Adversarial Hubness Detector: Detecting Hubness Poisoning in Retrieval-Augmented Generation Systems

Idan Habler
Cisco — OWASP
ihabler@cisco.com

Vineeth Sai Narajala
Cisco — OWASP
vineeth.sai@owasp.org

Stav Koren
Tel Aviv University
stavk@mail.tau.ac.il

Amy Chang
Cisco
changamy@cisco.com

Tiffany Saade
Cisco
tsaade@cisco.com

***Abstract*—Retrieval-Augmented Generation (RAG) systems are essential to contemporary AI applications, allowing large language models to obtain external knowledge via vector similarity search. Nevertheless, these systems encounter a significant security flaw: *hubness* - items that frequently appear in the top-k retrieval results for a disproportionately high number of varied queries. These hubs can be exploited to introduce harmful content, alter search rankings, bypass content filtering, and decrease system performance.**

We introduce ADVERSARIAL HUBNESS DETECTOR, an open-source security scanner that evaluates vector indices and embeddings to identify hubs in RAG systems. ADVERSARIAL HUBNESS DETECTOR presents a multi-detector architecture that integrates: (1) robust statistical hubness detection utilizing median/Median Absolute Deviation (MAD)-based z-scores, (2) cluster spread analysis to assess cross-cluster retrieval patterns, (3) stability testing under query perturbations, and (4) domain-aware and modality-aware detection for category-specific and cross-modal attacks. Our solution accommodates several vector databases (FAISS, Pinecone, Qdrant, Weaviate) and offers versatile retrieval techniques, including vector similarity, hybrid search, and lexical matching with reranking capabilities.

We evaluate ADVERSARIAL HUBNESS DETECTOR on Food-101, MS-COCO, and FiQA adversarial hubness benchmarks constructed using state-of-the-art gradient-optimized and centroid-based hub generation methods. ADVERSARIAL HUBNESS DETECTOR achieves 90% recall at a 0.2% alert budget and 100% recall at 0.4%, with adversarial hubs ranking above the 99.8th percentile. In testing, domain-scoped scanning recovered 100% of targeted attacks that evaded global detection. Production validation on 1M real web documents from MS MARCO demonstrates significant score separation between clean documents and adversarial content. Our work provides a practical, extensible framework for detecting hubness threats in production RAG systems. Our open-source implementation is available at https://github.com/cisco-ai-defense/adversarial-hubness-detector.

I. INTRODUCTION

Retrieval-Augmented Generation (RAG) systems have become an essential framework for grounding large language models (LLMs) in external knowledge [6]. RAG systems can deliver precise, current responses by extracting relevant items from a vector database prior to generation, eliminating the need for continuous model retraining. This framework facilitates several applications, including corporate knowledge repositories, customer support systems, and AI assistants across multiple sectors.

The fundamental mechanism of RAG systems is based on vector similarity search within high-dimensional embedding spaces. Upon submission of a query, it is transformed into a vector representation, and the system obtains the k most common items from a vector index. The collected texts are subsequently supplied as context to an LLM for answer generation. This architecture presents considerable benefits, although it also creates a crucial attack surface: the vector embedding space.

A. The Hubness Threat

Hubness is a recognized occurrence in high-dimensional spaces where specific locations frequently serve as nearest neighbors to several other points [13]. In information retrieval situations, this results in certain items appearing in the top-k results more frequently than anticipated. Hubness can arise organically; nevertheless, it also implies a significant security weakness that may be intentionally exploited.

Zhang *et al.* [19] demonstrated that attackers can craft embeddings that become "hubs" appearing in retrieval results for thousands of semantically unrelated queries. In their experiments on text-caption-to-image retrieval, a single hub generated using only 100 random queries was retrieved as the top-1 result for over 21,000 out of 25,000 test queries—far exceeding expected behavior for normal items.

Such *hub injection* attacks enable a form of universal **retrieval poisoning**: by inserting a malicious item into a database, an adversary can force irrelevant or harmful content to appear in a wide array of search results [19].

B. Real-World Attack Incidents

Real-world incidents underscore the feasibility and severity of retrieval poisoning:

- **Microsoft 365 Copilot Poisoning**: Zenity Labs demonstrated that "all you need is one document" to reliably mislead a production Copilot system [18]. A single crafted document made the Copilot confidently answer queries with attacker-chosen false facts.
- **GeminiJack Attack**: Noma Security's exploit showed that a single shared Google Doc with embedded instructions caused Google's Gemini AI search to exfiltrate months of private emails and documents [5], demonstrating a zero-click data breach through retrieval poisoning.
- **Content Injection**: Hubs enable attackers to inject malicious, misleading, or spam content that appears in responses to diverse queries.
- **Indirect Prompt Injection**: By placing prompt injection payloads in hub items, attackers can manipulate LLM behavior across many user queries [3].

- **The Promptware Kill Chain**: Recent research by [10] formalizes these exploits as part of a multi-step malware lifecycle. They demonstrate how retrieval poisoning serves as the primary *delivery vector* for **Promptware**—AI-native malware that utilizes RAG to move from initial access to lateral movement and data exfiltration and gain persistency.

These examples underscore a critical weakness in RAG architectures: the absence of a resilient poison detection layer to capture harmful payloads prior to their integration into the model's environment. In the absence of such identification, the RAG system autonomously compromises itself by considering contaminated external inputs as reliable information.

C. Challenges in Detection

Overcoming hubness is difficult as the concept of the attack exploits intrinsic characteristics of embedding spaces. Previous methods for decreasing *natural* hubness (e.g., similarity normalization [4], [16]) solely diminish hubs that are globally frequent neighbors. An adaptive adversary can establish a *domain-specific hub* that activates solely for inquiries on a single subject, circumventing overarching defenses [19].

Key detection challenges include:

1) **Statistical Robustness**: Hubs demonstrate hub rates exceeding the median by 5-10+ standard deviations, necessitating the use of robust statistical procedures that are resilient to the impact of outliers.
2) **Domain-Specific Attacks**: Advanced attackers establish hubs that focus on particular semantic categories (e.g., "medical advice" or "financial reports"), circumventing worldwide detection while exerting significant influence inside their designated domain.
3) **Cross-Modal Attacks**: In multimodal systems, items can be designed to seem related to queries in an alternate modality, taking advantage of modality boundaries that single-modality detection overlooks.
4) **Semantic Mismatch**: Hubs get high similarity scores at the expense of genuine semantic alignment even though their content frequently diverges from the queries that yield them.
5) **Retrieval Method Awareness**: Detection must be effective across several retrieval methods, including vector search, hybrid search, and reranking pipelines.

D. Contributions

We present ADVERSARIAL HUBNESS DETECTOR, the first comprehensive detection system for hubness in RAG systems. Our contributions include:

1) **Multi-Detector Architecture**: A pluggable detection framework combining hubness frequency analysis, cluster spread detection, stability testing, and deduplication—each targeting different aspects of hub behavior.
2) **Robust Statistical Methods**: Novel application of median/MAD-based z-scores that maintains numerical stability for detecting statistical anomalies in hub rate distributions.
3) **Domain-Aware Detection**: Methods for identifying hubs that focus on particular semantic categories via bucket analysis of specific RAG indices.
4) **Modality-Aware Detection**: Cross-modal anomaly detection for multimodal systems, incorporating parallel retrieval and late fusion architectures.
5) **Flexible Retrieval Support**: Integration with multiple vector databases and retrieval methods, including hybrid search and custom reranking algorithms through a plugin system and customizable interfaces.
6) **Open-Source Implementation**: A production-ready scanner with adapters to the most popular RAG frameworks.

Against SOTA hubs [19] (CLIP/Qwen3), ADVERSARIAL HUBNESS DETECTOR achieved **90% recall** at a 0.2% alert budget and **100% recall** at 0.4%. The Cluster Spread Detector provides critical lift (+10–20pp recall) for universal attacks. Production validation on 1 million MS MARCO [11] passages demonstrated **5.8× score separation** between adversarial hubs and the 99$^{\text{th}}$ percentile of clean data, with 0.1% operational overhead, confirming web-scale deployment feasibility.

The rest of this paper is organized as follows: Section II reviews related work. Section III presents our detection methodology and background. Section IV details the detection algorithms. Section V describes domain-aware and modality-aware detection. Section VI covers retrieval integration. Sections VII,VIII presents our evaluation. Section IX concludes.

II. RELATED WORK

A. Hubness in High-Dimensional Spaces

The hubness phenomenon was initially defined by Radovanović *et al.* [13], who illustrated that in high-dimensional spaces, specific points-termed **"hubs"**, frequently serve as nearest neighbors to numerous other points with disproportionate prevalence. Subsequent research investigated mitigation methods: Mutual Proximity [14] modifies distances according to local neighborhoods, while CSLS [4] standardizes similarities by employing average neighbor distances. Lin *et al.* [8] presented NeighborRetr to equilibrate hub centrality during training, and Wang *et al.* [16] introduced Dual Bank Normalization (DBNorm) as a post-processing method. However, these methodologies concentrate on *natural* hubness resulting from data distributions, rather than on adversarially constructed hubs.

B. Hubness Attacks

Zhang *et al.* [19] conducted the first extensive examination of hubness exploitation in multi-modal retrieval, demonstrating that adversaries can leverage the hubness phenomena to generate embeddings obtained from thousands of unrelated queries. Their research showed that a singular hub might prevail in top-1 outcomes for 84% of test queries in text-to-image retrieval, with hubs generalizing far beyond their training queries (100 queries used to create a hub affecting 21,000+

queries). Existing hubness mitigation techniques provide limited defense against targeted hubs, which can be crafted to target specific concepts while evading global detection. Our research expands upon these findings to establish a **detection framework** explicitly aimed at hubs.

Real-World Incidents: Security research firms have documented production RAG vulnerabilities. Zenity Labs [18] reported that "all you need is one document" to reliably mislead Microsoft 365 Copilot. Noma Security (GeminiJack) uncovered vulnerabilities where an indexed document with hidden instructions induced Google Gemini to leak confidential data—a zero-click data exfiltration attack. Greshake *et al.* [3] demonstrated indirect prompt injection through the retrieval pathway. These examples highlight the potential harm when retrieval is impaired; our research focuses on attacks that function at the embedding level, utilizing continuous vector spaces.

C. Common Retrieval Methods

Wu *et al.* [17] introduced RetrievalGuard, a demonstrably resilient 1-nearest neighbor image retrieval technique that preserves accuracy in the presence of adversarial perturbations. Hybrid search, which integrates vector similarity with lexical matching, has grown popular for enhancing retrieval quality [1]. Reranking methodologies employing cross-encoders [12] offer secondary filtering. Our work integrates with these approaches to provide defense-in-depth.

III. Methodology

A. Threat Model and Problem Definition

We assume an adversary can introduce or modify an item in the retrieval system's index by applying perturbations to a base item (image, audio, document, etc.). The adversary's goal is to maximize the number of query embeddings that rank this item among the top results. They may target either *universal hubness* (dominate retrieval for virtually all queries) or *domain-specific hubness* (dominate queries related to a particular topic). The attacker does *not* necessarily control query inputs; instead, they exploit embedding space geometry so that many legitimate user queries will naturally retrieve the adversarial item.

The attacker has the following capabilities: (1) **Write Access** to insert or modify documents in the vector database; (2) **Query Sampling** to sample queries from the target distribution; and (3) **Embedding Knowledge** of the embedding model used by the target system. The attacker's goal is to inject hub documents that appear in top-k results for a large fraction of user queries, carry malicious payloads, and evade detection by appearing statistically similar to normal content.

In our system, every gallery item is converted into a point in an embedding space, and searches work by retrieving the items located closest to the query. We define a "hub" as a gallery item that exploits this geometry to appear in the top results for a disproportionately large number of searches. Adversarial Hubness Detector's objective is to scan the gallery to identify these crafted items and then either remove them or suppress their ranking to prevent them from dominating the results.

B. Hubness Background

In vector search, we assess a gallery item's impact by quantifying its frequency of appearance in the top results across an extensive array of queries. In high-dimensional data, the distribution is inherently skewed [13]: certain data points, referred to as "hubs," function as attractors and are retrieved significantly more often than the mean, whereas others, termed "anti-hubs," are seldom accessed.

A *hub* is an entity that consistently ranks inside the top-k results for a significant proportion of varied searches. Hubness inherently occurs in high-dimensional embedding spaces, but it can also be deliberately employed to construct *adversarial hubs* that dominate in retrieval. As demonstrated by Zhang *et al.* [19], adversarial hubs can be produced by gradient-based optimization: the algorithm selects a limited number of queries, refines an embedding that corresponds with their common semantic direction, and implements a constrained modification to a gallery item to ensure its retrieval for many queries.

Hubs exhibit several distinguishing characteristics: (1) **Extreme Hub Rates** of 20–50%+ compared to expected rates of 2–5% for normal documents; (2) **Cross-Cluster Spread**—retrieved by queries from many diverse semantic clusters; (3) **Perturbation Stability**—maintaining high retrieval rates even under query perturbations; and (4) **Statistical Anomaly**—hub z-scores typically exceeding 5–10 standard deviations above the median.

C. System Architecture

Adversarial Hubness Detector implements a multi-detector architecture that analyzes vector indices through several complementary lenses. Figure 1 illustrates the overall pipeline.

The detection process operates as follows:

1) **Data Loading**: Load document embeddings, metadata, and vector index from supported backends (FAISS, Pinecone, Qdrant, Weaviate).
2) **Query Sampling**: Generate or sample representative queries from the document distribution.
3) **Retrieval Execution**: Execute k-NN queries using configurable retrieval methods, accumulating hit counts per document.
4) **Detection**: Run multiple detectors analyzing different aspects of adversarial behavior.
5) **Score Fusion**: Combine detector outputs using weighted scoring.
6) **Verdict Assignment**: Classify documents as HIGH, MEDIUM, or LOW risk based on configurable thresholds.

D. Query Sampling Strategy

Effective hubness detection requires representative queries that cover the semantic space of the document corpus. The

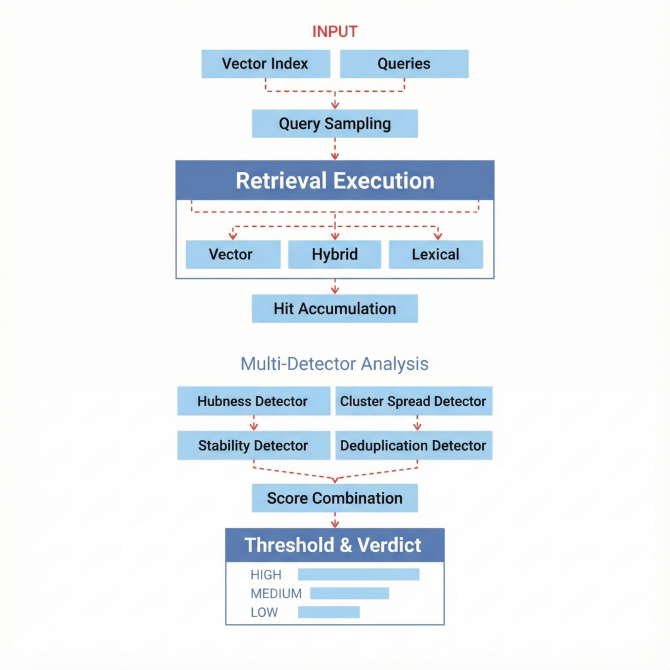

Fig. 1: Adversarial Hubness Detector detection pipeline overview showing the multi-stage process from input to verdict assignment.

query sampling employs a *mixed sampling strategy* combining multiple approaches:

1) **Cluster Centroids**: We apply MiniBatch K-Means clustering to the item embeddings and use cluster centroids as queries, ensuring queries are distributed across semantic regions.
2) **Random Items Sampling**: We randomly sample item embeddings to serve as queries, ensuring queries reflect the actual item distribution.
3) **Real Queries** (when available): For datasets with pre-existing query sets, we incorporate actual user queries to better reflect production query patterns.

This mixed strategy integrates semantic coverage (via centroids) with distributional authenticity (by random sampling), providing detection across varied query patterns while preserving computing efficiency with MiniBatch K-Means.

E. Robust Statistical Framework

A key problem in hubness detection is that hubs function as extreme outliers, distorting conventional mean and variance computations. To address this, we standardize hub rates utilizing the MAD. This rigorous method produces consistent z-scores in the existence of formidable adversary samples, enabling us to accurately identify papers with scores surpassing 5 as potential threats.

F. Weighted Hit Accumulation

A basic hub score quantifies the frequency with which a page appears in the top-k retrieval results across various queries. This strategy, while successful as a baseline, implicitly presumes that all appearances contribute equally to hubness, which is not accurate in actuality. Documents that consistently occupy high-ranking positions or are located at minimal embedding distances demonstrate far more pronounced hub behavior than those that appear infrequently near the threshold.

We developed a *weighted hit accumulation* system to measure this better. Instead of counting every match equally, we give more points to matches that appear at the top of the list or are very similar. This shows that powerful items dominate because they take the best spots, not just because they have been retrieved often.

G. Multi-Detector Scoring

Each detector produces a per-item suspicion score $s_i^{(d)}$, where higher values indicate more anomalous behavior. Since detectors may emit scores on different scales, we optionally apply a per-detector normalization $g_d(\cdot)$ to obtain comparable normalized scores $\tilde{s}_i^{(d)} \in [0, 1]$. We then fuse signals via a weighted sum:

$$s_i^{\text{combined}} = \sum_d \alpha_d \, \tilde{s}_i^{(d)}, \qquad \tilde{s}_i^{(d)} = g_d\left(s_i^{(d)}\right) \tag{1}$$

where α_d are configurable weights controlling each detector's contribution. Default weights are: Hubness z-score (1.0), Domain Hub (0.5), Cross-Modal Penalty (0.5), Cluster Spread (0.3), and Stability (0.2).

H. Verdict Classification

Items are classified into risk levels based on combined scores:

- **HIGH**: $h \geq 99^{\text{th}}$ percentile
- **MEDIUM**: $h \geq 98^{\text{th}}$ percentile
- **LOW**: Below MEDIUM thresholds

These thresholds are tunable and can be adjusted based on operational requirements (higher precision vs. higher recall).

I. Mitigation via Re-ranking

Once a candidate hub is confirmed, the defender can mitigate its impact through removal, quarantine, or a *re-ranking filter* that pushes flagged items down in results. One approach subtracts a penalty from the similarity score of flagged items. Although hubness reduction transforms alone are insufficient against adaptive, domain-specific hubs [19], in combination with explicit detection they can dramatically reduce a hub's visibility with minimal impact on normal results.

IV. Detection Algorithms

Adversarial Hubness Detector implements four complementary detectors, each targeting different aspects of hub behavior. This section details the algorithms and their underlying rationale, Figure 2 illustrates key detection metrics.

Algorithm 1 Hubness Detection

Require: Index $\mathcal{I}$, queries Q, documents D, k
Initialize accumulator A with $|D|$ documents
for each query $q \in Q$ **do**
  neighbors, distances $\leftarrow$ kNN$(\mathcal{I}, q, k)$
  for $r = 1$ to k **do**
    $d_{\text{idx}} \leftarrow$ neighbors$[r]$
    $w \leftarrow w_{\text{rank}}(r) \cdot w_{\text{dist}}(\text{distances}[r])$
    A.add_hit(d_{idx}, w)
  end for
end for
$h \leftarrow A$.compute_hub_rates()
$z, \mu, \sigma \leftarrow$ robust_zscore(h)
return z

A. Hubness Detector

The core hubness detector implements reverse k-NN frequency analysis with bucketed accumulation for efficient processing.

The bucketed accumulator tracks hits per document with optional bucketing by concept or modality, enabling both global and fine-grained analysis from a single pass over queries. The primary output is the robust z-score vector indicating how many "standard deviations" each document's hub rate deviates from the median. Gallery items in the 99^{th} percentile are highly anomalous—for comparison, in a normal distribution, $z = 6$ corresponds to approximately 1 in 500 million probability.

B. Cluster Spread Detector

Hubs are designed to capture queries from multiple semantic regions. The cluster spread detector measures this cross-cluster retrieval pattern.

Algorithm 2 Cluster Spread Detection

Require: Queries Q, query embeddings E_Q, k, n_{clusters}
Cluster queries: $C \leftarrow$ KMeans$(E_Q, n_{\text{clusters}})$
for each query $q_j \in Q$ **do**
  $c_j \leftarrow C$.predict(q_j)
  neighbors $\leftarrow$ kNN$(\mathcal{I}, q_j, k)$
  for each $d_{\text{idx}} \in$ neighbors **do**
    cluster_hits$[d_{\text{idx}}][c_j]$ += 1
  end for
end for
for each document d_i **do**
  $p \leftarrow$ normalize(cluster_hits$[d_i]$)
  $s_i \leftarrow$ entropy$(p)/\log(n_{\text{clusters}})$
end for
return s (normalized entropy scores)

We categorize search queries into semantic clusters utilizing MiniBatch K-Means and assess the extent of a gallery item's distribution among them. By computing a normalized Shannon entropy [15] score reflecting the diversity of hits across query clusters, we differentiate between authentic content and adversarial hubs. A high normalized entropy score (close to 1.0) signifies that the gallery item is distributed evenly across numerous unrelated semantic clusters—indicative of a hub intended for extensive coverage. Normal gallery items exhibit lower entropy, focusing their hits within their specific subject area.

C. Stability Detector

Hubs maintain high retrieval rates under query perturbations due to their central positioning in embedding space. The stability detector exploits this characteristic.

Algorithm 3 Stability Detection

Require: Top candidate documents D_{top}, original queries Q
Initialize stability scores s
for each candidate $d_i \in D_{\text{top}}$ **do**
  original_hits $\leftarrow$ count_hits(d_i, Q)
  perturbed_hits $\leftarrow$ []
  for $p = 1$ to $n_{\text{perturbations}}$ **do**
    $Q' \leftarrow Q + \mathcal{N}(0, \sigma^2 I)$ {Add Gaussian noise}
    $Q' \leftarrow$ normalize(Q')
    perturbed_hits.append(count_hits(d_i, Q'))
  end for
  $s_i \leftarrow$ mean(perturbed_hits)/original_hits
end for
return s

We apply Gaussian perturbations with $\sigma = 0.01$ (configurable) to query embeddings. Gaussian noise provides *isotropic perturbations* that uniformly explore the local embedding neighborhood. Gallery items maintaining high hit rates across perturbations receive higher stability scores. High stability ($s \approx 1$) suggests the document is geometrically central, a hallmark of hubs. Normal gallery items show lower stability as perturbed queries drift to other topics. For efficiency, we only test the top-X candidates (default: top 200) identified by the hubness detector.

D. Deduplication Detector

Attackers may inject multiple near-duplicate hubs to increase coverage or evade single-document thresholds. The deduplication detector identifies such clusters using: (1) **Exact Text-Hash Grouping** when a `text_hash` field is available; (2) **Embedding-Based Near-Duplicate Detection** via nearest-neighbor search with a distance threshold; and (3) **Boilerplate Suppression** to downweight large duplicate clusters (often templated content). Gallery items in duplicate clusters receive adjusted scores based on cluster size.

E. Detector Compatibility

Not all detectors are applicable to all retrieval methods. The Hubness and Deduplication detectors work with vector, hybrid, and lexical retrieval. The Cluster Spread and Stability detectors require semantic query embeddings, making them incompatible with pure lexical search. ADVERSARIAL HUBNESS DETECTOR automatically skips incompatible detectors based on the configured retrieval method.

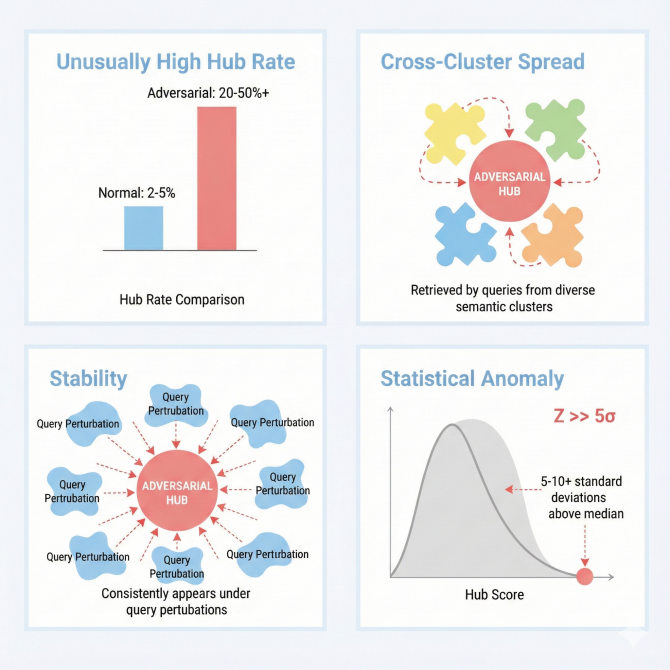

Fig. 2: Key detection metrics and their interpretation: Hub z-score measures statistical anomaly, cluster entropy captures cross-cluster spread, stability indicates robustness to perturbations, and combined scores provide holistic risk assessment.

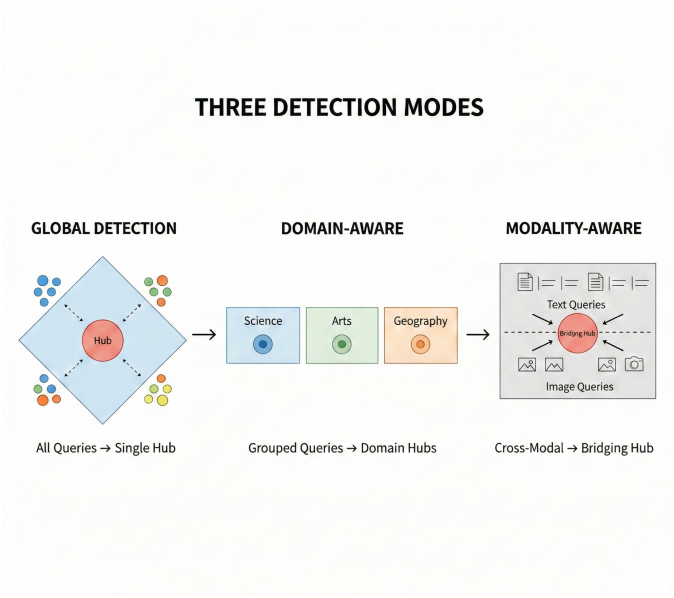

Fig. 3: Three detection modes: Global detection analyzes all queries together, Domain-Aware detection groups queries by semantic domains, and Modality-Aware detection handles cross-modal attacks.

F. Score Interpretation

The combined scoring produces interpretable risk assessments: **High Hub Z-Score** identifies extremely anomalous documents appearing in significantly more queries than statistically expected. **High Cluster Entropy** reflects wide cross-cluster spread across diverse and unrelated topics. **High Stability** indicates robustness to perturbations, where documents maintain retrieval dominance under query variations. The **Combined Score** provides a weighted aggregation for holistic risk assessment.

Default thresholds are calibrated for high precision (minimizing false positives) while maintaining strong recall on known hub patterns.

V. Domain-Aware and Modality-Aware Detection

Standard global hubness detection may miss sophisticated attacks targeting specific semantic domains or exploiting modality boundaries. Adversarial Hubness Detector provides specialized detection modes for these scenarios, as illustrated in Figure 3.

A. Domain-Specific Hub Detection

Domain-specific hub attacks aim to excel in retrieval within a certain semantic category while going undetected at a broader scale. Instead of establishing a universal hub for all inquiries, the attacker focuses influence on a singular notion (e.g., financial advice, medical advice, or legal interpretation), thereby circumventing detectors that depend on aggregate hubness statistics.

Attack Characteristics: Low global hubness avoiding detection by global threshold-based methods; extremely high hubness within a single semantic domain; disproportionate impact on targeted user segments; and applicability across modalities.

Detection Framework: We employ domain-aware hubness analysis where queries are categorized into semantic domains, and hubness is calculated independently for each domain. Domain assignment uses: (1) metadata-based labeling when available, (2) embedding-based clustering via MiniBatch K-Means, or (3) hybrid assignment with clustering fallback.

For each domain, we calculate a topic-specific hubness score—the proportion of queries within that domain for which a retrieval item is included in the top-k results. A retrieval item is flagged as suspicious when its hubness inside a single domain markedly exceeds its hubness across other domains (**contrastive detection**). We also compute a **concentration score** using the Gini coefficient based on the item's hubness distribution among domains:

$$G(d_i) = \frac{n + 1 - 2\sum_{j=1}^{n} \frac{\sum_{k=1}^{j} p_{(k)}}{\sum_{k=1}^{n} p_{(k)}}}{n} \tag{2}$$

where $p_{(k)}$ are the sorted normalized hub rates and n is the number of domains. Values approaching 1 indicate that the majority of hubness is concentrated within a singular domain, typical of focused domain-specific attacks.

Domain-Aware Scoring: For each item, the scoring module produces: (1) maximum domain-level hub z-score across all domains; (2) dominant domain identifier; and (3) hubness concentration score. This approach can identify globally benign but locally dominant items.

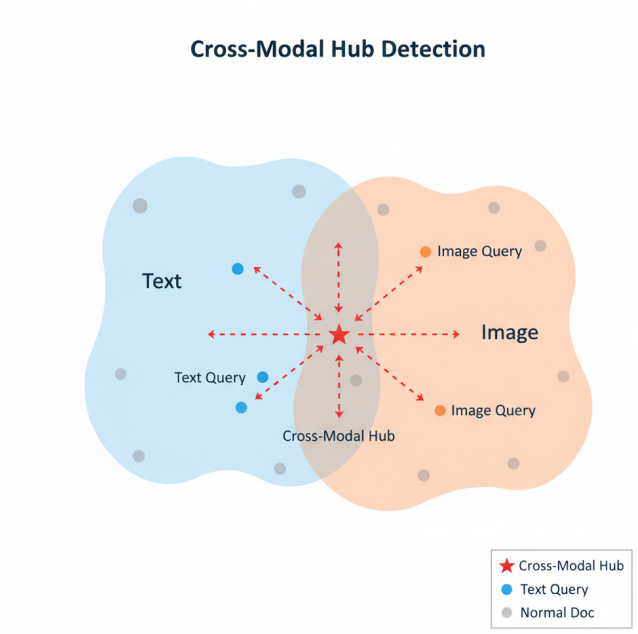

Fig. 4: Cross-modal hub detection: A hub (red star) positioned at the intersection of text and image modalities appears in top-k results for queries from both modalities, exploiting the modality boundary.

B. Cross-Modal Hub Detection

In multimodal retrieval systems (e.g., text–image or text–audio), adversarial hubs can exploit modality boundaries. A text-based retrieval item may be crafted to disproportionately appear in response to image-oriented queries, or vice versa. Such cross-modal dominance allows attackers to manipulate retrieval outcomes while evading detectors that operate within a single modality.

Attack Characteristics: The modality of the retrieval item differs from the dominant modality of the triggering queries; high retrieval frequency in cross-modal query settings; and evasion of single-modality hub detection mechanisms.

Detection Framework: If modality metadata is present for both queries and retrieved items, ADVERSARIAL HUBNESS DETECTOR tracks *cross-modal hits*—when the query modality differs from that of the retrieved item. We calculate a *cross-modal hub rate* for each item, normalized by the total number of searches, then compute a robust z-score over these rates to identify items receiving disproportionately many cross-modal hits. Items with high cross-modal z-scores receive an additive penalty term weighted in the final risk aggregate.

Figure 4 illustrates how cross-modal hubs exploit modality boundaries.

Modality-Aware Scoring: For each item, the scoring module produces: (1) cross-modal retrieval ratio; (2) dominant query modality; and (3) modality-adjusted hub score. The final score integrates global hubness with cross-modal behavior to identify items exploiting modality boundaries even when they appear benign within any single modality.

VI. RETRIEVAL AND RERANKING INTEGRATION

Effective hubness detection must operate within the retrieval and ranking pipelines used in production systems. ADVERSARIAL HUBNESS DETECTOR seamlessly connects with popular retrieval paradigms and reranking methodologies, aligning detections with user observations.

A. Supported Retrieval Methods

Vector Similarity Search: The standard retrieval method employs dense embeddings and approximate nearest neighbor search to obtain the top-k most similar items. Similarity is calculated using cosine similarity or inner product. ADVERSARIAL HUBNESS DETECTOR integrates with widely used vector databases including FAISS, Pinecone, Qdrant, and Weaviate.

Hybrid Search: Hybrid retrieval integrates semantic (dense) similarity with lexical (sparse) matching, enabling systems to reconcile conceptual relevance with keyword overlap. This is particularly relevant for hubness detection because adversarial items optimized for semantic similarity may not dominate lexical search, and vice versa—enabling detectors to analyze whether an item behaves as a hub in one signal or across both.

Lexical Search: Lexical-only retrieval depends solely on keyword-centric scoring methods like BM25 or TF-IDF. This mode is beneficial for systems that emphasize keyword search or for isolating purely lexical manipulation techniques like keyword stuffing. In lexical-only pipelines, detectors reliant on embeddings (cluster spread, stability) are automatically deactivated.

B. Reranking Support

Many production systems apply a second-stage reranking model to refine initial retrieval results. Reranking typically operates on a larger candidate set returned by the retriever and produces a final, user-facing ranking. ADVERSARIAL HUBNESS DETECTOR facilitates customizable reranking workflows and assesses hubness according to post-reranking results, demonstrating that detection corresponds with the items finally presented to users.

VII. EVALUATION

We evaluate ADVERSARIAL HUBNESS DETECTOR under the threat model outlined in Section III, quantifying its efficacy in identifying adversarial *hubs*—entities engineered to monopolize nearest-neighbor retrieval for numerous legitimate inquiries.

A. Benchmark Construction

Following standard adversarial ML evaluation methodology, we implemented the attack technique from Zhang et al. [19] to generate adversarial hubs, then evaluated ADVERSARIAL HUBNESS DETECTOR's ability to detect them. **Critically**, the attack and detection algorithms are independent: hub generation uses only embedding-space gradient optimization with no knowledge of ADVERSARIAL HUBNESS DETECTOR's

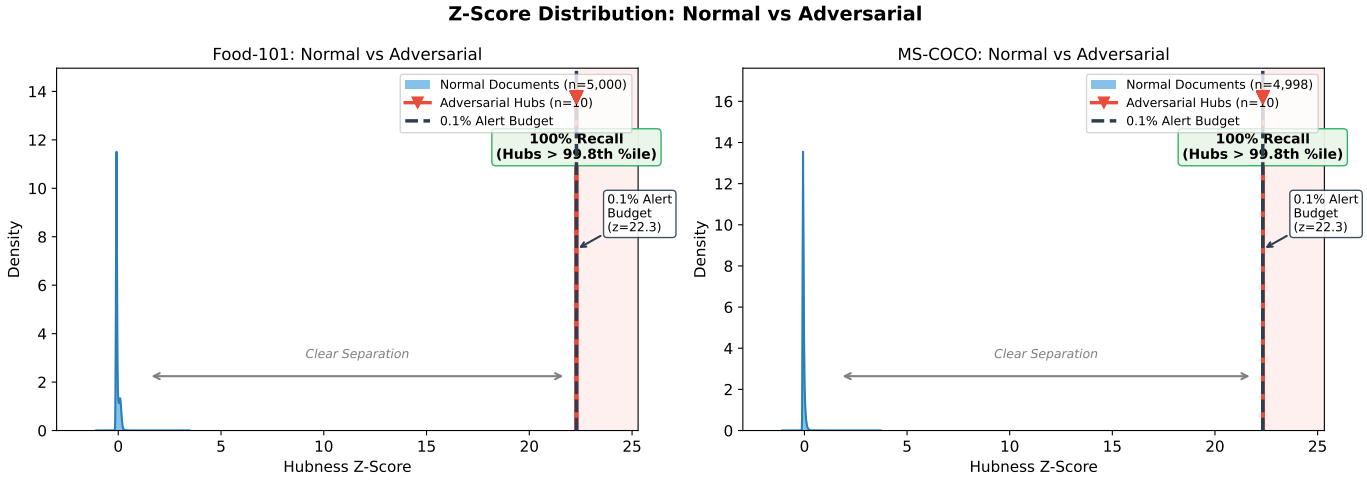

Fig. 5: Hubness score distribution for normal documents vs. planted adversarial hubs on Food-101 and MS-COCO. Normal items cluster near zero while adversarial hubs are extreme outliers (z-score >20). At a 0.1% alert budget, all planted hubs rank above the 99.8th percentile, achieving 100% recall with minimal false positives.

statistical detection methods. This reflects a realistic threat model where attackers optimize embeddings without access to the defender's detection logic.

The Zhang methodology optimizes a hub embedding h to maximize similarity to a target query set $\mathcal{T}$:

$$h^* = \arg\max_{\|h\|=1} \sum_{q \in \mathcal{T}} \text{sim}(h, q) \tag{3}$$

using momentum-based gradient ascent (μ=0.9, η=0.12) with cosine temperature annealing. We evaluate two attack variants: **Universal** (random diverse queries; hub generalizes broadly) and **Domain-targeted** (domain-specific queries with repulsion term λ_{neg}=3.0 penalizing out-of-domain similarity).

To validate supplementary detection capabilities, we also utilize **centroid-based hubs** constructed as weighted averages of diverse document embeddings. Unlike gradient-optimized hubs, centroid hubs have artificial geometric placements and are sensitive to slight perturbations (denoted as *brittle* hubs).

B. Experimental Configuration

Table I summarizes the benchmarks. Image datasets use CLIP ViT-B/32 embeddings (512 dimensions), while FiQA uses Qwen3-0.6B embeddings (1024 dimensions). Universal hubs are optimized over 200 target queries for 1,000 gradient steps.

TABLE I: Evaluation Benchmark Statistics

Benchmark	Docs	Queries	Hubs	Dim	Domains
Food-101 [2]	5,010	25,000	10	512	101
MS-COCO [7]	5,008	25,000	10	512	40
FiQA [9]	5,005	648	5	1024	—

Scanner configuration uses k=20 nearest neighbors, sampling 10,000 queries per scan via a mixed strategy (50% random documents, 50% cluster centroids).

C. Evaluation Protocol

We assess within a predetermined **alert budget**: with N gallery items and a budget fraction b, we designate only the top-K=$\lceil b \cdot N \rceil$ highest-scoring items for manual examination. For example, b=0.2% on a ∼5,000-item gallery corresponds to reviewing the top 10 flagged items. We report *recall* (fraction of adversarial hubs in top-K) and *precision* (fraction of top-K that are true hubs).

D. Detection Results

Table II reports detection performance across datasets. Results are consistent, demonstrating that Zhang-style optimization yields similar statistical signatures regardless of the underlying domain or modality.

TABLE II: Detection Performance (Full Configuration)

Dataset	Attack	Budget	K	Prec.	Recall
Food-101	Universal	0.2%	10	90%	90%
	Universal	0.4%	20	50%	100%
MS-COCO	Universal	0.2%	10	90%	90%
	Universal	0.4%	20	50%	100%
FiQA	Universal	0.1%	5	100%	100%

Key findings: With a 0.2% alert budget (K=10), ADVERSARIAL HUBNESS DETECTOR accomplishes **90% recall**, successfully identifying 9 out of 10 planted hubs. Expanding to K=2H achieves **100% recall** across tested datasets.

E. Ablation Study: Detector Contributions

We performed ablation tests to assess the contribution of each detector to universal hub detection. On Food-101, the Hubness-only configuration achieves 40% recall at 0.1% budget and 80% at 0.2%. Adding the **Cluster Spread detector**

produces a **10–20 percentage point increase**: 50% at 0.1% and 100% at 0.2%.

This enhancement arises because universal hubs, optimized to attract diverse queries, demonstrate significant cluster dispersal. Universal hubs achieve a cluster spread of 0.92, while domain-targeted hubs obtain only 0.06—a **14× difference**. This explains why cluster spread is effective for universal attacks but not domain-targeted ones, which require domain-scoped scanning.

F. Stability Detector: Catching Brittle Hubs

To validate the **stability detector's** coverage, we assess centroid-based hubs. Unlike Zhang's gradient optimization, centroid hubs are formed as weighted averages of document embeddings, making them naturally *fragile*.

Stability is evaluated by introducing Gaussian noise (σ=0.15) to query embeddings. Zhang hubs show minimal hit change (<4%), while brittle hubs show up to 68% hit change **20× more unstable**. This verifies the stability detector's role: it detects centroid-based attacks that avoid the cluster spread detector, while Zhang-style hubs are detected by hubness and cluster spread signals. The modular detector architecture provides *defense in depth* against multiple attack techniques.

G. Domain-Scoped Scanning

Domain-targeted attacks present a distinct challenge: under global scanning with tight alert budgets, they can be pushed out by items with higher global hubness signals. We establish a benchmark where 15 benign universal hubs compete against 10 adversarial domain-targeted hubs.

During global scanning at K=H, **recall drops to 0%**, i.e. all top-K positions are held by the benign universal hubs. However, these domain hubs are not invisible: they achieve an AUC-ROC of 0.995. The issue is *budget saturation*, not detection failure. ADVERSARIAL HUBNESS DETECTOR addresses this through **domain-scoped scanning**: filtering the query set to a specific domain before computing hubness statistics. When scoped to the target domain, during testing, domain-targeted hubs achieve **100% recall**. This demonstrates the importance of layered defense: global scanning catches universal attacks, while domain-scoped investigation catches targeted attacks.

H. Cross-Domain Generalization and Scaling Limits

The results confirm that adversarial hubness is **modality-agnostic**: FiQA achieves **AUC-ROC of 1.0**, results in perfect separation with **100% recall and 100% precision at 0.1% alert budget**.

To characterize the fundamental limits of statistical hub detection, we evaluate across varying adversarial corpus fractions (0.1% to 30%) on five domains: Food-101, MS-COCO (CLIP), FiQA (Qwen3), Code (CodeSearchNet with CodeBERT), Medical (PubMed abstracts with PubMedBERT). Table III reports results.

TABLE III: AUC-ROC by Adversarial Corpus Fraction (Hubness Z-Score Detector)

Domain	≤2%	5%	10%	20%	30%
Food-101	1.00	1.00	1.00	0.99	0.96
MS-COCO	1.00	1.00	1.00	1.00	1.00
FiQA	1.00	0.99	0.95	0.87	0.80
CodeSearchNet	1.00	0.92	0.76	0.67	0.62
PubMed	1.00	0.96	0.87	0.75	0.69
Average	**1.00**	**0.97**	**0.92**	**0.86**	**0.82**

Key finding: Detection is perfect (AUC=1.0) when adversarial content comprises ≤2% of the corpus, and remains highly effective (>0.9 AUC) up to 5%. Detection degrades at higher fractions as adversarial items shift the distribution baseline. However, attacks of 10–30% corpus fraction are unlikely threat scenarios that can be detected using simpler measures (corpus monitoring, ingestion rate alarms).

I. Detection Signal Analysis

Figure 5 shows the hubness score distribution for normal items versus planted adversarial hubs. Normal items aggregate around zero, but adversarial hubs acquire z-scores of more than 20 standard deviations, making them significant outliers by any measure. During testing, at a 0.1% alert budget (99.9$^{\text{th}}$ percentile threshold), all planted hubs exceed the 99.8$^{\text{th}}$ percentile, achieving **100% detection with few false positives**. This percentile-based technique adapts across datasets and does not require manual threshold setting.

VIII. PRODUCTION VALIDATION: NATURAL HUB AUDIT

A fundamental concern for production deployment is establishing a baseline for the system's noise level: **what is the false positive rate of ADVERSARIAL HUBNESS DETECTOR in a clean environment with no active adversary?** To address this at a realistic web scale, we performed a natural hub audit on the MS MARCO [11] passage retrieval corpus, an industry-standard dataset including 8.8 million documents sourced from actual Bing search results. We assessed a representative subset of **1 million passages**, proving validation at a production scale. We executed ADVERSARIAL HUBNESS DETECTOR using the same configuration employed for adversarial detection (mixed query sampling, k=20, complete multi-detector ensemble) and evaluated detection scores at 0.1% and 0.2% alert budgets to delineate operational workload.

Score Separation. Table IV demonstrates clear separation between clean and adversarial distributions. The 99$^{\text{th}}$ percentile of clean data scored 2.3, while planted adversarial hubs from our evaluation (Section VII) scored 13-17, resulting in a separation factor of **5.8×**. The 99.9$^{\text{th}}$ percentile of clean data (5.0) remains **2.7× lower** than adversarial hubs. This separation establishes distinct operational thresholds: score >10 demands investigation (adversarial range); score <5 indicates the clean baseline (99.9$^{\text{th}}$ percentile).

Operational Workload. At a 0.1% alert budget, operators assess the top 1,000 highest-scoring documents per 1 million corpus. All highlighted passages received scores below 10.0, remaining beneath the adversarial range of 13–17. The 99$^{\text{th}}$

TABLE IV: Score Distribution: Clean vs. Adversarial (1M Scale)

Metric	Clean MS MARCO	Adversarial	Separation
99^{th} pct.	2.3	13–17	**5.8**×
99.9^{th} pct.	5.0	13–17	2.7×

percentile threshold (2.3) provides **5.8**× **separation** from adversarial content, while the 99.9^{th} percentile (5.0) maintains **2.7**× **separation**. Increasing the budget to 0.2% (2,000 documents) preserved the same separation, with all scores remaining within the clean range. These represent expected false positives, documents scoring high relative to the clean baseline but far below true adversarial thresholds. The distinct score separation enables operators to implement an intuitive threshold (e.g., score >10) to quickly distinguish adversarial content from natural false positives, thus decreasing review time while preserving high detection sensitivity. This web-scale validation on 1 million real documents confirms operational feasibility for production RAG systems.

IX. Conclusion

We have presented an in-depth study of hubness in multimodal retrieval and a practical defense strategy. Hubs represent a potent attack vector: by exploiting the geometric properties of high-dimensional embeddings, a single malicious item can hijack retrieval results for a vast number of queries simultaneously [19]. Adversarial Hubness Detector is a comprehensive detection system addressing a critical security gap in the RAG ecosystem, where vector databases are increasingly targeted by sophisticated poisoning attacks.

Our multi-detector architecture combines statistical hubness detection, cluster spread analysis, stability testing, and domain/modality-aware detection. Evaluation demonstrates 90% recall at 0.2% alert budget and 100% recall at $K{=}2H$, with adversarial hubs ranking above the 99.8^{th} percentile. In realistic threat models where adversaries introduce a limited number of high-impact hubs ($\leq$2% of the corpus), the hubness z-score detector alone achieves perfect detection.

A. Limitations

Current limitations include:

- **Query Distribution Dependence**: Detection accuracy relies on representative query sampling or queries from production.
- **Adaptive Adversaries**: Attackers aware of detection mechanisms may attempt evasion.
- **Low-Volume Attacks**: Attacks with minimal hub rates may evade statistical detection.
- **Statistical Limits**: The hubness z-score detector's effectiveness degrades when adversarial content exceeds 10% of the corpus, requiring corpus monitoring for large-scale attacks.

B. Future Directions

To improve Adversarial Hubness Detector, future work will include:

- Real-Time Detection: Adjusting Adversarial Hubness Detector to flag potential hubs during the indexing process, by pre-calculating hubness score against set of queries.
- Intent Scanning: Extending Adversarial Hubness Detector with an intent scanner to search for malicious content inside flagged items (e.g. prompt injections or hidden content).
- Adaptive Adversaries: Explore how Adversarial Hubness Detector performs against attackers that adapt their planted items to bypass hubness-based detection, a potential strategy can be planting items that will disrupt such attempts.

C. Open Source Release

We release Adversarial Hubness Detector as open-source software to enable security auditing of production RAG systems. The release includes the complete implementation, adapters for popular vector databases, all detection algorithms described in Section IV, and reproduction scripts for our benchmarks.

Repository: https://github.com/cisco-ai-defense/adversarial-hubness-detector

Appendix

This work presents Adversarial Hubness Detector, a defensive tool designed to detect adversarial hubness attacks in RAG systems. We address the following ethical considerations:

Dual-Use Concerns. While we use hub creation techniques from previous works to contextualize our defense, we do not introduce any new attack strategies. Our primary focus is on detection and mitigation. The attack strategies we assess have already been publically described in peer-reviewed literature.

Responsible Disclosure. The real-world vulnerabilities discussed (Microsoft Copilot, GeminiJack) were discovered and disclosed by independent security researchers prior to our work. We cite these incidents to motivate the need for hubness detection, not to enable new attacks.

Benchmark Construction. Our evaluation benchmarks use publicly available datasets (Food-101, MS-COCO, FiQA) with synthetically planted adversarial hubs. No real user data or production systems were involved in our experiments.

Intended Use. Adversarial Hubness Detector is designed for security practitioners to audit and protect RAG deployments. We encourage responsible use for defensive purposes and discourage any application that could harm users or systems.

We are committed to open and reproducible research:

Open-Source Software. Adversarial Hubness Detector is released under the Apache 2.0 license at https://github.com/cisco-ai-defense/adversarial-hubness-detector. The repository includes the complete detection

framework, all detector implementations, and adapter interfaces for major vector databases (FAISS, Pinecone, Qdrant, Weaviate).

Benchmark Datasets. Our evaluation benchmarks, including the adversarial hub generation scripts following Zhang et al. [19], are included in the repository under `benchmarks/`. This enables researchers to reproduce our results and evaluate new detection methods.

Evaluation Scripts. All scripts used to generate the results in this paper are provided, including hub generation, detection execution, and metric computation. Configuration files for each experiment are included.

Documentation. Comprehensive documentation covers installation, configuration, API usage, and extension development. Example notebooks demonstrate common use cases.

Reproducibility. Random seeds are fixed for all stochastic components. Hardware requirements and expected runtimes are documented. We provide pre-computed embeddings for the benchmark datasets to reduce computational barriers to reproduction.

References

[1] Bansal, A. Optimizing rag with hybrid search and contextual chunking. *Journal of Engineering and Applied Sciences Technology. SRC/JEAST-E114. DOI: doi. org/10.47363/JEAST/2023 (5) E114 J Eng App Sci Technol 5*, 4 (2023), 2–5.

[2] Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *European conference on computer vision* (2014), Springer, pp. 446–461.

[3] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *arXiv preprint arXiv:2302.12173* (2023).

[4] Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. Word translation without parallel data. In *International conference on learning representations* (2018).

[5] Levi, S. GeminiJack: The Google Gemini zero-click vulnerability leaked Gmail, Calendar and Docs data. https://noma.security/blog/geminijack-google-gemini-zero-click-vulnerability/, Dec. 2025. Noma Security.

[6] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems 33* (2020), 9459–9474.

[7] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision* (2014), Springer, pp. 740–755.

[8] Lin, Z., Wang, Z., Qian, T., Mu, P., Chan, S., and Bai, C. Neighborretr: Balancing hub centrality in cross-modal retrieval. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 9263–9273.

[9] Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., and Balahur, A. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018* (2018), pp. 1941–1942.

[10] Nassi, B., Schneier, B., and Brodt, O. The promptware kill chain: How prompt injections gradually evolved into a multi-step malware. *arXiv preprint arXiv:2601.09625* (2026).

[11] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. Ms marco: A human-generated machine reading comprehension dataset.

[12] Nogueira, R., and Cho, K. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085* (2019).

[13] Radovanovic, M., Nanopoulos, A., and Ivanovic, M. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research 11*, sept (2010), 2487–2531.

[14] Schnitzer, D., Flexer, A., Schedl, M., and Widmer, G. Local and global scaling reduce hubs in space. *The Journal of Machine Learning Research 13*, 1 (2012), 2871–2902.

[15] Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal 27*, 3 (1948), 379–423.

[16] Wang, Y., Jian, X., and Xue, B. Balance act: Mitigating hubness in cross-modal retrieval with query and gallery banks. *arXiv preprint arXiv:2310.11612* (2023).

[17] Wu, Y., Zhang, H., and Huang, H. Retrievalguard: Provably robust 1-nearest neighbor image retrieval. In *International Conference on Machine Learning* (2022), PMLR, pp. 24266–24279.

[18] Zenity Security Research. Zero-click attacks: When TTPs resurface across platforms. https://zenity.io/blog/security/zero-click-attacks-when-ttps-resurface-across-platforms, 2025. Accessed: 2025-01-07.

[19] Zhang, T., Suya, F., Jha, R., Zhang, C., and Shmatikov, V. Adversarial hubness in multi-modal retrieval. *arXiv preprint arXiv:2412.14113* (2024).