, 68, 79] and (2) attacks targeting Neural Ranking Models (NRMs) [9, 39, 44, 71, 77]. Among these, adversarial attacks targeting NRMs are of particular interest and form the central focus of this work. Black-box adversarial attacks on NRMs can be further classified into three categories: (1) word-level-based attacks

[59, 71, 77], (2) trigger-based generation attacks [39, 71], and (3) prompt-based attacks [9, 44]. Furthermore, these methods can be characterized as either surrogate-aware or surrogate-agnostic, depending on whether they require feedback from a surrogate NRM during the adversarial text generation stage. We provide a comprehensive discussion on the differences between these levels of attacker's background knowledge in Section 3.2.

(1) Word-level-based attacks target semantically important words in the document and replace them with semantically similar words that have closer representations to query within an embedding space. For instance, Wu et al. [77] proposed PRADA attack method, which uses a surrogate model to detect important words within the target document and employ a greedy approach to replace those words with their nearest neighbors within the embedding perturbation space. EMPRA distinguishes itself by being a surrogate-agnostic attack method when generating adversarial documents and not being dependent on a specific model trained on either in-distribution or out-of-distribution data. As a result, it offers greater flexibility and adaptability in various attack settings.

(2) Trigger-based attacks aim to craft a short text and insert it into the document. Methods like Pairwise Anchor-based Trigger (PAT) [39] add several trigger tokens at the beginning of the document using a ranking incentive objective equipped with fluency and semantic constraints to craft the adversarial document. In a similar study, Wang et al. [71] proposed Brittle-BERT, which uses HotFlip [19], a gradient-based attacking technique, to add or replace tokens inside the target document to promote its ranking. Both PAT [39] and Brittle-BERT [71] use a surrogate model for generating adversaries that are injected into the target documents. EMPRA not only eliminates the need for a surrogate model but also generates adversarial texts that are semantically related to both the target query context and the original target document. This makes the final adversarial documents imperceptible to humans and machines by avoiding the use of irrelevant triggers and maintain the natural flow and coherence of the document. Besides, by incorporating semantically similar sentences related to both the target query context and the target original document, EMPRA demonstrates robustness against various victim models.

(3) Prompt-based attacks leverage the generative capabilities of Generative Language Models (GLMs), including encoder-decoder architectures such as BART [33] and Large Language Models (LLMs), to produce adversarial content. These approaches utilize prompting strategies to either generate specific segments for injection or modify the original document to enhance a target document's ranking for a specific query. For instance, Chen et al. [9] proposed IDEM, which utilizes a surrogate-agnostic attack strategy that prompts BART [33], a generative LM, to generate connection sentences for the target document and query pair. In the adversarial selection stage, by applying relevance and fluency constraints using a neural ranker, which can be either a surrogate or a generic NRM, and an LM, they inject the highest-scoring connection sentence into the document.

Additional prompt-based methods can constitute off-the-shelf attackers that rely solely on direct prompting of LLMs, which we refer to as LLM-Prompt, to generate an adversarial document given a target document and the search query. Such methods are not intended to be novel attack methods, but rather as a reference point to quantify the extent to which zero-shot generation capabilities of LLMs can manipulate ranking outcomes in the absence of targeted optimization. Inspired by strategies used to craft adversarial documents in related generative settings, such as the *Rewriter* attack in [3], these approaches enable us to evaluate the inherent vulnerability of NRMs to general-purpose generative models. To incorporate NRM feedback into the LLM generation process, Liu et al. [44] proposed AttChain, a method that leverages LLMs with chain-of-thought prompting to iteratively modify parts of the target document through multiple rounds of selecting higher-ranked documents as anchor documents. Under practical attack conditions, the generation process is guided by NRM feedback, requiring a surrogate model to evaluate the perturbed document at each iteration and to determine the higher-ranked anchor documents that serve as guidance nodes within the reasoning

Table 1. Taxonomy of adversarial attacks on Neural Ranking Models (NRMs). We adopt these methods in our evaluation, which comprehensively covers diverse perturbation units and mechanisms, spanning from classical gradient-guided optimization to state-of-the-art Zero-shot and Chain-of-Thought prompting strategies.

Method	Unit	Action	Mechanism	Surrogate Dependency
PRADA [77]	Word	Substitution	Gradient-guided Greedy Substitution	Surrogate-Aware
PAT [39]	Trigger	Injection	Ranking-Incentivized Trigger Search	Surrogate-Aware
Brittle-BERT [71]	Trigger	Injection	Gradient-based Trigger Optimization	Surrogate-Aware
IDEM [9]	Sentence	Injection	Zero-shot Prompting (GLM)	Surrogate-Agnostic
LLM-Prompt [3]	Document	Rewrite	Zero-shot Prompting (LLM)	Surrogate-Agnostic
AttChain [44]	Word	Iterative Rewrite	Chain-of-Thought Prompting (LLM)	Surrogate-Aware
EMPRA (Ours)	Sentence	Injection	Iterative Embedding Perturbation	Surrogate-Agnostic

chain. While effective, this approach introduces a risk of logical consistency drift, where the core facts or logic of the original document may be inadvertently altered or lost during the process.

Our proposed approach, EMPRA, distinguishes itself from IDEM [9] and other established baselines [71, 77] by expanding the scope of adversarial context beyond the target query to include information from the top-ranked document. Furthermore, EMPRA maintains two primary technical distinctions from AttChain [44]. First, EMPRA eliminates the model dependency inherent in AttChain, which relies on a surrogate NRM in realistic attack settings to provide iterative feedback and to identify specific higher-ranked anchor documents. Instead, EMPRA efficiently utilizes the readily accessible top-ranked document as a guidance anchor. Second, while AttChain relies on generative rewriting that risks introducing logical consistency drift or modifying the original document's facts and logic, EMPRA preserves content integrity by strategically injecting targeted adversarial text into the original version of the document. This approach exploits the embedding space to introduce potent relevance signals without altering the foundational facts or original document's core structure. To provide a clear overview of the current research landscape, we summarize the characteristics of the state-of-the-art adversarial ranking attacks and our proposed method in Table 1.

Various studies leverage these categories of attack across different domains and contexts. For instance, Liu et al. [42] introduce a framework that employs reinforcement learning with attacking actions drawn from [77] and [39], enabling the agent to perturb documents, elevating the target document's visibility for a set of semantically similar queries. A similar study [43] presents a framework utilizing reinforcement learning to orchestrate a diverse set of existing attacking methods, employing GPT-4 output fluency as the reward function at every state to craft adversarial documents. While these RL-based methods achieve slightly better attack performance compared to individual attack methods, they are significantly more time-consuming due to the complexity of RL and the computational demands of large language models. This may make them challenging to scale for practical real-world applications. Due to the lack of publicly available source code and insufficient implementation details in [43] for independent reimplementation, we instead incorporated their most recent method, AttChain [44], described earlier, as a more practical and effective baseline.

3 Threat Model

3.1 Attack Objective

Let $D_q^{\mathcal{M}_V} = [d_1, d_2, ..., d_m]$ represent a list of m ranked documents for a query q from a collection of documents C by a *victim* neural ranking model $\mathcal{M}_V$, which is targeted by the attack. These documents are ordered according to the relevance scores assigned by the victim neural ranking model to each query-document pair, denoted as $f_{rel}(q, d_j)$, where

j ranges from 1 to m, and it holds that $f_{rel}(q, d_j) < f_{rel}(q, d_{j-1})$. The attacker's objective is to design an adversarial threat model that applies perturbations p to a target document d within $D_q^{\mathcal{M}_V}$ to create an adversarial document d^{adv}. The adversarial document d^{adv} succeeds in the attack objective if the degree of perturbations $||p||$ applied to the target document results in a higher score with respect to the query, thus achieving a better (lower) ranking position, i.e.:

$$\text{Rank}(q, d^{\text{adv}}) < \text{Rank}(q, d), \tag{1}$$

where $\text{Rank}(q, d^{\text{adv}})$ and $\text{Rank}(q, d)$ denote the positions of the adversarial document d^{adv} and the target document d in the ranking for the query q, respectively. Additionally, the core content similarity between the original document d and the adversarial document d^{adv} must meet a minimum threshold to prevent from semantic drift after perturbation and make sure d^{adv} preserve the core content of the target document. The core content similarity threshold can be defined as:

$$CoreSim(d, d^{\text{adv}}) \geqslant \lambda, \tag{2}$$

where the core similarity function evaluates the key semantic elements to ensure that the core content intended to be exposed to the user remains aligned with the original content even after manipulation.

It is also important to make the perturbations in a manner that remains imperceptible to both humans and machines, while successfully deceiving the victim NRM. In particular, adding nonsensical or irrelevant phrases that degrade the readability of the adversarial document undermines the attack objective by decreasing user confidence and trust. Furthermore, the final adversarial document must maintain high logical consistency to ensure that the perturbations do not introduce factual contradictions or nonsensical transitions. Preserving this logical integrity is vital for the attack's utility, as it ensures that the document's original facts and intended information remain intact and preserved while its visibility is enhanced.

3.2 Attacker's Background Knowledge

To align with real-world scenarios and consistent with prior studies on adversarial attacks against neural ranking models [9, 39, 77], our work focuses on the black-box adversarial attack setting. In this framework, the attack strategy employed by the attacker to craft the perturbed document d^{adv} lacks access to any information regarding the victim neural ranking model, including its hyperparameters, gradients, and training data. As a result, the attacker can only query the victim neural ranking model $\mathcal{M}_V$ with a set of queries and use its output for constructing adversarial documents.

To guide the attack process in a black-box setting, the attacker can adopt one of two strategies, categorized as *surrogate-aware* or *surrogate-agnostic* attack methods:

- **Surrogate-Aware Methods:** Surrogate-aware methods [39, 44, 71, 77] are characterized by their reliance on guidance from a neural ranking model (NRM), distinct from the victim, to direct the adversarial generation process. These methods often involve training a surrogate model, denoted as $\mathcal{M}_S$, using in-domain training data derived from pseudo-relevance labels generated by querying the victim model and collecting its ranked output for a predefined set of queries. By learning to approximate the ranking criteria of the victim model, surrogate models enable the attacker to generate adversarial documents that exploit the victim model's vulnerabilities. This approach is practical only when sufficient query access to the victim model is available, allowing for a close approximation of its behavior. However, surrogate-aware methods require substantial computational resources for training, and the process must be repeated if the victim model changes, making it costly and time-consuming. Furthermore, the high volume of queries required to generate training data significantly increases the risk of exposure to detection systems. Critically, this dependency creates a single point of failure where the attack fails if the surrogate model, or the guiding NRM feedback loop, does not perfectly align with the victim due to differences in architecture, training data, or frequent updates, necessitating regular and costly retraining to maintain alignment. These disadvantages necessitate the development of attack methods that do not rely on training in-domain surrogate models during the adversarial generation process.

- **Surrogate-Agnostic Methods:** Surrogate-agnostic methods [3, 9] do not require explicit training on in-domain pseudo-relevance labels or extensive querying of the victim model. Instead, they leverage a generic neural ranking model, denoted as $\mathcal{M}_G$, trained on publicly available datasets to provide broad relevance signals rather than model-specific guidance. Critically, these methods bypass the need for a surrogate model trained specifically for the adversarial text generation process; instead, they utilize the generic NRM as a selection or scoring mechanism to inject effective adversarial text into the document. Unlike surrogate-aware methods, surrogate-agnostic methods operate independently of the target victim model and offer a practical alternative in scenarios where access to the victim model is restricted or infeasible. This independence not only reduces the computational overhead but also enhances the adaptability and robustness of the attack. By using models that are not coupled to a specific victim, these methods achieve broader applicability across diverse architectures without requiring realignment when the victim model is updated. Consequently, surrogate-agnostic approaches are more resilient to model mismatch while remaining significantly more stealthy and cost-efficient by eliminating the need for iterative victim interaction during the generation process.

4 Proposed Method

In this section, we introduce our proposed black-box adversarial attack method, referred to as `EMPRA`, a surrogate-agnostic attack strategy designed to manipulate the content of a target document with the goal of deceiving the victim neural ranking model into ranking the adversarial document higher in the rank list. `EMPRA` consists of two key stages: (1) *Adversarial Text Generation*; and, (2) *Adversarial Document Construction*. These stages integrate sentence-level embedding perturbations with strategic document construction to achieve an effective and imperceptible adversarial attack.

The first stage focuses on adversarial text generation, where the goal is to generate adversaries through a transition from discrete lexical modifications to more continuous adjustments in the embedding space. `EMPRA` leverages a transporter function to iteratively perturb the embedding of each sentence in the target document, moving it closer to an anchor text that is highly relevant to the query. The transformer function then maps the perturbed embedding back into lexical form. This iterative process generates a set of adversarial texts that are capable of boosting the ranking position of the target document and successfully manipulate the victim model.

The second stage involves constructing the adversarial document by strategically integrating the generated adversarial texts into the original document. Each adversarial text is evaluated for semantic coherence, measured using a Next Sentence Prediction (NSP) function, and query-document relevance, assessed through a generic neural ranking model. The final adversarial document is selected based on an interpolated scoring mechanism that balances coherence and relevance, ensuring that the document is both highly effective at deceiving the victim model and indistinguishable from natural text.

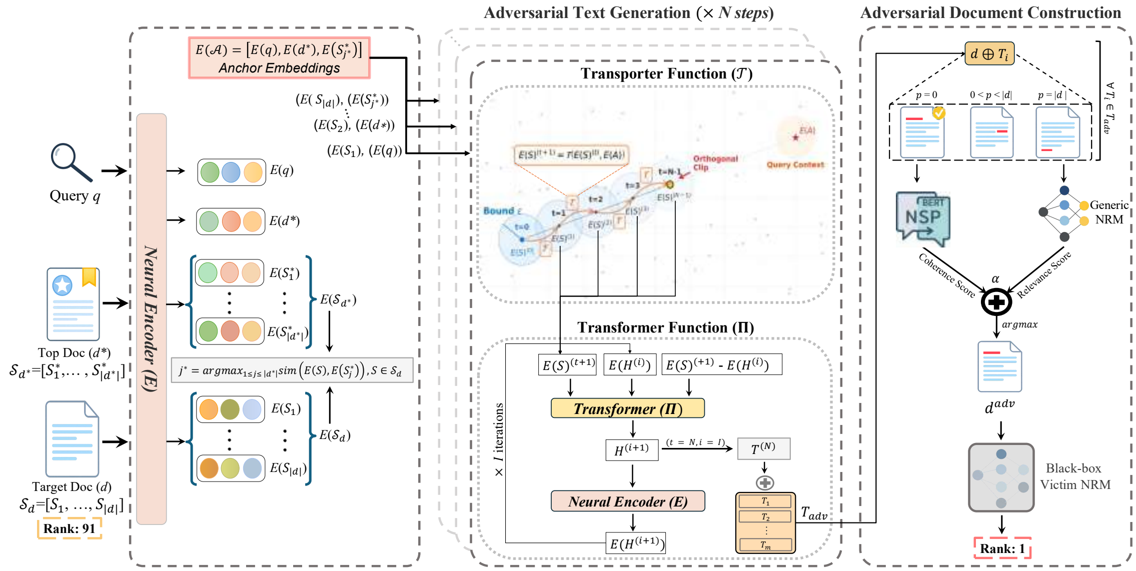

Fig. 1. Overview of EMPRA method. The process begins with a Neural Encoder E mapping the query, top document, and target document into a high-dimensional embedding space. These representations are used to define a collection of anchor embeddings $\mathcal{A}$, which include the query $E(q)$, the top-ranked document $E(d^*)$, and the most contextually similar sentence $E(S^*_{j^*})$ selected from d^* for each target sentence $S \in \mathcal{S}_d$. Adversarial Text Generation is then performed via a Transporter ($\mathcal{T}$), which optimizes embeddings toward these anchors over N steps, followed by a Transformer (Π), which iteratively decodes these vectors into fluent text over I iterations. Finally, in the Adversarial Document Construction stage, every generated adversarial text $T_i \in T_{\text{adv}}$ is evaluated across all potential insertion positions. The optimal adversarial document is then selected by maximizing a joint objective of NRM relevance and semantic coherence, measured via BERT's Next Sentence Prediction (NSP) scores.

The overall workflow of our proposed attack method, EMPRA, is illustrated in Figure 1. In addition, Algorithm 1 provides detailed pseudo-code that outlines the procedure for generating adversarial documents in the black-box attack setting.

4.1 Adversarial Text Generation

The adversarial text generation process is the core of our proposed attack method, with the goal of finding a set of text segments that, when appended to the target document, can boost its ranking in the rank list. Let d represent a target document from $D_q^{\mathcal{M}_V}$, which our proposed attacking model $\aleph$ would like to manipulate so that it achieves a higher ranking for query q. Let d^* denote the document currently ranked highest for query q. Also, let $\mathcal{S}_d = [S_1, S_2, ..., S_{|d|}]$ represent d as a sequence of sentences where $|d|$ is the total number of sentences in d. Similarly, let $\mathcal{S}_{d^*} = [S_1^*, S_2^*, \ldots, S^*_{|d^*|}]$ represent the sequence of sentences within the top-ranked document d^*. Also, let $\mathcal{A} = [A_1, A_2, ..., A_k]$ represent a collection of *anchor texts* that provide pertinent context related to q. These anchor texts can be defined as: 1) the query itself, 2) the top-ranked document d^*, or 3) the most similar sentence from the top-ranked document d^* to the target corresponding sentence in d. In order to generate a set of adversarial text that boost the ranking of d, our proposed attacking model $\aleph$ considers $\mathcal{S}_d$ and $\mathcal{A}$ to generate adversarial texts for d as follows:

$$T_{\text{adv}} = \aleph(\mathcal{S}_d, \mathcal{A}), \tag{3}$$

Algorithm 1: EMPRA

Input: Query q; Target Document d and its sentences $\mathcal{S}_d = [S_1, \dots, S_{|d|}]$; Anchor Texts $\mathcal{A} = [A_1, \dots, A_k]$; Embedding Function $E(\cdot)$; Transporter Function $\mathcal{T}(\cdot,\cdot)$; Transformer Function $\Pi(\cdot)$; Generic NRM $\mathcal{M}_G$;
Parameter: Step size η; Perturbation bound ϵ; Max Iterations N; Interpolation Coefficient α;
Output: Adversarial Document d^{adv}

Stage 1: Adversarial Text Generation
Initialize $T_{\text{adv}} \leftarrow []$
foreach $S \in \mathcal{S}_d$ **do**
  foreach $A \in \mathcal{A}$ **do**
    Initialize $T^{(0)} \leftarrow S$
    Initialize $E(S)^{(0)} \leftarrow E(S)$
    for $t = 0 \dots N-1$ **do**
      $E(S)^{(t+1)} = \mathcal{T}(E(S)^{(t)}, E(A))$
      $T^{(t+1)} \leftarrow \Pi\big(E(S)^{(t+1)}\big)$
    $T_{\text{adv}} \leftarrow T_{\text{adv}} \cup \{T^{(N)}\}$

Stage 2: Adversarial Document Construction
Initialize $d^{\text{adv}}_{\text{best}} \leftarrow d$, $bestScore \leftarrow -\infty$.
foreach $T_i \in T_{adv}$ **do**
  for $p = 0 \dots |d|$ **do**
    if $p = 0$ **then**
      $d^{\text{adv}}_{i,p} \leftarrow T_i \oplus d$
    else
      if $p = |d|$ **then**
        $d^{\text{adv}}_{i,p} \leftarrow d \oplus T_i$
      else
        $d^{\text{adv}}_{i,p} \leftarrow d_1^p \oplus T_i \oplus d_{p+1}^{|d|}$
    $C_{\text{coh}} \leftarrow$ Evaluate via $f_{\text{nsp}}(q, d^{\text{adv}}_{i,p})$
    $C_{\text{rel}} \leftarrow$ Calculate relevance score using $\mathcal{M}_G$ via $f_{\text{rel}}(q, d^{\text{adv}}_{i,p})$
    $Score_{\text{interp}} \leftarrow \alpha \cdot C_{\text{coh}} + (1-\alpha) \cdot C_{\text{rel}}$
    if $Score_{interp} > bestScore$ **then**
      $bestScore \leftarrow Score_{\text{interp}}$
      $d^{\text{adv}}_{\text{best}} \leftarrow d^{\text{adv}}_{i,p}$

Output: $d^{\text{adv}} \leftarrow d^{\text{adv}}_{\text{best}}$.

where $T_{\text{adv}} = [T_1, T_2, ..., T_m]$ consists of a set of adversarial text generated by $\aleph$. We note that the changes in the lexical form are limited and discrete, which increases the likelihood of deviating significantly from the original content. In contrast, working in the embedding space allows for more continuous adjustments, providing greater flexibility for slight perturbations. This enables a better balance between maintaining relevance and preserving semantic meaning. Therefore, instead of operating in the lexical space, we transition to the embedding space to achieve this balance. As such, in order to generate adversarial texts, the attacking model $\aleph$ leverages two components that work together in tandem, namely **(1)** a *transporter function* $\mathcal{T}(.)$; and, **(2)** a *transformer function* $\Pi(.)$.

Let $E(.)$ be the embedding function that maps target document sentences $\mathcal{S}_d$ and anchor texts $\mathcal{A}$ to their corresponding fixed-length vector representation within a high-dimensional embedding space. We define the specific anchor embeddings utilized in this process as the vector representations of the query $E(q)$, the top-ranked document $E(d^*)$, and the most contextually similar sentence $E(S^*_{j^*})$ from d^* relative to the target sentence $S \in \mathcal{S}_d$. The optimal index j^* for this sentence alignment is determined by maximizing the cosine similarity:

$$j^* = \underset{1 \le j \le |d^*|}{\operatorname{argmax}} \operatorname{sim}(E(S), E(S^*_j)), \quad \text{for } S \in \mathcal{S}_d. \tag{4}$$

The goal of the transporter function $\mathcal{T}$ is to manipulate the target sentence embedding representation $E(S)$ within the embedding space to align it more closely with a target anchor vector representation $E(A)$, where $A \in \mathcal{A}$. By anchoring these perturbations to semantically meaningful references, namely $E(q)$, $E(d^*)$, or the matched sentence embedding $E(S^*_{j^*})$, the transporter function encourages the updated representations to remain in a linguistically plausible region of the embedding space. The transformer function Π is then responsible to transform the perturbed embedding representation to lexical form. The adversarial attack process involves iteratively adjusting the embedding representation of the document sentences to converge towards the desired target anchor texts, thereby enhancing the similarity score between the sentences and anchor texts.

Given the sentence embedding $E(S)$ and the anchor embedding $E(A)$, the transporter function calculates the new coordinates of the sentence embedding representation through $E(S)^{(t+1)} = \mathcal{T}(E(S)^{(t)}, E(A))$, where t represents the iteration step in the adversarial text generation process. The transformer function then maps the embedding representation to its corresponding lexical form, defined as $T^{(t+1)} = \Pi(E(S)^{(t+1)})$. Each iteration of our approach can be defined as the following two steps:

$$\begin{aligned} T^{(t+1)} &= \Pi\left(E(S)^{(t)} + \eta \cdot \operatorname{clip}\left(\frac{\partial}{\partial \mathrm{S}}\left(\frac{E(S)^{(t)} \cdot E(A)}{\left\|E(S)^{(t)}\right\|_2 \left\|E(A)\right\|_2}\right), -\epsilon, \epsilon\right)\right), \\ t &= \{0, 1, 2, \ldots, N-1\}. \end{aligned} \tag{5}$$

Here, η denotes the step size of embedding perturbation and clip ensures the perturbation is within the specified ϵ bounds i.e., to retain the information of the original sentence. Such constraints are imposed on the search space to ensure that the representation of the perturbed embedding $E(S)$ does not deviate significantly from its original embedding state $E(S)^{(0)}$. This is achieved by limiting the magnitude of perturbations to ensure they fall within an L_∞ distance of the original representation with a specified radius of ϵ. After N iterations, the adversarial textual representation of the original sentence S that is closer to the anchor text but still close to the original representation of the sentence is obtained.

The transformer function Π is responsible for mapping the perturbed embedding representation $E(S)^{(t+1)}$ into a discrete lexical form $T^{(t+1)}$. This function iteratively enhances a textual hypothesis $H^{(i)}$, which is an intermediate version of a sentence generated during the attack process. For this purpose, at each iteration, the goal is to diminish the divergence between its associated embedding $E(H^{(i)})$ and a desired embedding target $E(S)^{(t+1)}$ during the transformation. In practice, Π performs conditioned lexical reconstruction to decode a text sequence whose embedding matches the target vector $E(S)^{(t+1)}$ while remaining close to the current hypothesis $H^{(i)}$ [34, 47]. To ensure high-fidelity reconstruction, Π leverages a conditioned encoder-decoder architecture that treats the transformation as an iterative refinement process. The decoding process conditions generation on (i) the target embedding, (ii) the embedding of the current hypothesis, and (iii) their residual error signal, so that each refinement step explicitly corrects the mismatch between $E(H^{(i)})$ and $E(S)^{(t+1)}$. These signals serve as a conditioned prefix for the decoder, providing the necessary context to guide the generation process. Specifically, the decoder's hidden state $\mathbf{h}_j$ is computed by attending to this multi-faceted context, ensuring that token generation is precisely steered by the target embedding, the current hypothesis, and their corresponding error residual. Consequently, the generation of the next token x_j is modeled as a conditional probability over the vocabulary $\mathcal{V}$:

$$P(x_j \mid x_{<j}, E(S)^{(t+1)}, E(H^{(i)}), E(S)^{(t+1)} - E(H^{(i)})) = \text{softmax}(\mathbf{W}_o \mathbf{h}_j), \tag{6}$$

where $\mathbf{h}_j$ is the decoder's hidden state and $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}| \times d_{\text{hid}}}$ is the output weight matrix. To navigate the discrete search space, Π employs a beam search heuristic to identify the sequence of tokens that best satisfies the conditional probability given the target embedding. The optimal textual hypothesis $H^{(i+1)}$ at each step is determined by maximizing the cumulative log-probability over the space of valid sequences $\mathcal{V}^*$:

$$H^{(i+1)} = \arg \max_{H \in \text{Beam}(\mathcal{V}^*)} \sum_{j=1}^{|H|} \log P(x_j \mid x_{<j}, E(S)^{(t+1)}, E(H^{(i)}), E(S)^{(t+1)} - E(H^{(i)})). \tag{7}$$

This refinement process continues for I iterations, and the final hypothesis $H^{(I)}$ is taken as the adversarial text segment for that step, that is, $T^{(t+1)} = H^{(I)}$.

Having the transporter and transformer functions working in tandem, for every sentence $S \in \mathcal{S}_d$ and every anchor $A \in \mathcal{A}$, the attack model can generate adversaries as follows:

$$T^{(t+1)} = \Pi\big(\mathcal{T}(E(S)^{(t)}, E(A))\big), \quad t = \{0, 1, 2, \ldots, N-1\}. \tag{8}$$

Where $T^{(0)} = S$. This iterative process generates a comprehensive set of adversaries T_{adv} for document d. Importantly, this generation phase is conducted entirely independently of any specific surrogate model. This decoupled architecture ensures that the candidate generation process remains agnostic to the eventual choice of victim neural ranking model used for evaluation.

4.2 Adversarial Document Construction

To construct the final adversarial document d^{adv}, we adopt a candidate selection strategy inspired by prior studies [9, 39]. Our proposed attacking model $\aleph$ injects each of the generated adversarial text $T_i \in T_{\text{adv}}$ into different positions within the target document d to construct the adversarial document d^{adv}. This approach identifies the most effective adversarial text in enhancing both the relevance score and fluency of the target document. This balance is crucial for maintaining the fidelity of the perturbed document and achieving an optimal trade-off between its effectiveness at tricking the victim neural ranking model $\mathcal{M}_V$ and imperceptibility to human judges and automated systems. To this end, the insertion operation $\mathcal{I}(d, T, p)$ places each $T_i \in T_{\text{adv}}$ at position p of the target document to build different candidates for d^{adv} as follows:

$$d_{i,p}^{\text{adv}} = \begin{cases} T_i \oplus d & \text{if } p = 0 \\ d_1^p \oplus T_i \oplus d_{p+1}^{|d|} & \text{if } 0 < p < |d| \\ d \oplus T_i & \text{if } p = |d| \end{cases} \tag{9}$$

where $\oplus$ denotes concatenation, and d_a^b represents the sub-sequence of sentences from S_a to S_b. Given different variations of adversarial candidates, the effectiveness and coherence of each candidate $d_{i,p}^{\text{adv}}$ are quantified by two principal metrics, namely 1) *semantic coherence*; and, 2) *relevance to the query*. In order to evaluate semantic coherency, a coherence score C_{coh} is calculate using pre-trained BERT Next Sentence Prediction (NSP) function denoted as f_{nsp} [18].

This function assesses the compatibility between adjacent document sentences. For an adversarial sentence T_i inserted at position p, the coherence score is defined as follows:

$$C_{\text{coh}}(d_{i,p}^{\text{adv}}) = \begin{cases} f_{\text{nsp}}(T_i, d) & \text{if } p = 0, \\ f_{\text{nsp}}(d, T_i) & \text{if } p = |d|, \\ \frac{1}{2}\left[f_{\text{nsp}}(d_1^p, T_i \oplus d_{p+1}^{|d|}) + f_{\text{nsp}}(d_1^p \oplus T_i, d_{p+1}^{|d|})\right] & \text{if } 0 < p < |d|. \end{cases} \tag{10}$$

To calculate the relevance score between each adversarial candidate and the query in the black-box setting, our proposed surrogate-agnostic attacking model utilizes the relevance scoring function of a generic neural ranking model M_G, as follows:

$$C_{\text{rel}}(q, d_{i,p}^{\text{adv}}) = f_{\text{rel}}(q, d_{i,p}^{\text{adv}}). \tag{11}$$

The generic model can be any neural ranking model other than the victim model itself, trained on in-distribution or out-of-distribution data to learn the relevance criteria. It is worth mentioning that this makes the process of constructing the adversarial document ranker-agnostic, as it does not necessarily depend on a specific surrogate model. This design choice ensures greater generalizability and flexibility of the proposed attack method across different ranking systems.

In order to select the best adversarial candidate, an interpolated score $Score_{\text{interp}}$ is computed to balance the trade-off between semantic coherence and query-document relevance as shown below:

$$Score_{\text{interp}}(q, d_{i,p}^{\text{adv}}) = \alpha \cdot C_{\text{coh}}(d_{i,p}^{\text{adv}}) + (1 - \alpha) \cdot C_{\text{rel}}(q, d_{i,p}^{\text{adv}}), \tag{12}$$

where α is the interpolation coefficient, and both C_{coh} and C_{rel} are normalized to be within the range of [0, 1]. The adversarial document d^{adv} for the target document d would be the candidate with the highest $Score_{\text{interp}}$, ensuring a balanced approach that maximizes attack efficacy while maintaining semantic coherence, thereby reducing the risk of detection.

5 Experimental Setup

To facilitate reproducibility, we make our code and experimental data publicly available at https://github.com/aminbigdeli/EMPRA.

5.1 Datasets

5.1.1 Benchmark Datasets. Similar to previous studies [9, 39, 44, 71, 77], we utilize the MS MARCO V1 Passage Collection [48], which encompasses 8.8 million passages. This collection includes over 500,000 training queries, a small validation set (dev small) with 6,980 queries, and a small eval set with 6,837 queries. An adequate number of training, validation, and test queries make this dataset suitable for training both victim and surrogate NRMs, as well as for evaluating the performance of attack methods. To further investigate the generalizability of our proposed attack method, we also conduct our experiments on the TREC DL 2019 [14] and TREC DL 2020 [13] benchmarks, each comprising 200 queries.

Additionally, a processed version of the Natural Questions (NQ) dataset [31] as prepared by [30], the SQuAD2.0 dataset [58], the Stanford Natural Language Inference (SNLI) dataset [4], and the Community QA (CommonQA) dataset [49] were used to serve as out-of-distribution training datasets for training generic NRMs. This allowed us to explore the stability and robustness of `EMPRA` and compare it against various attacking methods.

5.1.2 Target Queries and Documents. To evaluate the performance of the attack strategies, we follow the approach of Chen et al. [9] for selecting target query-document pairs. For this purpose, we randomly selected 1,000 queries from the MS MARCO Dev set, hereafter referred to as MS MARCO Dev, and utilized the complete set of 200 test queries for both TREC DL 2019 and TREC DL 2020. For each query, 10 documents were targeted from the re-ranked list generated by the target victim model from the top-1000 documents initially retrieved by BM25. These documents were classified into 'Easy-5' and 'Hard-5' groups based on the anticipated difficulty of boosting their rankings into the top-10 or top-50 ranking positions.

The Easy-5 documents were randomly chosen from each 10-document segment within positions 51-100. The Hard-5 documents comprised of the last five ranked documents by the victim model, occupying positions 996 to 1,000. As a result, each attacking method was tasked with producing adversarial documents for a total of 10,000 documents for MS MARCO Dev and 2,000 documents for each TREC DL dataset. Additionally, in line with prior work [9, 39, 44], we include a set of '*Mixture*' target documents from MS MARCO Dev for comprehensive analysis. This set comprises 32 target documents sampled from both Easy and Hard categories to ensure a balanced representation of varying difficulty levels. The Mixture target documents are specifically utilized for human-based evaluations and metrics requiring high computational costs, enabling an efficient and comprehensive analysis of the attack methods.

5.2 Evaluation Metrics

5.2.1 Attack Performance. Similar to previous studies [9, 39, 44], we consider a set of comprehensive attack performance metrics each capturing unique aspects of the effectiveness of our proposed adversarial method and baselines on document rankings.

Attack Success Rate (ASR). This metric evaluates the effectiveness of an attack by calculating the proportion of targeted documents that achieve a higher ranking post-attack compared to their original position, averaged across all queries:

$$ASR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{N_q} \sum_{i=1}^{N_q} \mathbb{1}\left(\text{Rank}_{\mathcal{R}_q}\left(d_i^{\text{adv}}\right) < \text{Rank}_{\mathcal{R}_q}(d_i)\right).$$

Here, Q is the set of evaluated queries, N_q represents the number of targeted documents for a specific query q, and $\mathbb{1}(\cdot)$ is the indicator function, which returns 1 if the condition inside is true, and 0 otherwise. A higher attack success rate, as measured by the proportion of adversarial documents achieving improved rankings, indicates a more effective attack strategy.

Boosted top-k. This metrics, represented as $\%r \leq k$, calculates the proportion of adversarial documents d^{adv}, originally ranked outside the top-k, that are moved into the top-k positions post-attack, averaged over all queries:

$$\%r_{\leq k} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{N_q} \sum_{i=1}^{N_q} \mathbb{1}\left(\text{Rank}_{\mathcal{R}_q}(d_i) > k \,\wedge\, \text{Rank}_{\mathcal{R}_q}\left(d_i^{\text{adv}}\right) \leq k\right).$$

This metric highlights the attack's capacity to significantly alter the visibility of lower-ranked documents. We report the performance of various attack methods using $\%r \leq 10$ and $\%r \leq 50$ to measure the percentage of adversarial documents boosted into the top-10 and top-50 positions, respectively.

Average boosted ranks (Boost). This metric measures the average improvement in rankings for adversarial documents d^{adv}, averaged across all queries:

$$Boost = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{N_q}\sum_{i=1}^{N_q}\left(Rank_{\mathcal{R}_q}(d_i) - Rank_{\mathcal{R}_q}(d_i^{\text{adv}})\right).$$

It reflects how effectively the attack method can elevate the ranking positions of target documents and highlights its impact on improving their visibility within the search results.

To assess whether the attack performance improvements achieved by EMPRA over the baselines are statistically significant, we performed a two-tailed paired t-test with a significance level of $p < 0.05$. To this end, we computed per-query metrics by averaging the results of the target documents within each query. These per-query measurements were then used to perform paired tests between EMPRA and the baseline that showed the highest performance for each attack metric. This approach confirms that the observed improvements are consistent across the distribution of attacked queries.

5.2.2 Content Fidelity Metrics. To evaluate the extent to which adversarial documents retain the informational integrity of the source original document, we report three fidelity measures covering lexical, semantic, and logical dimensions.
Lexical Overlap Recall (LOR): The average ROUGE-L Recall [37] between the original document and its generated adversarial version. This metric quantifies the strict preservation of the original lexical sequence, serving as a proxy for content retention within the adversarial document.
Semantic Similarity (SS): The average BERTScore F1 [78] computed utilizing the `roberta-large` model. This metric assesses the contextual similarity between the adversarial and original documents in high-dimensional embedding space. It captures general semantic alignment which may remain high even if logical coherence is compromised.
Logical Consistency (LC): To address the limitations of embedding-based similarity in capturing logical contradictions, we measure logical consistency using the average backward entailment rate [32, 40]. For this purpose, we employ the `microsoft/deberta-large-mnli` model, treating the adversarial document as the premise and the original document as the hypothesis. This metric strictly verifies the degree to which the original information remains logically supported by the adversarial document, effectively penalizing factual incoherence that standard embedding-based metrics often overlook.

5.2.3 Quality and Naturalness. To fully evaluate the quality and naturalness of the adversarial documents generated by our attacking method and baselines, we employed six different metrics, following the approach of previous work [9, 39, 44]. Among these metrics, Perplexity and Readability were measured at scale across all generated documents for all attacking strategies. Due to computational costs and the limited availability of human annotators,the remaining metrics were evaluated exclusively on the 'Mixture' target documents.
Perplexity. To evaluate the fluency of the generated adversarial documents, we utilize a pre-trained GPT-2 model [57] to compute the language model perplexity. Perplexity is defined as:

$$\text{Perplexity}(d^{\text{adv}}) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|w_1, w_2, \ldots, w_{i-1})\right),$$

where N is the total number of words in the adversarial document d^{adv}. A lower perplexity value indicates higher fluency, as it implies that the language model assigns higher probabilities to the sequence of words in the adversarial document. This metric is critical for assessing whether the adversarial text maintains linguistic coherence and naturalness and minimizes its likelihood of being detected as manipulated content.

Readability. To evaluate the readability of generated adversarial documents, we use the Dale-Chall readability score [15], which estimates text complexity based on familiar word usage and sentence structure. This metric compares the text against a list of 3,000 commonly known words and uses the proportion of unfamiliar words and average sentence length to determine the grade level needed for comprehension. Higher scores indicate greater difficulty and help assess whether adversarial modifications impact the text's accessibility and readability.

Grammar Assessment We employ Grammarly [26] to assess the quality of the adversarial documents by submitting them to Grammarly website to obtain their overall quality score. This score reflects the grammatical accuracy, sentence structure, and stylistic appropriateness of the text. Therefore, we can quantify the impact of adversarial perturbations on the grammatical quality of the generated documents and determine whether the perturbations introduce noticeable errors.

Linguistic Acceptability For assessing linguistic acceptability, we utilize a language classification model trained on the CoLA (Corpus of Linguistic Acceptability) dataset [74] to evaluate whether a text adheres to standard linguistic norms. This model determines if the adversarial document maintains proper syntax, semantics, and overall coherence to meet the linguistic standards of natural language.

Human Evaluation. This aspect assesses the imperceptibility and fluency of adversarial documents generated by different methods alongside the corresponding original documents, to measure the extent to which adversarial modifications remain imperceptible to human readers. To properly evaluate the imperceptibility and fluency of both Easy-5 and Hard-5 target documents, human evaluation was conducted on the *Mixture* set described in Section 5.1.2. For this purpose, we recruited two annotators with strong English proficiency, including one graduate student and one undergraduate student with non-Computer Science backgrounds, to better approximate typical reader judgments of naturalness and detectability. Prior to the main annotation task, we conducted an interactive training session with the annotators, following a written guideline and a calibration quiz with gold explanations, which we have released in our repository to facilitate reproducibility and support future human-evaluation studies. During this session, we clarified the scoring rubric and resolved questions to promote consistent application of the criteria. The training emphasized relying exclusively on human judgment, without the use of any AI-assisted or automated tools. Importantly, the training materials used synthetic, method-agnostic examples of fluency issues and suspicious artifacts, rather than examples drawn from any specific attack method, to avoid biasing annotators toward particular manipulation patterns.

After the training phase, each annotator was presented with a series of documents in randomized order, where the document was either the original document or an adversarial version generated by EMPRA or one of the baseline attacks. Annotators were not informed of the document type (original vs. adversarial). Following the approach used in prior work [9, 39, 44, 77], for each of the documents, annotators completed two tasks. (1) *Imperceptibility:* a binary judgment indicating whether the document appears natural and trustworthy for the query, with *(0)* denoting noticeable inconsistencies or signs of manipulation such as awkward insertions, redundant content, or unnatural phrasing and *(1)* denoting that the document reads as a coherent, well-formed passage, with no obvious artifacts that would raise suspicion for a typical reader. (2) *Fluency:* a 5-point ordinal rating (1 to 5) reflecting readability and linguistic quality, considering grammatical correctness, coherence, and overall naturalness, where 1 indicates very poor fluency and 5 indicates highly fluent text.

After completing the task, annotators were asked to review their ratings and judgments to improve internal consistency. This evaluation helps verify that adversarial documents remain difficult to detect by humans while maintaining adequate readability and perceived trustworthiness in realistic settings. We then summarize these judgments at the method level as follows. For each method, we report the mean imperceptibility rate and mean fluency score over the evaluated documents, computed separately for original documents and adversarial documents. Specifically, we first average the two annotators' ratings for each document and then aggregate across documents to obtain the method-level scores. To quantify annotation reliability, we measure inter-annotator agreement using Cohen's kappa [10] for imperceptibility as a binary variable, and weighted Cohen's kappa [11] for fluency as a five-point ordinal scale.

5.3 Models

5.3.1 Victim NRMs. Consistent with prior research [9, 39], we select the `msmarco-MiniLM-L-12-v2`[1] model as our primary victim black-box neural ranking model, denoted as $\mathcal{M}_V$. This cross-encoder-based ranker is fine-tuned on the MS MARCO training set, leveraging `MiniLM` [70] as its foundational language model for learning query-document semantic mapping. The model has demonstrated high retrieval effectiveness in terms of Mean Reciprocal Rank (MRR@10), as evidenced in Table 2.

To assess the robustness of different attacking strategies across various victim NRMs, we extend our investigation to include two additional models: `ms-marco-electrabase`[2] [60], and `DistilRoBERTa-base` [62]. Both models are fine-tuned on the MS MARCO training dataset and are based on distinct underlying language models compared to the primary victim NRM. Notably, the surrogate NRMs are not trained with pseudo-relevance feedback from these models. This setup enables a comprehensive evaluation of the effectiveness and robustness of different attack methods when targeting these additional victim NRMs.

5.3.2 Surrogate NRMs. Surrogate NRMs are employed as foundational components for executing surrogate-aware attack scenarios by interacting with the victim model through query submissions, leveraging the attacker's knowledge as outlined in Section 3.2, to approximate the victim model's behavior effectively. Consequently, varying numbers of queries from MS MARCO eval small dataset are employed to query the primary victim model $\mathcal{M}_V$ and train distinct surrogate models based on the re-ranked list of documents generated by the victim model. As proposed by the authors in [9], two query sets are utilized for this purpose, leading to the training of two surrogate models based on the pre-trained `BERT-base` [17] using different query quantities:

- $\mathcal{M}_{S_1}$: Trained on the full set of 6,837 eval small queries from MS MARCO, serving as an ID surrogate model.
- $\mathcal{M}_{S_2}$: Trained on a smaller subset of 200 eval small queries from MS MARCO, also serving as an ID surrogate model with considerably less number of training queries.

Table 2 compares the retrieval effectiveness of these two surrogate models against the primary target victim model $\mathcal{M}_V$ on the MS MARCO dev small set. As shown in the table, $\mathcal{M}_{S_1}$ is identified as the best-performing surrogate model, denoted as $\mathcal{M}_{S_{\text{best}}}$, due to its highest imitation capability, performing closely to the victim model. In contrast, $\mathcal{M}_{S_2}$ provides a less effective alternative which reflects the impact of its smaller training data. Both models serve as effective ID surrogates, facilitating the adversarial attack process by offering relevance predictions within the same data distribution as the victim model.

5.3.3 Generic NRMs. Generic NRMs serve as key components in implementing surrogate-agnostic attack scenarios, as detailed in Section 3.2. Unlike surrogate models, generic NRMs do not depend on pseudo-relevance labels generated through querying the victim model for training a neural ranking model. Instead, these models are pre-trained models that capture the concept of relevance by fine-tuning on diverse datasets, learning general principles of query-document

[1] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2
[2] https://huggingface.co/cross-encoder/ms-marco-electra-base

Table 2. The retrieval effectiveness (MRR@10) of the first-stage retriever (BM25), the primary victim model ($\mathcal{M}_V$), the surrogate models ($\mathcal{M}_{S_1}$ and $\mathcal{M}_{S_2}$), and the generic models ($\mathcal{M}_{G_{1-4}}$) on the dev small set of MS MARCO.

Model	BM25	$\mathcal{M}_V$	$\mathcal{M}_{S_1}$	$\mathcal{M}_{S_2}$	$\mathcal{M}_{G_1}$	$\mathcal{M}_{G_2}$	$\mathcal{M}_{G_3}$	$\mathcal{M}_{G_4}$
MRR@10	18.4	39.5	37.0	23.0	19.6	16.6	18.9	21.0

Instruction: You are an expert assistant in the field of information retrieval. Given the query and the document below, perform an adversarial attack on the document to make it rank high for the query, while maintaining the original content and structure of the document. Simply generate the perturbed document without explanation.

Query: {query}

Document: {document}

Perturbed Document:

Fig. 2. Prompt used by `LLM-Prompt` for generating adversarially perturbed documents.

matching. Generic models serve as an alternative approach in the black-box attack setting, eliminating the need for extensive querying of the victim model and enabling attack strategies that are independent of the victim model's specific behavior.

We utilize a variety of pre-trained embedding models that have been fine-tuned on publicly available datasets commonly used for training neural ranking models. These models are utilized as generic NRMs, designed to represent out-of-distributions scenarios to ensure the robustness and adaptability of `EMPRA`. The selected models are as follows:

- $\mathcal{M}_{G_1}$: A `mpnet-base` model [69] fine-tuned on a set of questions and answers from the NQ dataset[3].
- $\mathcal{M}_{G_2}$: A `distilbert-base` model [62] fine-tuned on the SQuAD2.0 dataset[4].
- $\mathcal{M}_{G_3}$: A `T5` model fine-tuned on the CommonQA and NLI datasets to create a sentence embedding model, referred to as `sentence-t5-base` [49][5].
- $\mathcal{M}_{G_4}$: A `BERT-base` model fine-tuned on the NQ dataset, following the methodology outlined in [9].

The retrieval effectiveness of the generic models on dev small is also reported in Table 2. Among these models, $\mathcal{M}_{G_4}$ is identified as the best-performing generic model, denoted as $\mathcal{M}_{G_{\text{best}}}$, in terms of retrieval effectiveness, while $\mathcal{M}_{G_2}$ exhibits the lowest performance. By employing these generic models as out-of-distribution models within the `EMPRA` framework, we can thoroughly assess the robustness and adaptability of our attack method.

5.3.4 Baselines. To demonstrate the performance of our attacking method, we conduct a comparative study against state-of-the-art baseline methods across word-level-based, trigger-based, and prompt-based categories. The following methods serve as our benchmarks:

- `Query+` [39] is a simple baseline that adds the query at the beginning of the target document. While the query could be inserted at any position within the document, placing it at the beginning yielded the most effective attack results.
- `LLM-Prompt` [3]: This baseline evaluates the performance of direct, zero-shot adversarial generation using `GPT-4` (`gpt-4-1106-preview`). Following the prompt design illustrated in Figure 2, the model is instructed to rewrite

[3]https://huggingface.co/tomaarsen/mpnet-base-nq
[4]https://huggingface.co/Pennywise881/distilbert-base-uncased-mnr-squadv2
[5]https://huggingface.co/sentence-transformers/sentence-t5-base

the target document to improve its ranking for the query while strictly preserving the original content and structure.

- PRADA [77] detects important terms in the target document using the surrogate model and replaces at-most 20 tokens with their synonyms within an embedding space.
- PAT [39] adds trigger words with the max length of 12 at the beginning of the target document. This method leverages a surrogate model in an incentivized manner to investigate whether the addition of these words enhances the document's ranking.
- Brittle-BERT [71] also adds trigger words at the beginning of the target document with the max length of 12.
- IDEM [9] generates 500 connection sentences using BART [33] with the max length of 12 and selects the best one in terms of the relevance and fluency trade-off in order to inject it into the original document for creating the adversarial document. The authors reported that adding sentences longer than 12 will maintain the attack performance at almost the same level.
- AttChain [44] iteratively perturbs the target document over several interaction rounds using an LLM guided by higher-ranked anchor documents returned by the neural ranking model. Through these repeated query-aware perturbations over the target document, its ranking is improved. We reproduce this baseline using the authors' best-performing LLM and adhered to their original hyperparameter configuration.

For implementing the baselines, we used the authors' released GitHub repositories and ensured that all settings were configured according to those reported in their official papers.

5.3.5 Implementation Details. For the implementation of the transporter function, we employ an L_∞ distance with a radius (ϵ) of 0.01, a step size of 0.1, and 25 iterations to guide the movement of sentence embeddings towards the anchors. For the transformer function, we run a single refinement pass and use greedy decoding by setting the beam width to 1 to select the highest-probability token at each generation step. Furthermore, the interpolation coefficient α in Equation 12 is set to 0.5. The impact of the number of iterations and α on the attack performance is investigated in the next section.

6 Results and Findings

In order to demonstrate the effectiveness of our proposed EMPRA method, we report and analyze the experimental results through the lens of the following research questions:

- **RQ1:** What is the impact of using various surrogate/generic NRMs on the attack performance of EMPRA?
- **RQ2:** How does EMPRA compare to baseline methods in terms of attack performance, fluency, and content fidelity in both in-distribution (ID) and out-of-distribution (OOD) settings?
- **RQ3:** How does the attack performance of EMPRA compare to baseline methods across various victim models?
- **RQ4:** How do different hyper-parameter settings affect the attack performance and semantic coherence of adversarial documents generated by EMPRA?
- **RQ5:** How well do the adversarial documents generated by EMPRA maintain quality, naturalness, and linguistic acceptability compared to baseline methods, and to what extent can they evade detection by human evaluators and automated detection systems?
- **RQ6:** How does the computational overhead of EMPRA compare to that of existing adversarial attack baselines?

Table 3. Attack performance of EMPRA across six different neural ranking models used for generating adversarial documents targeting the primary victim model $\mathcal{M}_V$.

		Easy-5					Hard-5				
Model	Fine-tune data	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓
$\mathcal{M}_{S_1}$	MS MARCO	99.9	95.6	99.8	72.5	34.4	99.9	64.9	87.0	948.4	47.1
$\mathcal{M}_{S_2}$	MS MARCO	99.4	83.1	98.5	68.6	35.4	99.5	47.5	73.2	909.9	50.3
$\mathcal{M}_{G_1}$	NQ	99.4	85.2	98.3	69.0	33.3	99.5	47.6	76.2	915.8	43.3
$\mathcal{M}_{G_2}$	SQuAD2.0	99.3	84.4	98.1	68.7	35.7	99.6	45.4	74.1	913.3	44.3
$\mathcal{M}_{G_3}$	CommonQA+NLI	99.4	82.5	98.2	68.4	34.7	99.5	46.0	74.5	910.1	46.7
$\mathcal{M}_{G_4}$	NQ	99.7	74.3	97.6	66.2	36.3	99.6	35.1	64.2	884.4	50.8

6.1 Surrogate-Agnostic Attack Performance Evaluation of EMPRA

To answer **RQ1**, we evaluate the robustness of EMPRA across diverse relevance-scoring models by running the attack with six in-domain and out-of-domain NRMs, each using a different sentence-embedding space during adversarial generation, while targeting the victim model $\mathcal{M}_V$. The results, presented in Table 3, report the impact of these models on the attack performance of EMPRA. As shown in the table, the first two models, $\mathcal{M}_{S_1}$ and $\mathcal{M}_{S_2}$, are surrogate NRMs trained on in-distribution data to mimic the behavior of the victim model. In contrast, the remaining models, $\mathcal{M}_{G_{1-4}}$, function as generic NRMs trained on out-of-distribution data, serving as relevance scoring models.

As shown in Table 3, EMPRA consistently achieves high attack performance across all NRMs, highlighting its robustness and surrogate-agnostic adaptability to variations in relevance scoring. Specifically, it achieves an Attack Success Rate (ASR) exceeding 99%, while successfully boosting over 97% and 64% of adversarial documents into the top-50 ($\%r \leq 50$) on the Easy-5 and Hard-5 subsets, respectively. In terms of more stringent metrics such as boosted in top-10 ($\%r \leq 10$) and how much the rank is boosted (Boost), $\mathcal{M}_{S_1}$ achieves the highest $\%r \leq 10$ (95.6%) and the largest rank boost (72.5) over Easy-5 subset, indicating that an in-distribution surrogate more effectively pushes adversarial documents to top ranks. Among the out-of-distribution generic NRMs $\mathcal{M}_{G_1}$ (85.2%) and $\mathcal{M}_{G_2}$ (84.4%) perform comparably well in terms of $\%r \leq 10$, illustrating that even OOD-based generic NRMs can cause substantial rank elevation.

For the more challenging Hard-5 subset, $\mathcal{M}_{S_1}$ offers the highest $\%r \leq 10$ (64.9) and rank boost (948.4). Meanwhile, OOD-trained generic NRMs such as $\mathcal{M}_{G_1}$ maintain competitive performance, achieving $\%r \leq 10$ values above 40% with boosts over 900. Although ID-trained surrogates like $\mathcal{M}_{S_1}$ provide a slight advantage in pushing adversarial documents to top-10 positions, OOD-trained generic models effectively undermine the victim model's ranking consistency while producing fluent and coherent adversarial texts, as indicated by $\mathcal{M}_{G_1}$'s lowest perplexity (PPL) in both Easy-5 and Hard-5 subsets.

A critical observation is that the performance difference between surrogate NRMs and generic NRMs may not justify the significant computational cost and time required to train surrogate models in dynamic attack scenarios where victim models frequently change in architecture and relevance criteria. The surrogate-agnostic nature of EMPRA allows it to effectively leverage generic NRMs, eliminating the need for frequent retraining. This design choice enhances its practicality and establishes EMPRA as a versatile attack method adaptable to various ranking systems in real-world black-box attack settings.

Table 4. Performance comparison on MS MARCO Dev across attack, fluency, and fidelity metrics targeting victim $\mathcal{M}_V$ using best-performing surrogate ($\mathcal{M}_{S_{\text{best}}}$) and generic ($\mathcal{M}_{G_{\text{best}}}$) NRMs. For attack performance metrics, † denotes statistically significant improvements over the best-performing baseline ($p < 0.05$). ↓ indicates metrics where lower values are preferred. While most methods achieve near 100% attack success rate (ASR), they vary significantly in their ability to position target documents within the top-10 ($\%r \leq 10$), where they are most visible to users.

MS MARCO Dev										
		Easy-5								
		Attack Performance				Fluency		Content Fidelity (%)		
Model	Method	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	Read.↓	LOR	SS	LC
-	`Original`	-	-	-	-	37.3	9.8	-	-	-
	`Query+`	100.0	86.9	99.2	70.3	45.4	9.6	100.0	97.7	99.2
	`LLM-Prompt`	94.1	65.0	90.1	49.9	49.0	11.0	76.5	92.0	91.9
$\mathcal{M}_{S_{\text{best}}}$	`PRADA`	77.9	3.5	46.2	23.2	94.4	9.9	92.7	94.9	90.2
	`Brittle-BERT`	98.7	81.3	96.7	67.3	107.9	10.7	100.0	95.5	98.8
	`PAT`	89.6	30.6	73.8	41.9	50.9	9.9	100.0	97.2	98.8
	`IDEM`	99.7	87.4	99.0	70.3	36.4	9.4	100.0	97.3	98.7
	`AttChain`	99.8	78.2	98.8	67.8	37.4	9.8	91.5	95.2	75.5
	`EMPRA`	**99.9**†	**95.6**†	**99.8**†	**72.5**†	34.4	9.2	100.0	95.2	95.3
$\mathcal{M}_{G_{\text{best}}}$	`PRADA`	71.5	1.9	37.5	19.1	91.5	9.8	93.7	95.1	91.5
	`Brittle-BERT`	90.0	43.4	80.1	46.2	117.7	11.0	100.0	95.3	98.8
	`PAT`	51.1	2.7	22.9	2.0	46.8	9.8	100.0	97.7	98.9
	`IDEM`	98.8	65.3	93.8	61.9	37.7	9.4	100.0	97.9	98.3
	`AttChain`	87.6	44.2	76.6	44.5	38.7	9.8	93.4	96.3	79.4
	`EMPRA`	**99.7**†	**74.3**†	**97.6**†	**66.2**†	36.3	9.2	100.0	96.2	97.7
		Hard-5								
		Attack Performance				Fluency		Content Fidelity (%)		
Model	Method	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	Read.↓	LOR	SS	LC
-	`Original`	-	-	-	-	50.5	9.0	-	-	-
	`Query+`	100.0	47.8	78.3	955.1	67.5	9.0	100.0	97.4	99.1
	`LLM-Prompt`	99.3	28.7	59.4	873.8	58.7	10.2	86.8	91.7	88.7
$\mathcal{M}_{S_{\text{best}}}$	`PRADA`	68.0	0.0	0.1	65.2	154.4	9.2	87.8	93.9	87.1
	`Brittle-BERT`	100.0	61.5	85.9	965.5	152.5	10.1	100.0	95.2	98.4
	`PAT`	98.0	6.2	20.1	589.1	71.4	9.2	100.0	96.9	98.7
	`IDEM`	99.8	54.3	79.3	933.0	54.9	8.9	100.0	96.8	98.4
	`AttChain`	100.0	55.8	**87.3**	**967.4**	51.3	9.3	89.9	92.9	72.4
	`EMPRA`	99.9	**64.9**†	87.0	948.4	47.1	8.8	100.0	94.0	95.2
$\mathcal{M}_{G_{\text{best}}}$	`PRADA`	71.9	0.0	0.1	73.4	168.7	9.3	86.8	93.6	86.0
	`Brittle-BERT`	**99.9**	17.7	47.6	845.2	156.8	10.3	100.0	95.0	98.7
	`PAT`	79.0	0.0	0.7	92.9	64.2	9.0	100.0	97.5	98.8
	`IDEM`	99.8	29.1	57.9	866.2	56.0	8.8	100.0	97.5	98.4
	`AttChain`	99.3	24.4	50.9	827.5	55.5	9.2	94.1	95.2	81.4
	`EMPRA`	99.6	**35.1**†	**64.2**†	**884.4**†	50.8	8.7	100.0	95.3	97.8

6.2 Attack Performance and Content Integrity Comparison in ID and OOD Settings

To answer **RQ2**, we evaluate the attack performance and content integrity of our proposed method, `EMPRA`, in comparison with established state-of-the-art baselines targeting the primary victim model $\mathcal{M}_V$. The evaluation is conducted using the best-performing surrogate model ($\mathcal{M}_{S_{\text{best}}}$) to represent the in-distribution (ID) setting and the best-performing generic model ($\mathcal{M}_{G_{\text{best}}}$) to represent the out-of-distribution (OOD) setting for generating adversarial documents. These models, introduced in Section 5.3, are the best-performing surrogate and generic NRMs in terms of retrieval effectiveness, as previously mentioned and demonstrated by MRR@10 in Table 2. In addition, the underlying architecture of these models is based on the BERT-base model, which is also the foundation for several state-of-the-art baselines such as Brittle-BERT, thereby ensuring a fair and meaningful comparison. We report the results of our experiments on the MS MARCO Dev, TREC DL 2019, and TREC DL 2020 in Tables 4, 5, and 6, respectively. These tables present attack performance using ASR, $\%r \leq 10$, $\%r \leq 50$, and Boost; fluency using PPL and readability (denoted as Read.); and content fidelity using lexical overlap recall (LOR), semantic similarity (SS), and logical consistency (LC) for EMPRA and all baselines across Easy-5 and Hard-5 targets under both ID and OOD NRMs. Our analysis reveals several key findings:

(i) Attack Performance. We observe that adversarial attacks consistently enhance document rankings across both surrogate and generic models and both sets of target documents. Notably, while nearly all evaluated methods achieve an Attack Success Rate (ASR) close to 100%, they vary significantly in their capacity to boost documents into high-visibility top-10 positions ($\%r \leq 10$). This distinction is critical because placement within the top-10 represents the highest probability of user exposure, which is the ultimate objective of adversarial ranking manipulation. Based on the experimental results, EMPRA demonstrates superior performance over all baselines across Easy-5 target documents on all three datasets and achieves higher attacking performance across most metrics on Hard-5 documents, particularly in the OOD setting where it surpasses almost all baselines in ranking metrics. For instance, when $\mathcal{M}_{\mathcal{S}_{\text{best}}}$ is used to generate adversarial documents across Easy-5 group, EMPRA achieves dominant boosted top-10 of 95.6% on MS MARCO Dev, 96.2% on TREC DL 2019, and 93.9% on TREC DL 2020. Conversely, while IDEM generally performs well, it falls short of outperforming Query+ in the majority of scenarios across these datasets. LLM-Prompt and AttChain occupy intermediary positions, surpassing trigger-based and word-level methods but lagging behind Query+, IDEM, and EMPRA in overall performance on the Easy-5 subset. While AttChain utilizes chain-of-thought prompting to generate adversarial documents, it struggles to match the effectiveness of EMPRA's embedding-level perturbations in promoting documents into the top-10.

Lower boosted top-k values in Hard-5 target documents compared to Easy-5 ones are attributed to containing more irrelevant information relative to the query. Consequently, effective perturbations are required to increase exposure likelihood to users. PRADA and PAT exhibit limited effectiveness in boosting Hard-5 target documents within the top-10 or top-50 across all three datasets. Conversely, EMPRA emerges as the most effective method for high-visibility ranking, boosting nearly 65% and 87% of documents into the top-10 and top-50, respectively, on MS MARCO Dev using $\mathcal{M}_{\mathcal{S}_{\text{best}}}$, while maintaining the lowest perplexity and readability grade level. This dominance is further confirmed on TREC DL 2019 and 2020, where EMPRA achieves top-10 success rates of 66.9% and 67.7% respectively, consistently outperforming other baselines. This superiority over one of the best baseline, IDEM, amounts to a relative improvement of 19.52% in boosting top-10 and 9.70% in boosting top-50 documents on MS MARCO Dev, with even higher relative gains of 45.1% and 33.5% in top-10 ranking observed on TREC DL 2019 and 2020, respectively.

Compared to AttChain, EMPRA increases the boosted top-10 from 55.8% to 64.9% on MS MARCO Dev. Although AttChain achieves a good boosted top-50 of 87.3% and Boost of 967.4 on $\mathcal{M}_{\mathcal{S}_{\text{best}}}$ for Hard-5 targets, its logical consistency is notably lower, 75.5% on MS MARCO Dev and 71.8% on TREC DL 2019, meaning that information inside the original document is not well preserved. The superiority of EMPRA over other attack baselines, particularly IDEM and AttChain, can be attributed to its ability to generate adversarial texts that not only maintain semantic proximity to the query but also with the top-ranked document and its sentences. This enables more effective adversarial texts that, when appended to the target document, significantly boost its ranking, as shown by metrics such as boosted top-k.

(ii) Robustness Across In-Distribution and Out-of-Distribution Settings. One of the main important aspects of an effective adversarial attack strategy is its robustness in attack performance against ID and OOD models. However,

Table 5. Performance comparison on TREC DL 2019 across attack, fluency, and fidelity metrics targeting victim $\mathcal{M}_V$ using best-performing surrogate ($\mathcal{M}_{S_{\text{best}}}$) and generic ($\mathcal{M}_{G_{\text{best}}}$) NRMs. For attack performance metrics, † denotes statistically significant improvements over the best-performing baseline ($p < 0.05$). ↓ indicates metrics where lower values are preferred. While most methods achieve near 100% attack success rate (ASR), they vary significantly in their ability to position target documents within the top-10 ($\%r \leq 10$), where they are most visible to users.

TREC DL 2019										
		Easy-5								
		Attack Performance				Fluency		Content Fidelity (%)		
Model	Method	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	Read.↓	LOR	SS	LC
–	`Original`	–	–	–	–	37.3	9.8	–	–	–
	`Query+`	100.0	84.8	99.4	69.8	45.1	9.6	100.0	97.7	99.2
	`LLM-Prompt`	93.0	63.5	89.1	48.7	48.6	11.1	78.3	92.2	92.5
$\mathcal{M}_{S_{\text{best}}}$	`PRADA`	78.2	2.6	46.9	23.3	96.2	9.9	92.5	94.9	89.4
	`Brittle-BERT`	97.0	77.5	95.0	64.5	110.8	10.7	100.0	95.4	98.7
	`PAT`	88.0	31.4	73.8	40.1	52.2	9.9	100.0	97.1	98.8
	`IDEM`	99.9	85.9	98.8	69.4	36.5	9.4	100.0	97.3	98.7
	`AttChain`	99.8	78.5	99.4	67.7	38.1	9.9	91.7	95.2	76.9
	`EMPRA`	**100.0**	**96.2**†	**100.0**†	**72.2**†	36.0	9.4	100.0	95.1	95.8
$\mathcal{M}_{G_{\text{best}}}$	`PRADA`	73.1	1.6	39.4	19.7	97.0	9.8	93.4	95.0	90.8
	`Brittle-BERT`	87.5	42.1	78.9	44.9	120.9	11.1	100.0	95.2	98.8
	`PAT`	51.1	3.6	22.7	-0.4	47.1	9.8	100.0	97.7	98.9
	`IDEM`	98.1	63.9	92.9	60.6	38.2	9.4	100.0	97.9	98.5
	`AttChain`	97.6	67.0	93.6	61.9	39.8	9.8	92.5	95.6	78.4
	`EMPRA`	**99.8**†	**78.7**†	**97.3**†	**67.1**†	40.0	9.4	100.0	96.2	97.7
		Hard-5								
		Attack Performance				Fluency		Content Fidelity (%)		
Model	Method	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	Read.↓	LOR	SS	LC
–	`Original`	–	–	–	–	61.5	9.3	–	–	–
	`Query+`	100.0	49.8	78.1	950.1	83.5	9.4	100.0	97.2	99.0
	`LLM-Prompt`	99.4	27.2	55.3	866.2	70.3	10.6	89.1	92.4	91.2
$\mathcal{M}_{S_{\text{best}}}$	`PRADA`	71.1	0.0	0.2	77.1	225.7	9.7	86.5	93.7	86.3
	`Brittle-BERT`	100.0	61.9	86.2	962.7	206.7	10.7	100.0	94.8	98.3
	`PAT`	98.1	8.4	25.5	627.1	90.0	9.6	100.0	96.5	98.6
	`IDEM`	100.0	46.1	73.6	913.7	65.7	9.1	100.0	96.7	98.5
	`AttChain`	100.0	54.5	86.0	967.2	62.1	9.5	90.5	92.8	71.8
	`EMPRA`	100.0	**66.9**†	**92.1**†	**979.1**†	55.7	9.1	100.0	93.5	94.7
$\mathcal{M}_{G_{\text{best}}}$	`PRADA`	75.6	0.0	0.2	85.5	255.1	9.7	85.0	93.3	83.4
	`Brittle-BERT`	99.9	16.1	47.6	848.3	210.9	10.8	100.0	94.7	98.6
	`PAT`	80.5	0.2	0.5	110.1	80.0	9.4	100.0	97.2	98.7
	`IDEM`	99.8	26.2	54.3	842.3	66.6	9.0	100.0	97.3	98.1
	`AttChain`	100.0	39.0	70.7	931.4	64.9	9.4	93.9	94.5	75.8
	`EMPRA`	100.0	**44.2**†	**74.4**†	**943.2**†	61.1	9.1	100.0	94.8	97.0

the performance of methods such as `PRADA`, `Brittle-BERT`, and `PAT` is heavily dependent on the surrogate models for adversarial text generation, resulting in lack of robustness across generic models trained on OOD data and a significant decrease in performance variability. For example, the attack performance of `Brittle-BERT` declines sharply on MS MARCO Dev across Easy-5 target documents when the external NRM is changed from $\mathcal{M}_{S_{\text{best}}}$ to $\mathcal{M}_{G_{\text{best}}}$, with the boosted top-10 value decreasing from 81.3% to 43.4%. This drop is even more severe on TREC DL 2019, where `Brittle-BERT` decreases from 77.5% to 42.1%, and on TREC DL 2020 Hard-5 targets, where it falls from 65.2% to just 20.1%.

Table 6. Performance comparison on TREC DL 2020 across attack, fluency, and fidelity metrics targeting victim $\mathcal{M}_V$ using best-performing surrogate ($\mathcal{M}_{S_{\text{best}}}$) and generic ($\mathcal{M}_{G_{\text{best}}}$) NRMs. For attack performance metrics, † denotes statistically significant improvements over the best-performing baseline ($p < 0.05$). ↓ indicates metrics where lower values are preferred. While most methods achieve near 100% attack success rate (ASR), they vary significantly in their ability to position target documents within the top-10 ($\%r \leq 10$), where they are most visible to users.

TREC DL 2020										
		Easy-5								
		Attack Performance				Fluency		Content Fidelity (%)		
Model	Method	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	Read.↓	LOR	SS	LC
–	Original	–	–	–	–	40.2	9.9	–	–	–
	Query+	99.9	85.5	99.3	69.9	49.4	9.7	100.0	97.7	99.2
	LLM-Prompt	93.3	61.7	89.7	49.5	49.4	11.1	77.1	92.2	92.8
$\mathcal{M}_{S_{\text{best}}}$	PRADA	78.2	2.8	46.4	23.1	105.4	9.9	92.3	94.9	87.6
	Brittle-BERT	97.0	79.0	93.6	64.6	125.4	10.8	100.0	95.4	98.8
	PAT	89.1	30.5	72.7	41.2	54.5	9.9	100.0	97.1	98.9
	IDEM	99.6	84.0	98.4	69.1	38.8	9.5	100.0	97.3	98.6
	AttChain	99.6	77.6	98.2	67.2	39.1	9.9	91.4	95.3	73.7
	EMPRA	**100.0**†	**93.9**†	**99.8**†	**72.2**†	38.5	9.3	100.0	95.3	95.8
$\mathcal{M}_{G_{\text{best}}}$	PRADA	73.2	1.7	37.7	19.3	104.1	9.9	93.3	95.0	90.4
	Brittle-BERT	86.5	42.2	77.6	44.0	127.3	11.1	100.0	95.3	98.8
	PAT	53.2	2.3	23.6	2.7	50.9	9.9	100.0	97.7	99.0
	IDEM	98.4	60.4	91.3	59.6	39.2	9.4	100.0	97.9	98.4
	AttChain	97.5	67.5	93.0	61.9	44.3	9.8	92.7	95.7	77.2
	EMPRA	**99.8**†	**78.3**†	**98.5**†	**67.5**†	42.2	9.4	100.0	96.2	97.7
		Hard-5								
		Attack Performance				Fluency		Content Fidelity (%)		
Model	Method	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	PPL↓	Read.↓	LOR	SS	LC
–	Original	–	–	–	–	52.8	9.1	–	–	–
	Query+	100.0	46.0	77.3	946.5	73.2	9.1	100.0	97.2	99.1
	LLM-Prompt	99.4	27.6	59.7	876.2	63.3	10.4	89.2	92.2	91.7
$\mathcal{M}_{S_{\text{best}}}$	PRADA	69.0	0.0	0.0	67.8	215.9	9.5	86.3	93.7	84.6
	Brittle-BERT	100.0	65.2	84.2	958.7	189.3	10.4	100.0	94.9	98.5
	PAT	98.3	9.4	25.6	627.8	77.5	9.4	100.0	96.5	98.7
	IDEM	100.0	50.7	74.8	919.7	57.1	8.9	100.0	96.7	98.4
	AttChain	100.0	57.8	86.8	966.9	51.9	9.4	90.3	92.8	71.2
	EMPRA	99.9	**67.7**	**89.5**	**977.2**†	49.8	9.0	100.0	93.8	95.7
$\mathcal{M}_{G_{\text{best}}}$	PRADA	74.9	0.0	0.1	80.4	222.9	9.5	85.1	93.5	84.6
	Brittle-BERT	100.0	20.1	47.8	834.8	192.4	10.6	100.0	94.7	98.7
	PAT	80.8	0.0	0.4	112.2	68.7	9.2	100.0	97.3	98.8
	IDEM	99.9	25.5	53.1	854.9	58.0	8.8	100.0	97.2	98.2
	AttChain	100.0	36.6	68.8	927.2	57.5	9.2	92.7	94.5	78.7
	EMPRA	99.9	**44.6**†	**71.6**	**930.2**	56.4	8.9	100.0	95.2	97.5

The relative robustness of IDEM and AttChain is dataset-dependent. Across Easy-5 target documents over MS MARCO Dev, IDEM is more stable, achieving 65.3% top-10 success with $\mathcal{M}_{G_{\text{best}}}$ versus 87.4% with $\mathcal{M}_{S_{\text{best}}}$, while AttChain drops from 78.2% to 44.2%. Conversely, on TREC DL 2020, AttChain demonstrates superior resilience, maintaining 67.5% with $\mathcal{M}_{G_{\text{best}}}$ after achieving 77.6% with $\mathcal{M}_{S_{\text{best}}}$, whereas IDEM falls from 84.0% to 60.4%. This suggests that AttChain's anchor-guided approach provides transferable relevance for TREC, though its surrogate-aware feedback loop remains susceptible to the specific distribution of MS MARCO Dev.

In contrast, EMPRA exhibits the greatest stability across various external NRMs, as its adversarial text generation does not rely on any specific surrogate model characteristics and instead leverages fundamental geometric shifts in the embedding space, making it more adaptable to real-world attacking scenarios. Based on the results, EMPRA maintains dominant effectiveness across all settings, preserving a top-10 success rate of 74.3% on MS MARCO Dev when using $\mathcal{M}_{G_{\text{best}}}$ compared to 95.6% with $\mathcal{M}_{S_{\text{best}}}$ across Easy-5 target documents. This performance is significantly higher than the next-best baselines, IDEM and AttChain, which achieve success rates of 65.3% and 44.2%, respectively. This superiority is further evidenced in a more challenging scenarios, such as the TREC DL 2020 Hard-5 targets, where EMPRA maintains a leading success rate of 44.6% with $\mathcal{M}_{G_{\text{best}}}$, while AttChain and IDEM decrease to 36.6% and 25.5%, respectively.

(iii) Textual Fluency and Content Fidelity. Another essential aspect in adversarial document generation is semantic fluency and content fidelity, measured through perplexity, readability, lexical overlap recall, semantic similarity, and logical consistency compared to baseline adversarial documents. Lower perplexity and better readability ensure that adversarial documents remain imperceptible and are less likely to be detected by detection models, while high fidelity scores indicate that the logical flow, facts, and critical content of the original document are preserved. Our findings reveal significant perplexity increases, particularly with Brittle-BERT and PRADA, notably evident in Hard-5 target documents across all three datasets. Query+, LLM-Prompt, and PAT exhibit moderate levels of perplexity, striking a balance between complexity and fluency. However, adversarial documents generated by Query+ are more easily detected and filtered due to simply appending the query text to the document, thereby undermining the purpose of the attack. Notably, EMPRA surpasses IDEM and AttChain in perplexity, particularly in Hard-5 target documents, claiming the top spot. This suggests EMPRA achieves superior attack performance while maintaining lower perplexity levels compared to both baselines and original documents in many cases. Additionally, EMPRA demonstrates almost the best readability scores among the baselines by covering the lowest grade-level required for comprehension consistently on MS MARCO Dev, TREC DL 2019, and TREC DL 2020.

Regarding content fidelity, EMPRA achieves a perfect lexical overlap recall score of 100.0% across all settings, matching Query+, IDEM, PAT, and Brittle-BERT, which all preserve the original text without deletion. However, PRADA shows a slight drop in lexical overlap recall to 86.8% in Hard-5 on MS MARCO Dev and 86.5% on TREC DL 2019, indicating some loss of original tokens. In terms of semantic similarity and logical consistency, methods like IDEM, PAT, and Brittle-BERT generally maintain high scores above 95% comparable to EMPRA. In contrast, LLM-Prompt and AttChain exhibit notable declines across all benchmarks. On the MS MARCO Dev Hard-5 setting, LLM-Prompt drops to 76.5% in lexical overlap recall, while AttChain decreases to 89.9% in lexical overlap recall and 72.4% in logical consistency. This trend persists on TREC DL 2019 and 2020, where AttChain's logical consistency falls to 71.8% and 71.2%, respectively. Specifically for AttChain, although it maintains high topical relevance with at least 92.9% semantic similarity, its significantly lower logical consistency implies it may fail to preserve and boost the logic, facts, and critical content of the original document, which can ultimately undermine the attack objective.

6.3 Attack Performance Evaluation Across Various Victim Models

In real-world settings, attackers may not have detailed information about the target victim NRM used by search engines. In addition, due to continuous training, model updates, and potential replacement with different NRMs, the victim model changes frequently. As a result, an effective attacking method must perform reasonably across various victim

Table 7. Attack performance of adversarial documents using $\mathcal{M}_{S_{\text{best}}}$ and $\mathcal{M}_{G_{\text{best}}}$ against different victim NRMs on MS MARCO Dev. † denotes statistically significant improvements over the best-performing baseline ($p < 0.05$).

MS MARCO Dev									
Victim Model	Method	Easy-5				Hard-5			
$\mathcal{M}_{S_{\text{best}}}$		ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost
ELECTRA	PRADA	59.9	3.3	31.6	31.7	35.3	0.0	0.1	4.9
	Brittle-BERT	98.5	83.6	95.8	132.3	99.9	**73.5**	**88.0**	**710.1**
	PAT	88.6	29.0	66.6	89.5	78.6	6.2	18.0	323.5
	IDEM	99.5	85.8	97.9	133.6	98.2	56.3	75.9	667.4
	AttChain	99.1	70.1	96.6	129.9	99.9	52.7	82.3	698.2
	EMPRA	**99.8**†	**92.2**†	**99.3**†	**136.1**†	97.7	66.8	84.0	685.9
DistilRoBERTa	PRADA	62.9	4.2	29.4	31.7	57.7	0.0	0.4	27.4
	Brittle-BERT	96.6	71.9	92.2	142.8	99.5	57.3	78.4	731.9
	PAT	85.7	25.6	61.0	90.1	89.3	5.1	18.1	422.4
	IDEM	99.0	83.2	96.9	148.9	98.9	57.3	78.3	724.3
	AttChain	99.3	70.3	95.9	146.8	**99.8**	50.9	81.1	**747.4**
	EMPRA	**99.6**†	**89.6**†	**98.6**†	**152.3**†	97.8	**65.1**†	**84.6**†	735.5
$\mathcal{M}_{G_{\text{best}}}$		ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost
ELECTRA	PRADA	52.8	2.2	27.1	25.7	36.5	0.0	0.1	3.9
	Brittle-BERT	88.2	46.8	78.2	104.9	**97.6**	20.7	47.3	576.2
	PAT	57.0	3.0	26.0	18.6	39.4	0.1	0.5	-16.6
	IDEM	97.7	60.2	89.2	120.6	96.1	27.2	51.8	574.4
	AttChain	87.4	40.5	73.6	96.0	94.7	21.7	47.0	542.7
	EMPRA	**98.9**†	**69.3**†	**94.4**†	**127.4**†	96.2	**35.8**†	**60.0**†	**605.9**†
DistilRoBERTa	PRADA	59.9	3.3	25.8	26.7	58.3	0.0	0.2	32.4
	Brittle-BERT	86.9	39.3	73.4	112.2	96.5	15.9	39.6	593.7
	PAT	52.9	2.5	22.6	13.4	61.2	0.1	0.7	59.5
	IDEM	96.8	57.9	86.8	132.6	97.1	27.2	52.9	633.5
	AttChain	88.1	40.7	72.4	111.3	96.2	21.7	45.0	600.5
	EMPRA	**98.1**†	**64.5**†	**91.4**†	**140.1**†	96.2	**33.7**†	**58.7**†	**651.9**†

models without the need for frequent retraining of surrogate models and regeneration of adversarial documents, which can be both costly and time-consuming.

To answer **RQ3**, we adopt two different victim models and evaluate the performance of EMPRA and baseline attack methods using one surrogate NRM and one generic NRM on the same targeted Easy-5 and Hard-5 documents from ms-marco-MiniLM-L-12-v2. For this purpose, we compare the original rankings and the rankings after adversarial attacks when evaluated by the new victim model, with results presented in Table 7 for MS MARCO Dev, and Tables 8 and 9 for TREC DL 2019 and 2020, respectively. Consistent with the previous section, we report how various victim NRMs rank adversarial documents generated by the best-performing surrogate NRM ($\mathcal{M}_{S_{\text{best}}}$) and the best-performing generic NRM ($\mathcal{M}_{G_{\text{best}}}$). These models, selected based on their retrieval effectiveness, represent the best-case (ID) and extreme-case (OOD) scenarios, respectively. This provides a comprehensive view of the attack method's robustness across varying victim model configurations.

Table 8. Attack performance of adversarial documents using $\mathcal{M}_{S_{\text{best}}}$ and $\mathcal{M}_{G_{\text{best}}}$ against different victim NRMs on TREC DL 2019. † denotes statistically significant improvements over the best-performing baseline ($p < 0.05$).

TREC DL 2019									
Victim Model	Method	Easy-5				Hard-5			
$\mathcal{M}_{S_{\text{best}}}$		ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost
ELECTRA	PRADA	59.1	2.5	31.0	29.2	36.1	0.0	0.1	0.6
	Brittle-BERT	96.6	80.6	94.4	126.3	**100.0**	**72.3**	88.5	699.7
	PAT	87.0	30.6	66.1	87.1	79.7	8.0	22.5	354.5
	IDEM	99.2	84.5	97.9	128.9	97.6	46.8	71.0	631.5
	AttChain	99.5	71.6	96.5	125.8	99.9	51.3	82.2	687.5
	EMPRA	**99.7**	**90.7**†	**99.5**†	**132.8**†	99.7	66.6	**88.9**	**705.0**
DistilRoBERTa	PRADA	58.7	2.5	26.7	28.0	55.6	0.0	0.0	27.1
	Brittle-BERT	95.1	67.8	88.1	134.9	99.6	58.7	79.3	724.1
	PAT	84.3	23.5	60.4	91.4	91.9	7.1	22.0	456.4
	IDEM	99.0	83.7	95.8	146.5	99.5	48.1	73.1	702.6
	AttChain	98.9	69.9	94.0	142.8	100.0	51.1	81.0	739.0
	EMPRA	**99.8**†	**86.3**	**99.0**†	**150.2**†	100.0	**65.5**†	**88.4**†	**760.0**†
$\mathcal{M}_{G_{\text{best}}}$		ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost
ELECTRA	PRADA	53.5	2.2	28.3	23.3	39.5	0.0	0.2	8.6
	Brittle-BERT	88.6	48.3	78.9	102.7	97.2	19.7	46.7	571.3
	PAT	54.6	3.7	26.0	15.8	39.1	0.1	0.5	-13.9
	IDEM	96.9	59.6	89.5	116.4	93.8	23.0	47.6	532.8
	AttChain	97.0	60.1	90.5	119.4	**99.6**	34.5	63.1	639.6
	EMPRA	**99.2**†	**75.1**†	**96.3**†	**126.8**†	99.4	**45.6**†	**69.8**†	**656.5**†
DistilRoBERTa	PRADA	56.8	1.9	23.5	21.8	57.2	0.0	0.0	27.9
	Brittle-BERT	88.3	36.9	72.5	111.7	97.5	16.5	42.9	600.7
	PAT	51.3	2.9	21.7	17.6	63.4	0.0	0.8	73.5
	IDEM	96.3	55.8	84.9	130.2	98.0	25.5	49.4	613.0
	AttChain	97.0	60.3	89.0	136.3	99.5	33.6	61.8	687.5
	EMPRA	**98.9**†	**68.5**†	**92.0**†	**140.7**†	**99.8**	**43.5**†	**68.9**†	**705.0**†

Cross-Victim Performance on MS MARCO Dev. Based on the results in Table 7, EMPRA demonstrates the most robust performance in cross-victim NRM attacks compared to the baselines, exhibiting the lowest decrease ratio when transitioning from the ID surrogate model ($\mathcal{M}_{S_{\text{best}}}$) to the OOD generic model ($\mathcal{M}_{G_{\text{best}}}$), maintaining boosted top-50 rankings above 90% across Easy-5 target documents and above 58% across Hard-5 target documents. In contrast, PRADA and PAT exhibit the lowest attack performance due to their heavy reliance on the surrogate model for adversarial document generation, necessitating continuous surrogate model retraining for optimal performance, rendering them impractical for real-world applications.

IDEM, AttChain, and Brittle-BERT occupy an intermediate position, displaying moderate attack performance. Notably, Brittle-BERT exhibits high attack performance when $\mathcal{M}_{S_{\text{best}}}$ is used against ELECTRA-base; however, this performance diminishes by more than half when shifted to $\mathcal{M}_{G_{\text{best}}}$. Furthermore, the high perplexity scores reported for Brittle-BERT in Table 4 indicate poor linguistic quality, which directly undermines the imperceptibility of the generated adversarial documents. While AttChain demonstrates competitive results by achieving the highest average

Table 9. Attack performance of adversarial documents using $\mathcal{M}_{S_{\text{best}}}$ and $\mathcal{M}_{G_{\text{best}}}$ against different victim NRMs on TREC DL 2020. † denotes statistically significant improvements over the best-performing baseline ($p < 0.05$).

TREC DL 2020									
Victim Model	Method	Easy-5				Hard-5			
$\mathcal{M}_{S_{\text{best}}}$		ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost
ELECTRA	PRADA	61.2	3.5	32.4	34.6	34.8	0.0	0.1	5.7
	Brittle-BERT	97.4	81.2	94.2	132.3	99.9	**73.8**	**88.2**	689.8
	PAT	88.2	26.6	66.8	87.9	79.3	9.5	24.6	362.8
	IDEM	99.0	81.0	96.8	134.5	98.3	52.3	71.5	644.9
	AttChain	99.1	69.2	95.5	130.5	**100.0**	52.0	83.2	690.1
	EMPRA	**99.9**†	**90.4**†	**99.2**†	**139.0**†	99.8	69.4	87.4	**703.4**†
DistilRoBERTa	PRADA	64.9	4.0	28.8	38.4	61.4	0.0	0.4	48.4
	Brittle-BERT	95.2	70.8	89.7	144.9	99.8	61.9	77.9	721.5
	PAT	84.0	24.9	60.4	88.6	89.0	8.7	24.5	442.4
	IDEM	99.2	80.9	96.2	151.6	99.3	52.8	71.9	704.4
	AttChain	99.2	69.1	94.7	150.8	99.9	52.7	82.6	743.8
	EMPRA	**99.9**†	**88.5**†	**99.5**†	**159.8**†	99.9	**69.8**†	**87.7**†	**758.9**†
$\mathcal{M}_{G_{\text{best}}}$		ASR	$\%r \leq 10$	$\%r \leq 50$	Boost	ASR	$\%r \leq 10$	$\%r \leq 50$	Boost
ELECTRA	PRADA	55.0	2.0	26.4	23.7	37.6	0.0	0.0	11.3
	Brittle-BERT	87.1	46.2	77.0	107.5	97.0	21.0	49.3	562.9
	PAT	58.3	3.1	25.5	20.0	40.4	0.0	0.6	-9.8
	IDEM	97.3	55.3	86.3	119.8	95.8	21.9	48.7	554.0
	AttChain	97.2	61.4	90.1	125.3	99.1	33.3	62.7	632.8
	EMPRA	**99.2**†	**73.3**†	**95.1**†	**131.9**†	**99.4**	**44.6**†	**67.4**†	**643.9**
DistilRoBERTa	PRADA	61.2	3.4	25.9	31.3	62.1	0.0	0.3	47.9
	Brittle-BERT	86.7	38.1	72.9	119.2	96.4	17.8	43.7	594.1
	PAT	55.3	2.8	21.0	18.8	63.6	0.1	1.2	87.9
	IDEM	96.6	55.1	84.3	134.7	97.7	24.1	50.2	620.4
	AttChain	96.7	61.7	87.9	143.9	99.1	31.8	59.9	683.8
	EMPRA	**98.8**†	**70.0**†	**93.0**†	**148.9**†	**99.7**	**43.3**†	**65.2**†	**687.6**

boost scores in two specific Hard-5 scenarios against ELECTRA and DistilRoBERTa, it consistently falls behind EMPRA in nearly all other experimental configurations. Specifically, EMPRA maintains a substantial advantage in securing top-10 rankings ($\%r \leq 10$), which is the primary metric for user visibility. For instance, on Easy-5 targets against ELECTRA, EMPRA achieves a 92.2% success rate compared to AttChain's 70.1%. This consistent superiority across varied architectures validates EMPRA's victim model-agnostic capability to boost adversarial documents into high-visibility ranks, demonstrating its effectiveness against various victim model to maximize user exposure.

Cross-Victim Performance on TREC DL Benchmarks. To further assess the robustness of EMPRA beyond MS MARCO Dev, we extend our cross-victim evaluation to the TREC DL 2019 and TREC DL 2020 datasets, with results presented in Tables 8 and 9, respectively. The results indicate that the trends observed on MS MARCO Dev persist and become even more prominent on the TREC DL datasets, confirming EMPRA's superior generalizability over other datasets. A critical insight from this analysis is the significant performance degradation suffered by baseline methods

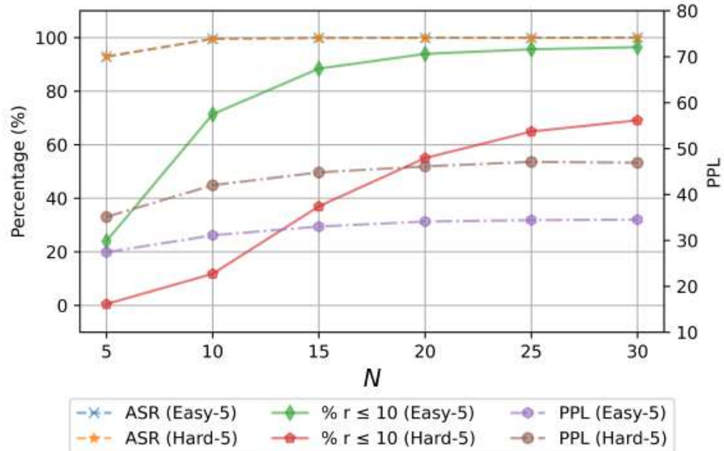

Fig. 3. Impact of the number of iterations.

when transitioning to the out-of-distribution (OOD) setting using the generic NRM $\mathcal{M}_{G_{\text{best}}}$. On TREC DL 2019 , EMPRA demonstrates exceptional stability on Easy-5 targets against the ELECTRA victim, maintaining a boosted top-10 of 75.1%. In comparison, AttChain achieves only 60.1%, while Brittle-BERT experiences a drastic decline to 48.3%. This sharp drop for Brittle-BERT suggests that its gradient-based token replacements are highly specific to the surrogate NRM and fail to transfer effective relevance signals.

This performance gap is further widened when analyzing Hard-5 targets on TREC DL 2020. EMPRA proves to be the only method capable of maintaining substantial ranking improvements against ELECTRA using $\mathcal{M}_{G_{\text{best}}}$ NRM, achieving a boosted top-10 of 44.6%. In contrast, the strongest baselines fail to generalize effectively as AttChain drops to 33.3%, and IDEM reduces significantly to 21.9%. A similar trend is observed against the DistilRoBERTa victim, where EMPRA achieves a top-10 rate of 43.3% for Hard-5 targets, outperforming AttChain and IDEM by substantial margins. The superior transferability of EMPRA across these diverse victim models and datasets, confirms that it does not merely exploit artifacts of a specific surrogate model but rather injects robust adversarial text that remain effective even against state-of-the-art neural rankers.

It is important to note that the average boost value exceeding 100 across Easy-5 target documents in Table 7, 8, and 9 occurs because the target documents were randomly sampled from rankings 51-100 of the primary victim model and may, in some cases, rank above 100 by the new victim models.

6.4 The Impact of Hyper-parameters

To answer **RQ4**, we evaluate EMPRA by exploring the impact of two hyper-parameters on its attack performance: 1) the number of iterations by the transporter function, and 2) the α interpolation coefficient, which balances relevance and coherence. The evaluation is conducted on the MS MARCO Dev dataset using the $\mathcal{M}_{S_{\text{best}}}$ model to analyze the performance of the proposed method under varying hyper-parameter settings.

Figure 3 shows the impact of the number of iterations performed by the transporter function. We observe that as the number of iterations increases from 20 to 30, the improvement of attack performance by EMPRA becomes less substantial,

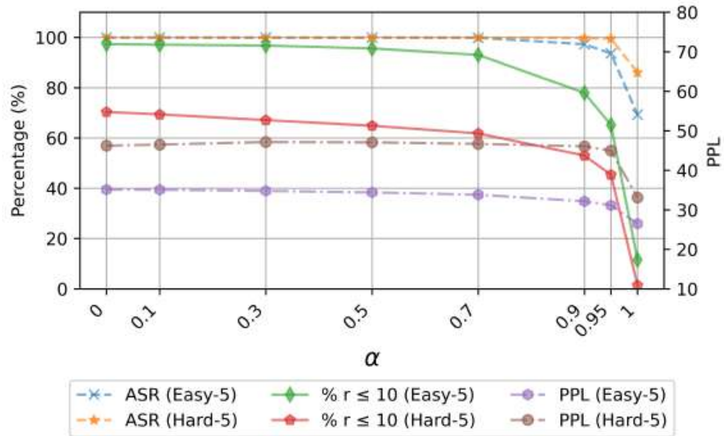

Fig. 4. Impact of the interpolation coefficient α.

particularly in comparison to the range of 5-20, especially noticeable with Easy-5 target documents. In terms of ASR, EMPRA can achieve comparable attack performance across both Easy-5 and Hard-5 target documents, indicating its capabilities of boosting both document sets.

Moreover, Figure 4 explores the impact of the interpolation coefficient α in Equation 12 that balances between semantic coherence and query relevance. It is shown that when α falls within the range of 0-0.95, the attack performance remains consistently high, indicating that the adversarial sentences exhibit both strong attack capabilities and low perplexity. However, as the emphasis on coherency reaches its peak at α equal to 1, the attack performance begins to decrease, particularly in terms of boosted top-10.

6.5 Quality and Naturalness Evaluation

In addition to evaluating attack performance, the quality, naturalness, linguistic acceptability, and imperceptibility of generated adversarial documents are important factors in maintaining reader confidence and achieving attack objectives. When reading the perturbed document the reader should not immediately suspect is has been manipulated. To assess these aspects and answer **RQ5**, we conduct an analysis to evaluate generated adversarial documents based on model and human evaluation metrics. Model-based metrics consist of text perplexity (PPL), grammar quality, and linguistic acceptability. Human-based evaluations metrics consist of imperceptibility and fluency, measured by human annotators as described in Section 5.2.3. We evaluate the quality and naturalness of the 'Mixture' target documents produced by each attacking method under $\mathcal{M}_{S_{\text{best}}}$ model using model-based and human-based evaluation metrics. These results are compared with the overall attack performance ($\%r \leq 10$) in Table 10. A detailed explanation of the metrics follows.

Model-Based Evaluation. Perplexity, measured using the GPT-2 model [57], serves as a proxy for fluency, with lower values indicating higher fluency. EMPRA achieves the lowest perplexity, indicative of its high fluency. To evaluate grammar quality, given the discontinuation of the Grammarly SDK as of January 10, 2024, we utilized the Grammarly website [26] to assess the overall quality score of each method's adversarial documents. Results indicate that documents generated by EMPRA closely match the quality of original documents in terms of grammar. LLM-Prompt attains the highest quality scores, reflecting its proficiency in generating text, without even the errors that might be observed

Table 10. Trade-off between attack performance (%$r \leq 10$) and the naturalness of adversarial documents generated by various attack methods. Naturalness is assessed using both model-based and human-based evaluation metrics. For ease of comparison, attack performance is taken from Table 4 ($\mathcal{M}_{S_{\text{best}}}$). EMPRA provides the best attack performance while maintaining among the best naturalness scores, often close to the original scores.

Method	Attack Performance (%$r \leq 10$)		Model-based Evaluation				Human-based Evaluation			
	Easy-5	Hard-5	PPL↓	Grammar	Acceptability Score	Class. Accuracy	Impercept.	kappa	Fluency	kappa
EMPRA	95.6	64.9	35.30	79.34	0.61	0.28	0.58	0.56	3.27	0.57
AttChain	78.2	55.8	37.64	81.72	0.61	0.34	0.55	0.69	3.33	0.62
IDEM	87.4	54.3	39.27	80.44	0.64	0.28	0.55	0.56	3.28	0.56
PAT	30.6	6.24	49.99	76.81	0.43	0.53	0.41	0.74	2.97	0.39
Brittle-BERT	81.3	61.5	114.96	71.34	0.20	0.94	0.16	0.53	2.83	0.52
PRADA	3.52	0.02	126.24	51.72	0.40	0.69	0.52	0.69	3.17	0.46
LLM-Prompt	65.0	28.7	53.42	87.03	0.66	0.16	0.66	0.59	3.66	0.54
Query+	86.9	47.8	45.03	74.91	0.46	0.53	0.47	0.62	3.19	0.53
Original	-	-	35.11	83.22	0.73	0.78	0.75	0.50	3.52	0.43

in the original document. However, its attack performance is considerably lower compared to EMPRA, particularly in boosting Hard-5 target documents.

To investigate the linguistic acceptability of the generated adversarial documents and explore whether they can be detected by trained Natural Language Processing (NLP) models, we employed the RoBERTa-base [6] classification model fine-tuned on the Corpus of Linguistic Acceptability (CoLA) [74] to specifically detect attacked texts. Using this model, we measure the linguistic acceptability scores of the original documents and their adversarial counterparts. In addition, the classification accuracy (denoted as 'Class. Accuracy' in the table) is also calculated to determine the accuracy of the model in detecting original documents vs adversarial documents correctly. The model correctly confirms that 78% of the original documents have not been detected as adversarial documents. Moreover, the model's accuracy in classifying the adversarial documents generated by EMPRA, IDEM, and LLM-Prompt is below 28% demonstrating that over 70% of these documents sufficiently resemble the original documents content and are linguistically acceptable, without containing any junk or garbage text. In contrast, AttChain's adversarial documents exhibit a higher detection rate, as 34% are flagged as linguistically unacceptable by the classification model. This higher detection rate suggests that while AttChain's LLM-based rewrites are fluent, they may introduce subtle structural or semantic patterns that trained classifiers can identify easier compared to EMPRA and IDEM.

Furthermore, other baselines achieve a accuracy of more than 50% having Brittle-BERT as the one with accuracy score of 94%. This shows that despite decent attack performance, its adversarial documents can be easily and accurately detected using an NLP classification model, pointing that model has most likely added junk irrelevant trigger terms.

Human-Based Evaluation. Following prior studies [9, 39, 44, 77], we evaluate imperceptibility (denoted as 'Impercept.' in the table) and fluency based on human annotations, as described in detail in Section 5.2.3. We report in Table 10 the mean imperceptibility rate and mean fluency score on the Mixture set for both the original documents and the adversarial documents produced by each method, along with inter-annotator agreement measured by Cohen's kappa for imperceptibility and weighted Cohen's kappa for fluency. Overall, EMPRA preserves human-perceived naturalness well, with an imperceptibility rate of 0.58 and a fluency score of 3.27, which are comparable to competitive baselines such as

[6] https://huggingface.co/textattack/roberta-base-CoLA

IDEM and AttChain. In contrast, trigger-based methods degrade perceived quality more substantially, with PAT and especially Brittle-BERT receiving the lowest imperceptibility and fluency scores, indicating that their perturbations are more likely to appear suspicious or disruptive to typical readers.

LLM-Prompt achieves the highest imperceptibility and fluency scores. To assess whether these differences are robust, we additionally conduct paired statistical testing on the Mixture set using the Wilcoxon signed-rank test. While EMPRA ranks second in mean imperceptibility, the difference between EMPRA and LLM-Prompt is not statistically significant ($p > 0.05$), suggesting that the available evidence is insufficient to conclude a meaningful difference in perceived detectability between the two methods. In contrast, LLM-Prompt exhibits a statistically significant improvement in fluency ($p < 0.05$), which is expected given its substantially larger model capacity and extensive linguistic pretraining. It is worth mentioning that naturalness alone does not determine attack success, as a method may produce highly readable and seemingly trustworthy text while still varying substantially in its ability to manipulate ranking. We therefore treat human judgments as a complementary signal that characterizes perceived detectability and readability, and examine their interaction with attack effectiveness in the subsequent trade-off analysis.

The reported inter-annotator agreement levels validate the reliability of these comparative trends. As shown in Table 10, Cohen's kappa for imperceptibility generally falls in the moderate to substantial agreement range, suggesting that annotators often converged on whether a document appears suspicious or natural under the same criteria. Likewise, the weighted Cohen's kappa scores for fluency are mostly moderate, which is expected for a five-point ordinal scale where borderline cases can be interpreted differently. Overall, these agreement levels are consistent with prior human evaluations [9, 39, 44, 77] of adversarial text quality and support the stability of the reported imperceptibility and fluency trends across methods.

Trade-off Between Attack Performance and Naturalness. While evaluating the quality and naturalness of adversarial documents is essential, it is important to consider how these factors interact with attack performance. In this context, there should be a balance between achieving high attack performance and maintaining the naturalness and quality of the generated adversarial documents. For instance, although attacking methods like Brittle-BERT are effective in promoting the ranking of documents, they can easily be filtered out by linguistic acceptability models with the accuracy of 94%, ruining the attack. In the trade-off between attack performance and naturalness, EMPRA excels by providing both high attack performance and high naturalness. While LLM-Prompt, when prompted to generate an adversarial document, produces the most natural text according to linguistic measures and human assessment, EMPRA substantially outperforms it in terms of attack performance boosting the document among top-10 and increase the user exposure to adequately natural documents that maintain the attack objectives. At the same time, EMPRA outperforms other perturbation methods in this trade-off, emerging as a well-rounded approach that offers a balance between attack performance and naturalness.

6.6 Time Complexity Analysis

To address **RQ6** and assess the practicality of EMPRA, we analyze its time complexity and compare it with baseline attacks for generating adversarial documents over a collection of attacked query-document pairs. Let Q denote the number of queries to be attacked and D denote the number of target documents per query, yielding $Q \cdot D$ attacked pairs. Let S be the number of sentences in a target document. We characterize how the time complexity of each method scales with these attack inputs.

EMPRA. For each query-document pair, EMPRA operates in two distinct stages. First, it generates a pool of adversarial text candidates by processing each sentence against a constant-size anchor set. This generation stage scales linearly with the number of sentences, incurring an $O(S)$ cost. Second, EMPRA performs adversarial document construction by evaluating each generated candidate across $S + 1$ potential insertion positions. This systematic search introduces an additional $O(S)$ factor, resulting in a quadratic complexity of $O(S^2)$ per pair. Consequently, the total complexity across the experimental set is:

$$C_{\text{total}}^{\text{EMPRA}} = O(Q \cdot D \cdot S^2). \tag{13}$$

Baselines. We next compare how the baseline attacks scale with the number of attacked pairs $(Q \cdot D)$ and with document length S. Word-level-based and trigger-based methods such as PRADA, Brittle-BERT, and PAT utilize internal search loops or optimization procedures that do not explicitly scale with the document's sentence count. Their total complexity grows linearly with the number of attacked pairs:

$$C_{\text{total}}^{\text{PRADA}}, C_{\text{total}}^{\text{Brittle}}, C_{\text{total}}^{\text{PAT}} = O(Q \cdot D). \tag{14}$$

Similarly, AttChain relies on multi-round candidate generation where the primary overhead is fixed per document, leading to an $O(Q \cdot D)$ scaling. While these methods offer cheaper scaling, they often lack the explicit position-level optimization required to preserve document-level coherence. In contrast, IDEM generates a fixed pool of candidate sentences and evaluates them across all insertion positions, resulting in a linear scaling with respect to document length:

$$C_{\text{total}}^{\text{IDEM}} = O(Q \cdot D \cdot S). \tag{15}$$

Overall, all evaluated methods scale at least linearly with the volume of attacked pairs $(Q \cdot D)$, but differ in how they integrate adversarial content. While EMPRA exhibits an $O(Q \cdot D \cdot S^2)$ complexity, this increased overhead is a trade-off inherent to its dual-objective optimization. By explicitly searching for insertion placements that simultaneously maximize ranking exposure and semantic coherence, EMPRA provides more robust ranking manipulation and decent linguistic quality, making it imperceptible to humans and automated tools. Given that S is typically a small, bounded integer in standard retrieval benchmarks, this quadratic complexity remains highly practical. Specifically, for both IDEM and EMPRA, the number of sentences per document is significantly smaller than the total number of target documents $(S \ll D)$. Therefore, the quadratic overhead per pair is computationally negligible, and the total complexity for both methods effectively becomes $O(Q \cdot D)$. We complement this complexity analysis with quantitative execution time measurements under experimental settings in the following section.

6.7 Execution Time Cost Analysis

To complement our complexity analysis for **RQ6**, we measured the execution time required to generate adversarial documents for 1,000 query-document pairs on a dedicated RTX A6000 GPU. Figure 5 reports the time cost in hours alongside the corresponding attack performance based on $\%r \leq 10$ on the Easy-5 target sets of TREC DL 2019 under the best-performing surrogate model $\mathcal{M}_{S_{\text{best}}}$ (Table 5).

As shown in the figure, the competing methods exhibit distinct execution time–effectiveness patterns. At one extreme, high-computation approaches such as Brittle-BERT and PAT require substantial generation time while yielding only moderate or weak attack performance. Brittle-BERT represents the most expensive baseline, requiring over 140 hours to generate 1,000 adversarial documents with only moderate effectiveness, whereas PAT exhibits both high execution time and weak attack performance, making it the least favorable option in terms of efficiency. A second group consists of moderate-cost methods with divergent effectiveness, represented by IDEM and PRADA. IDEM achieves relatively strong attack performance, boosting 85.9% of documents into the top-10, but requires a substantially higher

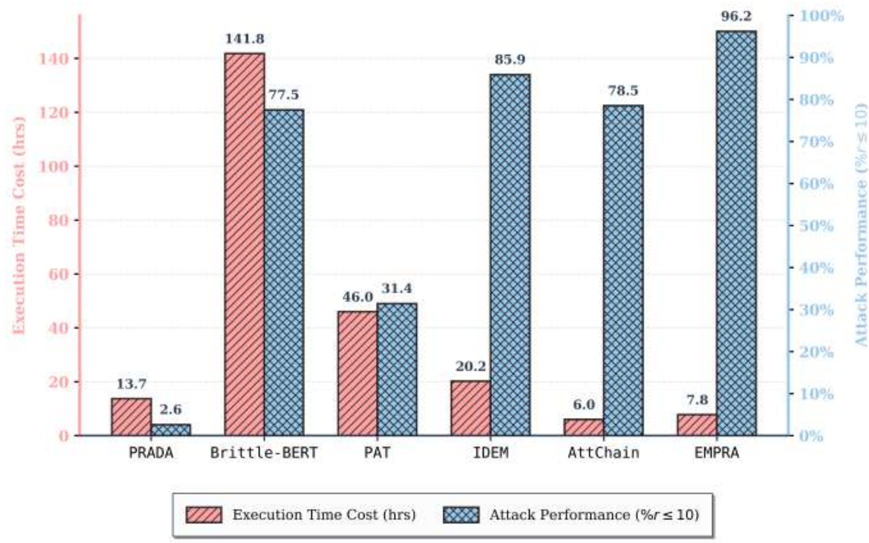

Fig. 5. Comparison of total generation time for producing 1,000 adversarial documents and corresponding attack performance ($\%r \leq 10$) for each attack method on the Easy-5 target sets, evaluated under the best-performing surrogate model $\mathcal{M}_{S_{\text{best}}}$ (Table 5). Our proposed EMPRA method achieves the strongest balance between attack performance and execution time cost, delivering high effectiveness with substantially lower time requirements than competing methods.

execution time of 20.2 hours to generate 1,000 adversarial documents due to its heavy reliance on large-scale generation and candidate evaluation. In contrast, PRADA operates at a comparatively low execution time of 13.7 hours, but fails to achieve meaningful attack success, illustrating that reduced computational cost alone does not guarantee effective ranking manipulation.

AttChain occupies a distinct position as a low-latency prompt-based method, requiring only 6.0 hours to complete generation while achieving a moderate attack performance of 78.5%. Its efficiency stems from its multi-round guided perturbation strategy relying on repeated LLM API calls for adversarial document generation, yet its effectiveness remains clearly below that of the top-performing approach. In contrast, EMPRA uniquely occupies the optimal region of the trade-off curve, combining near-minimal execution time with the highest attack performance. EMPRA achieves 96.2% attack performance while requiring only 7.8 hours to generate 1,000 adversarial documents. Compared with IDEM, EMPRA delivers stronger effectiveness while using less than half of the execution time. More importantly, when directly compared with AttChain, EMPRA achieves a relative improvement of 22.5%. This clear separation demonstrates that EMPRA improves the efficiency–effectiveness trade-off while also shifting the practical performance boundary, achieving a level of attack strength that competing fast methods fail to reach.

6.8 Adversarial Examples

In order to have a better qualitative understanding of how each adversarial method perturbs the document, we present examples of adversarial documents generated by EMPRA and baseline methods, including Query+, LLM-Prompt, PRADA, Brittle-BERT, IDEM, and AttChain in Table 11. These documents were generated in response to the query: *"Can anyone take prenatal vitamins?"* and target a specific document that was originally ranked at position 91. All documents, except those generated by Query+ and LLM-Prompt, were crafted using the $\mathcal{M}_{S_{\text{best}}}$ surrogate model. The examples are evaluated based on their new ranking positions, perplexity, logical consistency (LC), and whether they are deemed linguistically acceptable.

As shown in the table, Query+, Brittle-BERT, PAT, IDEM, AttChain, and EMPRA successfully promote the target document into the top-10 ranking positions for the given query. However, Brittle-BERT and PAT are flagged as unacceptable by the linguistic acceptability model (RoBERTa-base), indicating their adversarial modifications are easily detectable. Additionally, while Brittle-BERT and PAT achieve competitive ranking positions, they do so at the expense of significantly higher perplexity scores of 95.2 and 33.8, respectively. This undermines the fluency and naturalness of their generated documents. The reliance of Query+ on inserting the exact query sequence into the document also makes it highly detectable, particularly if systems are designed to filter out documents containing such exact matches.

In contrast, adversarial modifications by IDEM, AttChain, and EMPRA are subtle and impactful, as reflected in their lower perplexity scores and acceptance by the linguistic acceptability model. However, a closer examination reveals critical differences in content fidelity. While AttChain achieves a high ranking and the lowest perplexity of 17.3, it exhibits a critically low logical consistency score of 38.9%. This drop is attributed to its rewriting mechanism, which deletes and alters original content. Notably, EMPRA outperforms both IDEM and AttChain by achieving a better ranking position (1st) for the target document and obtaining a highly competitive perplexity score of 17.6. More importantly, unlike AttChain, EMPRA maintains a high logical consistency of 99.1%, demonstrating its ability to inject effective adversarial perturbations while preserving the integrity and facts of the original content. Its ability to craft high-quality, imperceptible modifications not only enhances ranking performance but also increases the likelihood of user exposure, underscoring its effectiveness in performing adversarial attacks.

LLM-Prompt and PRADA are the only two methods that fail to promote the target document into the top-10 rankings, achieving positions 33 and 49, respectively. While LLM-Prompt produces linguistically acceptable outputs, it suffers from the third-highest perplexity score and completely rewrites the target document. Furthermore, LLM-Prompt struggles to optimize ranking performance effectively, highlighting its limitation in crafting highly targeted adversarial modifications. PRADA demonstrates the poorest performance among all methods, achieving the lowest ranking boost and the highest perplexity score. Additionally, its output is not linguistically acceptable, with significant grammatical and semantic errors, such as "you are took" and "metallurgical." These issues make PRADA the least effective method in terms of both ranking performance and linguistic quality.

7 Conclusion and Discussion

In this paper, we introduced EMPRA, a novel method for executing adversarial attacks on black-box neural ranking models. EMPRA is a surrogate-agnostic method that operates independently of any specific surrogate model by utilizing two key components: the transporter and transformer functions. The transporter function adjusts sentence embeddings of the target document by shifting them closer to anchor texts, which include the target query, its top-ranked document, and the most similar sentence within that document. The transformer function then reconstructs these adjusted embeddings into coherent and fluent adversarial text. This dual-function design ensures that the generated adversarial texts can effectively manipulate the ranking of target documents while remaining imperceptible to human reviewers and automated systems when injected into the document.

EMPRA's performance is validated through its ability to significantly boost the rankings of target documents across diverse challenging scenarios. In our evaluations across three benchmark datasets including, MS MARCO V1 passage collection, TREC DL 2019, and TREC DL 2020, EMPRA consistently outperformed all state-of-the-art baselines. On the MS MARCO V1 passage collection, EMPRA successfully re-ranked nearly 96% of target documents from positions 51-100

Table 11. Adversarial documents generated using the $\mathcal{M}_{S_{\text{best}}}$ model (except for Query+ and LLM-Prompt) by various attack methods targeting a document originally ranked at position 91 for the query *"Can anyone take prenatal vitamins?"*. The new ranking positions, perplexity, logical consistency, and linguistic acceptability of the generated documents are reported. Modified text is highlighted in bold and removed text from the original document is shown with strikethrough for easy comparison, showcasing EMPRA's superior balance of attack effectiveness, fluency, logical consistency, and linguistic acceptability.

Query: *"Can anyone take prenatal vitamins?"*					
Method	**Document**	**Rank↓**	**PPL↓**	**LC (%)**	**Linguistically Acceptable**
Original	Always let your health care provider know what nutritional supplements you are taking. Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a woman's daily intake requirements for certain nutrients, such as folic acid (folate), calcium, and iron, will increase.	91	19.2	–	Yes
Query+	**Can anyone take prenatal vitamins?** Always let your health care provider know what nutritional supplements you are taking. Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a woman's daily intake requirements for certain nutrients, such as folic acid (folate), calcium, and iron, will increase.	1	20.4	99.2	Yes
LLM-Prompt	**Always consult your caregiver or medical specialist before starting any nutritional supplements, including those designed for prenatal care. Can individuals not expecting to conceive also consider prenatal vitamins? It's a common inquiry. These vitamins and minerals blends, typically referred to as prenatal vitamins, are critical during gestation. During such a crucial timeline, a female body's daily intake necessities for pivotal nutrients, such as folic acid (more commonly known by its synthetic form, folate), calcium, and iron, will see a notable escalation.**	33	49.7	99.4	Yes
PRADA	Always let your health care **purveyor** know what nutritional supplements you are **took**. Prenatal vitamins consist of a variety of vitamins and **metallurgical**. During pregnancy, a woman's daily **admitting** requirements for certain **vitamin**, such as folic acid (folate), calcium, and iron, will increased.	49	100.5	98.6	No
Brittle-BERT	**aanatnat anyone can...va taking 167 x <token> whether** Always let your health care provider know what nutritional supplements you are taking. Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a woman's daily intake requirements for certain nutrients, such as folic acid (folate), calcium, and iron, will increase.	1	95.2	98.8	No
PAT	**no, if anyone could even take preca** Always let your health care provider know what nutritional supplements you are taking. Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a womanÃ¢Â€Â™s daily intake requirements for certain nutrients, such as folic acid (folate), calcium, and iron, will increase.	1	33.8	99.1	No
IDEM	**Children, not pregnant mothers, cannot take prenatal vitamins.** Always let your health care provider know what nutritional supplements you are taking. Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a woman's daily intake requirements for certain nutrients, such as folic acid (folate), calcium, and iron, will increase.	2	18.4	97.2	Yes
AttChain	Always ~~let~~ **inform** your ~~health care~~ **healthcare** provider ~~know what~~ **about** the nutritional supplements you are taking. ~~Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a woman's daily intake requirements for certain nutrients, such as~~ Prenatal vitamins, **including** folic acid (folate), calcium, and iron, ~~will increase.~~ **play a crucial role during pregnancy. Can anyone take prenatal vitamins?**	2	17.3	38.9	Yes
EMPRA	**During pregnancy, anyone can take a prenatal vitamin (folic acid, iron, and calcium) to increase their daily requirements for these nutrients.** Always let your health care provider know what nutritional supplements you are taking. Prenatal vitamins consist of a variety of vitamins and minerals. During pregnancy, a woman's daily intake requirements for certain nutrients, such as folic acid (folate), calcium, and iron, will increase.	1	17.6	99.1	Yes

into the top 10. This superior efficacy was maintained across the TREC benchmarks, where EMPRA achieved a high boosted top-10 across all target documents. Furthermore, its ability to elevate 65% of hard target documents into the top-10 and 87% into the top-50 underscores its robustness in the most challenging scenarios. EMPRA stands out as a robust, efficient, and victim model-agnostic attack method, highlighting the urgent need for future research to develop defenses and practical countermeasures against such sophisticated adversarial technique.

At ingestion and indexing time, one line of defense is adversarial artifact detection and filtering based on signals such as linguistic acceptability, redundancy, and local coherence inconsistencies that can arise from injected content. At retrieval time, robustness can be improved via consistency checks across multiple rankers, cross-encoder verification of top-ranked candidates, and outlier detection for suspicious documents whose relevance signals are concentrated in short spans. Finally, defensive training strategies such as adversarial data augmentation and robustness-oriented fine-tuning can reduce sensitivity to small, targeted perturbations while maintaining retrieval effectiveness. We note that no single heuristic is sufficient in a black-box setting, and a systematic evaluation of defense combinations remains an important direction for future work.

References

[1] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access* 6 (2018), 14410–14430.

[2] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. 2021. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access* 9 (2021), 155161–155196.

[3] Amin Bigdeli, Negar Arabzadeh, Ebrahim Bagheri, and Charles L. A. Clarke. 2025. Adversarial Attacks against Neural Ranking Models via In-Context Learning. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (China) *(SIGIR-AP 2025)*. Association for Computing Machinery, New York, NY, USA, 211–220. https://doi.org/10.1145/3767695.3769517

[4] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lluís Màrquez, Chris Callison-Burch, and Jian Su (Eds.). Association for Computational Linguistics, Lisbon, Portugal, 632–642. https://doi.org/10.18653/v1/D15-1075

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017).

[6] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. Ieee, 39–57.

[7] Carlos Castillo, Brian D Davison, et al. 2011. Adversarial web search. *Foundations and trends® in information retrieval* 4, 5 (2011), 377–486.

[8] Jingfan Chen, Wenqi Fan, Guanghui Zhu, Xiangyu Zhao, Chunfeng Yuan, Qing Li, and Yihua Huang. 2022. Knowledge-enhanced black-box attacks for recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 108–117.

[9] Xuanang Chen, Ben He, Zheng Ye, Le Sun, and Yingfei Sun. 2023. Towards Imperceptible Document Manipulations against Neural Ranking Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 6648–6664. https://doi.org/10.18653/v1/2023.findings-acl.416

[10] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[11] Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin* 70, 4 (1968), 213.

[12] Gordon V Cormack, Mark D Smucker, and Charles LA Clarke. 2011. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval* 14 (2011), 441–465.

[13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR* abs/2102.07662 (2021). arXiv:2102.07662 https://arxiv.org/abs/2102.07662

[14] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[15] Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin* (1948), 37–54.

[16] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. 2021. A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[19] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 31–36. https://doi.org/10.18653/v1/P18-2006

[20] Wenqi Fan, Xiangyu Zhao, Xiao Chen, Jingran Su, Jingtong Gao, Lin Wang, Qidong Liu, Yiqi Wang, Han Xu, Lei Chen, et al. 2022. A comprehensive survey on trustworthy recommender systems. *arXiv preprint arXiv:2209.10117* (2022).

[21] Wenqi Fan, Xiangyu Zhao, Qing Li, Tyler Derr, Yao Ma, Hui Liu, Jianping Wang, and Jiliang Tang. 2023. Adversarial Attacks for Black-Box Recommender Systems Via Copying Transferable Cross-Domain User Profiles. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[22] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. 2020. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*. 3019–3025.

[23] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. 2018. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference*. 381–392.

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[25] Shreya Goyal, Sumanth Doddapaneni, Mitesh M Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *Comput. Surveys* 55, 14s (2023), 1–39.

[26] Grammarly. 2023. Grammarly. https://app.grammarly.com/ Accessed: 2023-05-28.

[27] Zoltán Gyöngyi, Hector Garcia-Molina, et al. 2005. Web Spam Taxonomy.. In *AIRWeb*, Vol. 5. Citeseer, 39–47.

[28] Niddal H Imam and Vassilios G Vassilakis. 2019. A survey of attacks against twitter spam detectors in an adversarial environment. *Robotics* 8, 3 (2019), 50.

[29] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8018–8025.

[30] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

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

[32] Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics* 10 (2022), 163–177.

[33] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[34] Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010* (2023).

[35] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6193–6202. https://doi.org/10.18653/v1/2020.emnlp-main.500

[36] Chen Lin, Si Chen, Hui Li, Yanghua Xiao, Lianyun Li, and Qian Yang. 2020. Attacking recommender systems with augmented user profiles. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 855–864.

[37] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[38] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.

[39] Jiawei Liu, Yangyang Kang, Di Tang, Kaisong Song, Changlong Sun, Xiaofeng Wang, Wei Lu, and Xiaozhong Liu. 2022. Order-Disorder: Imitation Adversarial Attacks for Black-box Neural Ranking Models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2025–2039.

[40] Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. 2024. Measuring, Evaluating and Improving Logical Consistency in Large Language Models. *arXiv preprint arXiv:2311.00685* (2024).

[41] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. 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*. 1647–1656.

[42] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Topic-oriented Adversarial Attacks against Black-box Neural Ranking Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information*

Retrieval. 1700–1709.

[43] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Multi-granular Adversarial Attacks against Black-box Neural Ranking Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1391–1400.

[44] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2025. Attack-in-the-chain: bootstrapping large language models for attacks against black-box neural ranking models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 12229–12237.

[45] Teng Long, Qi Gao, Lili Xu, and Zhangbing Zhou. 2022. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions. *Computers & Security* 121 (2022), 102847.

[46] Gallil Maimon and Lior Rokach. 2022. A universal adversarial policy for text classifiers. *Neural Networks* 153 (2022), 282–291.

[47] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. 2023. Text Embeddings Reveal (Almost) As Much As Text. arXiv:2310.06816 [cs.CL]

[48] 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. (2016).

[49] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1864–1874. https://doi.org/10.18653/v1/2022.findings-acl.146

[50] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[51] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[52] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 582–597.

[53] P Patil Swati, BV Pawar, and S Patil Ajay. 2013. Search engine optimization: A study. *Research Journal of Computer and Information Technology Sciences* 1, 1 (2013), 10–13.

[54] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the robustness of retrieval pipelines with query variation generators. In *European conference on information retrieval*. Springer, 397–412.

[55] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667* (2021).

[56] Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing* 492 (2022), 278–307.

[57] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[58] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 784–789. https://doi.org/10.18653/v1/P18-2124

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

[60] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[61] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 232–241.

[62] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[63] Minoru Sasaki and Hiroyuki Shinnou. 2005. Spam detection using text clustering. In *2005 International Conference on Cyberworlds (CW'05)*. IEEE, 4–pp.

[64] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*. 1528–1540.

[65] Bhambri Siddhant, Muku Sumanyu, Tulasi Avinash, and Buduru Arun Balaji. 2019. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667* (2019).

[66] Congzheng Song, Alexander Rush, and Vitaly Shmatikov. 2020. Adversarial Semantic Collisions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 4198–4210. https://doi.org/10.18653/v1/2020.emnlp-main.344

[67] Junshuai Song, Zhao Li, Zehong Hu, Yucheng Wu, Zhenpeng Li, Jian Li, and Jun Gao. 2020. Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 157–168.

[68] 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*. 191–203.

[69] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems* 33 (2020), 16857–16867.

[70] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[71] Yumeng Wang, Lijun Lyu, and Avishek Anand. 2022. BERT rankers are brittle: a study using adversarial document perturbations. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. 115–120.

[72] Zongwei Wang, Min Gao, Jundong Li, Junwei Zhang, and Jiang Zhong. 2022. Gray-box shilling attack: an adversarial learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 5 (2022), 1–21.

[73] Zongwei Wang, Junliang Yu, Min Gao, Wei Yuan, Guanhua Ye, Shazia Sadiq, and Hongzhi Yin. 2024. Poisoning Attacks and Defenses in Recommender Systems: A Survey. *arXiv preprint arXiv:2406.01022* (2024).

[74] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics* 7 (2019), 625–641.

[75] Chenwang Wu, Defu Lian, Yong Ge, Zhihao Zhu, and Enhong Chen. 2021. Triple adversarial learning for influence based poisoning attack in recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1830–1840.

[76] Chen Wu, Ruqing Zhang, Jiafeng Guo, Wei Chen, Yixing Fan, Maarten de Rijke, and Xueqi Cheng. 2022. Certified Robustness to Word Substitution Ranking Attack for Neural Ranking Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2128–2137.

[77] Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten De Rijke, Yixing Fan, and Xueqi Cheng. 2023. Prada: practical black-box adversarial attacks against neural ranking models. *ACM Transactions on Information Systems* 41, 4 (2023), 1–27.

[78] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[79] Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. Poisoning Retrieval Corpora by Injecting Adversarial Passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 13764–13775. https://doi.org/10.18653/v1/2023.emnlp-main.849

[80] Bin Zhou and Jian Pei. 2009. OSD: An online web spam detection system. In *In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, Vol. 9.

[81] Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2020. A Reinforced Generation of Adversarial Examples for Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 3486–3497. https://doi.org/10.18653/v1/2020.acl-main.319