

---

# TRANSFORM BEFORE YOU QUERY: A PRIVACY-PRESERVING APPROACH FOR VECTOR RETRIEVAL WITH EMBEDDING SPACE ALIGNMENT

---

A PREPRINT

**Ruiqi He\***

Nankai University

heruiqi@mail.nankai.edu.cn

**Zekun Fei\***

Nankai University

feizekun@mail.nankai.edu.cn

**Jiaqi Li**

Nankai University

lijiaqi@mail.nankai.edu.cn

**Xinyuan Zhu**

Nankai University

zhuxinyuan@mail.nankai.edu.cn

**Biao Yi**

Nankai University

yibiao@mail.nankai.edu.cn

**Siyi Lv<sup>†</sup>**

Nankai University

lvsiyi@nankai.edu.cn

**Weijie Liu**

Nankai University

weijieliu@nankai.edu.cn

**Zheli Liu**

Nankai University

liuzheli@nankai.edu.cn

## ABSTRACT

Vector Database (VDB) can efficiently index and search high-dimensional vector embeddings from unstructured data, crucially enabling fast semantic similarity search essential for modern AI applications like generative AI and recommendation systems. Since current VDB service providers predominantly use proprietary black-box models, users are forced to expose raw query text to them via API in exchange for the vector retrieval services. Consequently, if query text involves confidential records from finance or healthcare domains, this mechanism inevitably leads to critical leakage of user's sensitive information. To address this issue, we introduce STEER (**S**ecure **T**ransformed **E**mbedding **v**ector **R**etrieval), a private vector retrieval framework that leverages the alignment relationship between the semantic spaces of different embedding models to derive approximate embeddings for the query text. STEER performs the retrieval using the approximate embeddings within the original VDB and requires no modifications to the server side. Our theoretical and experimental analyses demonstrate that STEER effectively safeguards query text privacy while maintaining the retrieval accuracy. Even though approximate embeddings are approximations of the embeddings from proprietary models, they still prevent the providers from recovering the query text through Embedding Inversion Attacks (EIAs). Extensive experimental results show that Recall@100 of STEER can basically achieve a decrease of less than 5%. Furthermore, even when searching within a text corpus of millions of entries, STEER achieves a Recall@20 accuracy 20% higher than current baselines. Our code is available at <https://anonymous.4open.science/r/STEER>.

## 1 Introduction

Vector Database (VDB) now serves as the critical infrastructure enabling the efficient semantic search in diverse Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Peng et al., 2024) systems, which fundamentally empowers Large Language Models (LLMs) with real-time knowledge retrieval capabilities. However, current manufacturers (e.g., AWS (Enevoldsen et al., 2025)) primarily rely on their own, black-box embedding models to power their VDB services. This approach restricts users to accessing these models solely via API for retrieval tasks. Thus, users in sectors with sensitive data, such as financial or healthcare institutions, are compelled to transmit confidential proprietary information through exposed query texts (Mao et al., 2024; Xia et al., 2024). As shown in Figure 1, user queries get encoded via proprietary models into embeddings for top-k retrieval, inevitably exposing sensitive text to service providers, creating critical privacy leakage risks. To address such risks, mitigation approaches can be categorized into encryption-based solutions and non-encryption schemes.

---

\* Equal contribution; <sup>†</sup> Corresponding author

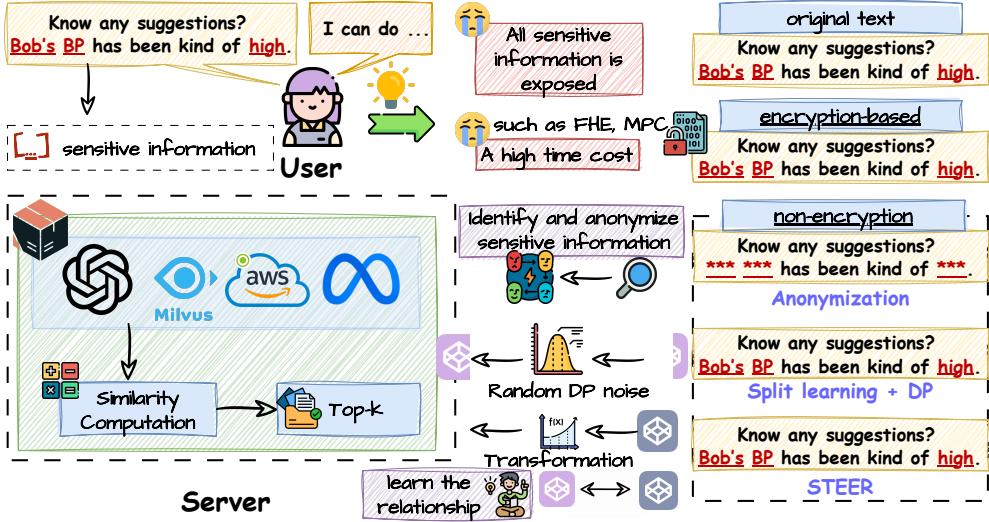


Figure 1: Overview of privacy risks, representative existing solutions, and our proposed method in vector retrieval.

Encryption-based solutions (Zyskind et al., 2024; Henzinger et al., 2023) secretly obtain the query embeddings and execute retrieval with guaranteed security. However, the schemes using Fully Homomorphic Encryption (FHE) (Bian et al., 2023) have to take nearly 2 days to complete a query over 4,096 data items, rendering them impractical for real-world deployment.

Non-encryption schemes adopt anonymization (Yu et al., 2023b) and Split Learning (Mai et al., 2024) to avoid the sensitive information being obtained by the service provider. For anonymization, detected sensitive information within the query text is desensitized. For split learning, embedding models are partitioned and deployed on user-side and server-side. Users compute local Intermediate Results (IRs), while servers process IRs into final embeddings, requiring Differential Privacy (DP) noise injection into IRs during transmission to resist Embedding Inversion Attacks (EIAs).

**We reveal that current feasible schemes, while preserving the privacy of user’s query text, severely compromise its semantics and thereby substantially degrade retrieval performance.** Retrieval employing anonymization techniques obscures critical information in query text, thereby compromising retrieval performance. Meanwhile, the key challenge in solutions based on split learning and DP lies in balancing utility and privacy. The random noise introduced by DP creates semantic-agnostic perturbations in query embeddings, consequently degrading retrieval performance. Furthermore, split learning requires the modification of the server, which is actually difficult to deploy. Our goal is to design a solution with high efficiency and retrieval accuracy.

In this paper, leveraging the insight that embedding spaces (semantic spaces) of different embedding models exhibit latent alignment (Sutskever et al., 2014; Chen et al., 2025), we propose STEER (Secure Transformed Embedding vEctor Retrieval), a novel privacy-preserving retrieval framework. **To guarantee that query text remains within private domain**, STEER deploys a suitable open-source embedding model locally without any training, considering user’s hardware constraints. The query text can be converted to a local embedding. **To enable the utilization of local embedding for subsequent retrieval**, STEER computes the mapping function between the local embedding space and the server-side embedding space using common text without sensitive information. Thus, the local embedding can be transformed to approximate the query text representation in VDB’s embedding space. Finally, the approximate embedding is used for retrieval, ensuring the service provider can’t access the original query text.

**STEER achieves resistance against EIAs through the deviation between approximate embedding and the original embedding derived from server-side model. Moreover, this deviation achieved via transformation, distinct from anonymization and random DP noise, effectively leverages the prior knowledge of the semantic alignment to maintain the retrieval quality.** Our main contributions are as follows.

- We reveal that inherent relationship between embedding spaces can be used to implement private vector retrieval. After transformation, approximate embeddings maintain analogous semantics and consistent relative positional relationships, thereby ensuring retrieval accuracy.

- We propose STEER, a novel privacy-preserving vector retrieval framework. Extensive experiments demonstrate the effectiveness of STEER, basically achieving an accuracy (Recall@100) decrease of less than 5%.
- STEER establishes the mapping function during a single transformation phase, requiring no modification to the server-side embedding models. This approach offers practical advantages for deployment.

## 2 Related Work

### 2.1 Vector Retrieval

Vector retrieval (Lee et al., 2023) enables efficient search of similar items from massive datasets, like finding relevant text, images, or products by their semantics rather than exact keywords. Its core workflow involves: (1) converting data into numerical vector representations (embeddings), (2) comparing the input query vector to find nearest neighbors based on similarity metrics like cosine distance. Vector retrieval empowers applications such as Retrieval Augmented Generation (RAG) (Lewis et al., 2020) for enhancing the generative capabilities of LLMs (Fan et al., 2024; Ram et al., 2023; Shi et al., 2024). The flexibility of vector retrieval has enabled widespread applications across specialized domains, including legal consultation, recommendation, and other systems (Panagoulias et al., 2024).

### 2.2 Privacy Protection for Retrieval

Regarding the query text privacy, solutions can be mainly divided into four directions: cryptography, Trusted Execution Environment (TEE), anonymization, and Split Learning.

The efficiency of cryptography-based schemes (Lu et al., 2023; Li et al., 2023; Chen et al., 2022) is severely affected by ciphertext computation. TEE-based schemes (South et al., 2023) highly rely on the security of hardware itself and still require the architecture transformation of the VDB service provider. Anonymization-based schemes (Yu et al., 2023b; Cheng et al., 2024) will desensitize important information in the query text using k-anonymity or DP. For the schemes based on Split Learning, due to the influence of Embedding Inversion Attacks (EIAs) (Morris et al., 2023; Chen et al., 2024), the embedding can be successfully converted to the original text. Therefore, a relatively large DP noise is needed to make the text embedding resistant to EIAs. The way of adding perturbations in DP, which is semantically independent, will have a significant impact on the retrieval accuracy.

Our work is different from previous schemes. It uses a local model for approximate retrieval without any fine-tuning. It does not rely on cryptography and can efficiently complete the retrieval task within an acceptable loss of accuracy.

## 3 Methodology

### 3.1 Threat Model

Based on the real-world scenarios and prior work (Mai et al., 2024), we conduct a detailed analysis of the threat model during the vector retrieval process, including the objectives, knowledge, and capabilities of VDB service providers. Specifically, we argue that the service providers are honest-but-curious: they faithfully execute user's retrieval operations while attempting to infer user's sensitive query text.

- **Objectives:** The service provider, without altering the retrieval workflow, seeks to obtain sensitive information within the query text by directly extracting the query text, or applying EIAs, etc.
- **Knowledge:** The service provider knows only the architecture of the server-side embedding model and can obtain (text, embedding) pairs via this model, as well as the content of user queries. The specific content of a user query may be either text or an embedding. Users can pre-process sensitive information before transmitting the final query, while the service provider has no knowledge of how the user generates this final query content.
- **Capabilities:** The service provider can strive to reconstruct the original user query based on its existing knowledge. For example, it could train an embedding-to-text model utilizing EIAs to recover the query text from query embeddings as faithfully as possible.

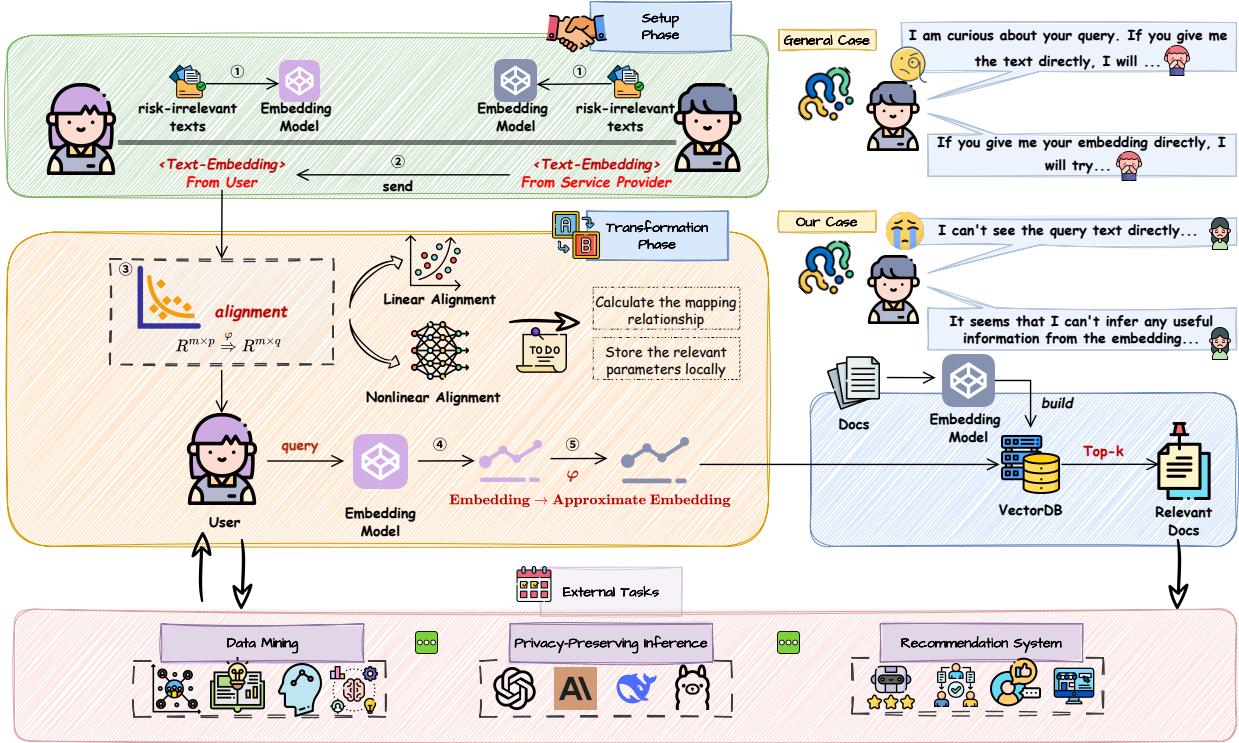


Figure 2: Overview of our STEER framework. It includes a setup phase that prepares the necessary data for learning a semantic mapping between local and server-side embedding spaces, and a transformation phase that computes this mapping and enables private retrieval using approximate embeddings.

### 3.2 Design and Workflow of STEER

#### 3.2.1 Overview of STEER

Our framework design is illustrated in Figure 2, and it consists of two main phases: the Setup Phase and the Transformation Phase.

- **Setup Phase:** A set of risk-irrelevant and general-purpose texts is selected. Both the client and the service provider independently generate embedding vectors for these texts using their respective embedding models, thereby forming two distinct embedding spaces. This phase prepares all the data necessary for learning the alignment between embedding spaces, with all relevant data stored locally on the client side.
- **Transformation Phase:** The client computes a transformation that aligns the local embedding space with that of the server, based on the embedding pairs prepared during the setup phase. **This transformation is calculated only once, entirely on the client side, and securely stored locally for future use.** When the client initiates a query, it is first encoded into an embedding using the local embedding model. The stored transformation is then applied to approximate the embedding into the service provider's embedding space, which enables the subsequent retrieval process to be performed in the target space.

Although STEER is motivated by the privacy concerns in RAG scenarios, it is not limited to retrieval-augmented generation. Since it performs protection directly at the vector retrieval level, it can be seamlessly integrated into a wide range of downstream applications that rely on vector-based similarity search, such as private recommendation systems, data mining pipelines, or any LLM-based inference tasks. As long as the system depends on embedding retrieval, our framework can provide privacy-preserving capability without modifying the server infrastructure.

#### 3.2.2 Workflow of STEER

We decompose the two-phase system framework into several fundamental steps, as illustrated in Figure 2:

1.  $\text{Gen}(\text{texts}, \text{model}) \rightarrow \text{embs}$ . A risk-irrelevant, general-purpose text set is selected. Both the user and the service provider independently generate embedding vectors for these texts using their respective embedding models.
2.  $\text{SendEmbs}(\text{user}, \text{embs})$ . The service provider transmits all generated embeddings  $\text{embs}$  to the user.
3.  $\text{CalculateRel}(\text{embs}_1, \text{embs}_2) \rightarrow \varphi$ . Upon receiving embeddings of risk-irrelevant text from the service provider, the user aligns the local and remote embedding spaces by computing a transformation  $\varphi$ . This transformation  $\varphi$  is kept private and stored locally by the user.
4.  $\text{QueryGen}(\text{query}, \text{model}) \rightarrow \text{emb}$ . When formulating a query, the user encodes it into an embedding using the locally deployed model.
5.  $\text{Trans}(\text{emb}, \varphi) \rightarrow \hat{\text{emb}}$ . The stored transformation  $\varphi$  is applied to the query embedding, yielding an approximate mapping  $\varphi(\text{emb}) \rightarrow \hat{\text{emb}}$  in the service provider's embedding space. The transformed embedding  $\hat{\text{emb}}$  is then sent to the service provider, which retrieves the top-k relevant documents using  $\hat{\text{emb}}$ .

### 3.3 Embedding Space Alignment

As described in the Transformation Phase of STEER, the user generates a query embedding using a local encoder and then applies a learned mapping to approximate the embedding space of the VDB server. This part details how to construct such a mapping function  $\varphi$ , which aligns the local embedding space to the server-side embedding space.

Let us consider a set of textual data denoted as  $\mathbf{X}$ , where  $\mathbf{X} \subseteq D$  is a dataset consisting of  $m \in \mathbb{N}$  samples. Suppose we have two distinct embedding models with encoders denoted as  $\text{enc}_L$  and  $\text{enc}_S$ . The corresponding embedding matrices generated by these models are  $\mathbf{E}_L = \text{enc}_L(\mathbf{X}) \in \mathbb{R}^{m \times p}$  and  $\mathbf{E}_S = \text{enc}_S(\mathbf{X}) \in \mathbb{R}^{m \times q}$ , where  $p$  and  $q$  represent the dimensions of the embeddings from each model, respectively. The goal is to learn a mapping function  $\varphi$  such that the transformed space of  $\mathbf{E}_L$  approximates  $\mathbf{E}_S$ :

$$\varphi(\mathbf{E}_L) \approx \mathbf{E}_S, \quad (1)$$

#### 3.3.1 Linear Alignment

Due to the discrete property of the vector feature space, linear transformations tend to preserve a certain degree of relative position consistency. Hence, we aim to find a linear alignment matrix  $\mathbf{A}$  such that the transformed embedding  $\mathbf{A}\mathbf{E}_L$  approximates the target embedding  $\mathbf{E}_S$ :

$$\mathbf{A}\mathbf{E}_L \approx \mathbf{E}_S, \quad (2)$$

which can be formulated as the minimization of the following loss function:

$$\mathcal{L}(\mathbf{A}) = \frac{1}{m} \sum_{i=1}^m \|\mathbf{A}\mathbf{e}_{L_i} - \mathbf{e}_{S_i}\|^2, \quad (3)$$

where  $\mathbf{e}_{L_i}$  and  $\mathbf{e}_{S_i}$  denote the embeddings of the  $i$ -th sample from each model, and  $\|\cdot\|$  denotes the Euclidean norm. The optimal alignment matrix  $\mathbf{A}$  can be solved via least squares:

$$\mathbf{A} = \arg \min_{\mathbf{A}} \|\mathbf{A}\mathbf{E}_L - \mathbf{E}_S\|_F^2 \rightarrow \mathbf{A} = (\mathbf{E}_L^T \mathbf{E}_L)^{-1} \mathbf{E}_L^T \mathbf{E}_S, \quad (4)$$

where  $\|\cdot\|_F$  denotes the Frobenius Norm, which is the square root of the sum of squared matrix elements. The embedding matrix  $\mathbf{E}_L$  is thus transformed from  $\mathbb{R}^{m \times p}$  to the target space  $\mathbb{R}^{m \times q}$  via matrix  $\mathbf{A}$ .

#### 3.3.2 Nonlinear Alignment

In some cases, linear transformations struggle to capture the intricate and complex relationships between embedding spaces, especially when such spaces exhibit highly dense structures. To address this, we explore the use of deep neural networks (DNNs) for embedding space alignment. In this work, we employ a simple multilayer perceptron (MLP) to model the mapping from  $\mathbf{E}_L$  to the target embedding space  $\mathbf{E}_S$ :

$$\mathbf{E}'_S = \text{MLP}(\mathbf{E}_L). \quad (5)$$

Our objective is to train the MLP such that the transformed embeddings  $\mathbf{E}'_S$  approximate the target embeddings  $\mathbf{E}_S$ , thereby improving performance in downstream retrieval tasks. The alignment quality is initially measured by the mean squared error (MSE) loss:

$$\mathcal{L}_{\text{MSE}}(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^m \|\text{MLP}(\mathbf{e}_{L_i}) - \mathbf{e}_{S_i}\|^2, \quad (6)$$

where  $\mathbf{W}$  represents all trainable parameters in the MLP, including weights and biases. By minimizing this loss, we optimize the network parameters to reduce the discrepancy between transformed and target embeddings.

We also recognize that excessive alignment may expose structural details of the target embedding space, potentially leading to security vulnerabilities such as inversion attacks. To mitigate this, we incorporate a security-oriented regularization term into the loss function to penalize overly similar embedding pairs:

$$\mathcal{L}(\mathbf{W}) = \mathcal{L}_{\text{MSE}} + \alpha \cdot \mathcal{L}_{\text{cos}} + \beta \cdot \mathcal{L}_{\text{Huber}} + \gamma \cdot \mathcal{R}_{\text{sim}}, \quad (7)$$

where  $\mathcal{L}_{\text{cos}}$  is the mean cosine distance loss, and  $\mathcal{L}_{\text{Huber}}$  is the Smooth  $L_1$  loss that provides robustness to outliers. The similarity penalty term is defined as:

$$\mathcal{R}_{\text{sim}} = \frac{1}{m} \sum_{i=1}^m \max\left(\cos\left(\text{MLP}(\mathbf{e}_{L_i}), \mathbf{e}_{S_i}\right) - \tau, 0\right), \quad (8)$$

where  $\alpha$  and  $\beta$  control the strength of the respective loss terms,  $\gamma$  regulates the penalty intensity, and  $\tau$  is a threshold above which similarity triggers regularization. During training, the model learns to align  $\mathbf{E}_L$  to  $\mathbf{E}_S$  while preserving safety, thus achieving nonlinear embedding space alignment.

The core idea of STEER stems from two components: (1) **Relative positional relationships maintain consistency**. An approximate embedding is obtained by transforming the embedding generated from a local model using an alignment transformation between different embedding spaces. Although it differs from the original embedding directly generated by the server-side embedding model, they are similar in numerical value and semantics they represent. As shown in Figure 3, even when using approximate embedding for retrieval, the top-k nearest items exhibit consistency in their relative positional relationships. (2) **Structured transformations are more effective**. A variation exists between the approximate embedding and the original embedding. This variation arises from structured transformation, and STEER leverages this variation to defend against EIAs. In schemes combining split learning with DP, a noisy query embedding can also be obtained. DP adds semantic-agnostic noise to embeddings, which reshapes their positional relationships and harms retrieval accuracy. In contrast, structured transformations in STEER perturb embeddings in a way that preserves their semantics and relative positions (nearest neighbors), thereby maintaining retrieval performance.

### 3.4 Security Analysis

In STEER, the service provider can't know any information about the user's local embedding model or any sensitive data in the query text. The service provider can only know the common text used during the setup phase, and these common texts can't provide effective information to the service provider. The local embedding model and the alignment transformation  $\varphi$ , which are stored locally by the user, can be regarded as the encryption key. Suppose the set of local models that users can choose is  $M$ , and the set of possible alignment methods is  $N$ . As the service provider without the knowledge of  $\varphi$ , each random attack attempt only has a probability of  $P = \frac{1}{|M| \cdot |N|}$  to obtain the correct transformation  $\varphi$ . Since there are various choices for local models, opportunities to fine-tune them for personalization, and diverse alignment methods available. **Therefore, both  $|M|$  and  $|N|$  can be considered to approach infinity in practice.** We consider this probability negligible, so it is theoretically difficult for the service provider to obtain the real alignment transformation  $\varphi$ . They can only utilize the user's approximate embedding for retrieval to attempt EIAs. As the approximate embedding obtained by the service provider has a deviation from the original embedding generated directly using the server-side model, it leads to the failure of query text reconstruction through EIAs. Our experimental results further prove our point.

## 4 Experiment

### 4.1 Experiment Setup

**Datasets and Models.** We consider datasets including Natural Questions (Kwiatkowski et al., 2019), Quora, ArguAna, SCIDOCs, and SciFact, all of which are available from the BEIR benchmark (Thakur et al., 2021). The sizes of these datasets range from thousands to millions of samples. We use gtr-t5-base (Ni et al., 2021) as the embedding model on the service provider side, and evaluate four embedding models on the client side: all-MiniLM-L6-v2, all-MiniLM-L12-v2, bge-small-en, and e5-small (Reimers & Gurevych, 2019). In addition, we use only 20% of the Natural Questions corpus to train linear and nonlinear mappings between the embedding spaces of the client and the service provider. This mapping is then generalized and applied in a cross-dataset setting. All experimental evaluations were run on an NVIDIA A800-80G GPU.

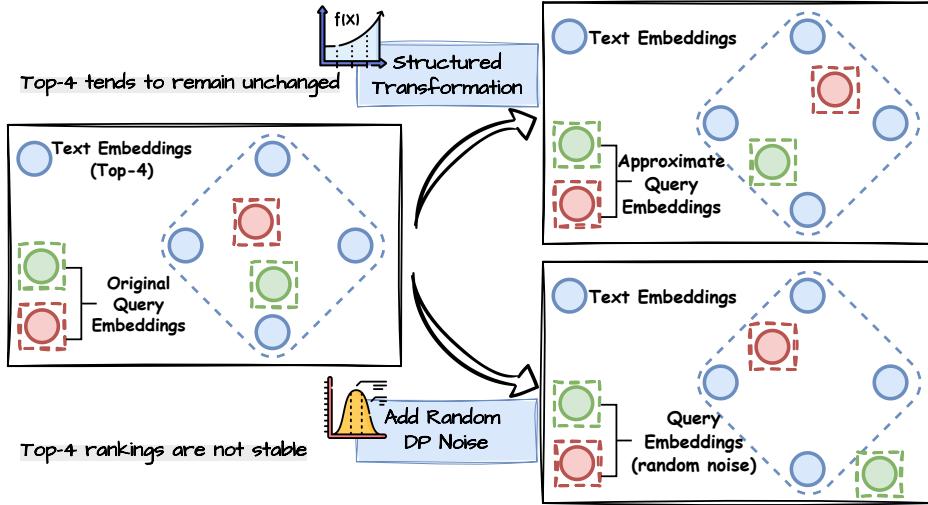


Figure 3: Demonstration in Embedding Space. The structured transformation demonstrates superior performance to unstructured random noise in maintaining retrieval accuracy.

**Baseline.** We adopt the split learning framework SnD (Mai et al., 2024) and anonymization techniques (Yu et al., 2023b; Balepur et al., 2023) as our baselines. These methods do not involve any encryption strategies and have a negligible impact on practical retrieval efficiency.

**Metrics.** We use two categories of metrics to evaluate both retrieval effectiveness and embedding security under EIAs. For retrieval performance, we follow Yu et al. (2023a); Karpukhin et al. (2020); Lewis et al. (2020) and adopt Recall@k as the evaluation metric. For assessing embedding security, we use the following: Rouge-L (Lin, 2004) to measure the accuracy and overlap between original and reconstructed texts; BLEU score (Papineni et al., 2002) to evaluate n-gram similarity between them; and Cosine Similarity (COS) to measure the alignment between the reconstructed and original embedding vectors.

## 4.2 Main Results

We conduct extensive experiments across multiple datasets and models, evaluating retrieval performance and security from both linear and nonlinear alignment perspectives. We design three nonlinear alignment settings (small, medium, and base), each corresponding to an MLP with increasing parameter sizes. More details can be found in the Appendix B. The results further highlight the superiority and advancement of the proposed STEER framework.

### 4.2.1 Retrieval Performance

To ensure a fair comparison under the same security level, we set the noise standard deviation to 25 for the SnD method under EIAs, aligning it with our nonlinear alignment (base) scheme (Figure 6b shows more details). The detailed experimental results are shown in Table 1. Our proposed STEER framework significantly outperforms both SnD and Anonymization across all datasets and settings. Moreover, the nonlinear alignment strategy generally surpasses the linear alignment approach. In terms of Recall@100, our method only suffers a minor drop of about 1%–4% across different datasets. For Recall@300, the performance drop is less than 1% on SciFact, SCIDOCs, ArguAna, and Quora, and only about 4% on NQ. In contrast, the Anonymization method consistently underperforms across all datasets, with Recall@100 typically ranging between 0.3 and 0.6. This indicates that anonymization strategies severely impair semantic representation, thus degrading retrieval quality. Compared to the SnD method at the same privacy level, our framework achieves a retrieval accuracy improvement of approximately 20%–40%. **This demonstrates that the semantically structured transformations in STEER substantially outperform the random perturbations induced by DP-based methods, resulting in smaller accuracy loss and stronger robustness.** Overall, STEER strikes a favorable balance between privacy protection and retrieval performance.

Table 1: The Retrieval Performance of STEER.

Methods→	gtr-t5-base	SnD	anonymize	all-MiniLM-L6-v2		all-MiniLM-L12-v2		bge-small-en		e5-small		
	\	\	\	linear	nonlinear	linear	nonlinear	linear	nonlinear	linear	nonlinear	
SciFact	R@5↑	0.660	0.443	0.163	0.397	0.547	0.413	0.550	0.337	0.593	0.337	0.553
	R@20↑	0.787	0.583	0.223	0.540	0.680	0.583	0.700	0.507	0.727	0.500	0.673
	R@50↑	0.820	0.663	0.297	0.633	0.773	0.667	0.777	0.620	0.807	0.607	0.800
	R@100↑	0.863	0.710	0.357	0.720	0.830	0.737	0.867	0.720	0.860	0.667	0.857
	R@200↑	0.897	0.780	0.417	0.807	0.900	0.810	0.897	0.780	0.887	0.760	0.883
	R@300↑	0.913	0.813	0.457	0.840	0.930	0.857	0.920	0.807	0.910	0.810	0.913
SCIDOCs	R@5↑	0.381	0.189	0.064	0.216	0.320	0.212	0.314	0.162	0.321	0.153	0.311
	R@20↑	0.553	0.320	0.121	0.405	0.532	0.389	0.518	0.330	0.522	0.321	0.482
	R@50↑	0.664	0.440	0.204	0.551	0.651	0.530	0.658	0.480	0.665	0.444	0.614
	R@100↑	0.753	0.553	0.298	0.655	0.742	0.666	0.743	0.605	0.750	0.582	0.709
	R@200↑	0.838	0.662	0.420	0.771	0.816	0.784	0.826	0.725	0.839	0.698	0.811
	R@300↑	0.873	0.734	0.518	0.827	0.867	0.844	0.870	0.808	0.881	0.775	0.861
ArguAna	R@5↑	0.573	0.238	0.204	0.341	0.452	0.356	0.415	0.386	0.495	0.352	0.484
	R@20↑	0.893	0.513	0.383	0.679	0.823	0.691	0.799	0.701	0.838	0.648	0.824
	R@50↑	0.948	0.660	0.508	0.822	0.920	0.824	0.911	0.841	0.929	0.804	0.918
	R@100↑	0.973	0.751	0.598	0.908	0.962	0.910	0.955	0.904	0.967	0.871	0.958
	R@200↑	0.987	0.833	0.676	0.947	0.978	0.952	0.982	0.957	0.982	0.918	0.972
	R@300↑	0.992	0.875	0.731	0.961	0.987	0.967	0.984	0.974	0.985	0.945	0.982
Quora	R@5↑	0.956	0.782	0.342	0.657	0.873	0.685	0.883	0.740	0.894	0.643	0.877
	R@20↑	0.989	0.876	0.445	0.806	0.954	0.835	0.957	0.873	0.967	0.789	0.954
	R@50↑	0.995	0.914	0.511	0.876	0.978	0.904	0.981	0.927	0.985	0.859	0.978
	R@100↑	0.998	0.939	0.565	0.920	0.989	0.939	0.989	0.957	0.992	0.901	0.988
	R@200↑	0.999	0.954	0.616	0.950	0.995	0.963	0.996	0.972	0.996	0.932	0.994
	R@300↑	1.000	0.963	0.647	0.961	0.996	0.973	0.998	0.981	0.997	0.947	0.996
NQ	R@5↑	0.628	0.361	0.159	0.235	0.530	0.249	0.540	0.203	0.512	0.247	0.538
	R@20↑	0.806	0.495	0.225	0.391	0.714	0.406	0.716	0.338	0.695	0.389	0.707
	R@50↑	0.871	0.584	0.276	0.497	0.801	0.513	0.804	0.438	0.778	0.491	0.793
	R@100↑	0.907	0.643	0.316	0.572	0.848	0.581	0.853	0.510	0.833	0.573	0.840
	R@200↑	0.936	0.699	0.360	0.661	0.886	0.665	0.892	0.592	0.874	0.648	0.884
	R@300↑	0.949	0.723	0.383	0.705	0.906	0.708	0.909	0.643	0.895	0.701	0.904

#### 4.2.2 Security

We adopt Vec2Text (Morris et al., 2023) to perform inversion attacks. Table 2 shows the reconstruction quality of embeddings after being processed by our method. The results demonstrate that STEER significantly reduces reconstruction quality, with Rouge-L dropping to about 16%-30% of the original and BLEU scores down to about 0.8%-4%. As model capacity increases, we observe a rise in cosine similarity and a corresponding improvement in reconstruction quality. However, **a stronger fitting capacity does not imply a higher attack risk. Even under the best-fitting scenario, all inversion metrics remain substantially lower than the original, with only about a 1% drop in retrieval accuracy (Recall@300), effectively safeguarding the security of both queries and embeddings.**

Table 2: Inversion Performance in Rouge-L, BLEU and Cosine Similarity.

Methods→	gtr-t5-base	SnD	all-MiniLM-L6-v2			all-MiniLM-L12-v2				
	\	\	linear	small	medium	base	linear	small	medium	base
Rouge-L	77.438	24.104	18.598	19.716	20.127	22.085	17.587	20.426	20.234	21.982
BLEU-1	74.170	25.590	16.750	17.570	17.230	17.700	15.590	16.590	16.170	16.900
BLEU-2	55.700	5.160	2.690	3.300	2.990	3.960	2.360	2.480	2.980	3.890
BLEU	52.490	1.980	1.200	1.600	1.600	2.560	1.070	0.780	1.500	2.210
cos	\	0.526	0.531	0.575	0.617	0.684	0.510	0.575	0.610	0.678
Methods→	gtr-t5-base	SnD	bge-small-en			e5-small				
Metrics	\	\	linear	small	medium	base	linear	small	medium	base
Rouge-L	77.438	24.104	12.757	19.466	20.117	23.694	17.559	20.715	20.597	23.583
BLEU-1	74.170	25.590	15.180	22.390	15.810	18.780	16.940	18.390	17.290	18.150
BLEU-2	55.700	5.160	1.490	3.590	3.180	4.200	2.910	3.590	3.040	4.050
BLEU	52.490	1.980	0.450	1.300	1.570	2.670	1.710	2.160	1.560	2.430
cos	\	0.526	0.396	0.517	0.580	0.680	0.476	0.577	0.616	0.676

#### 4.3 Ablation Study

We conducted three main ablation studies under the STEER framework to further validate its effectiveness, using the low-dimensional embedding model all-MiniLM-L12-v2.

**STEER Using Different MLPs.** Figure 4 shows that as the model’s fitting capacity increases, retrieval performance consistently improves and **remains comparable to that of gtr-t5-base across all datasets**. This indicates that our structured semantic alignment mapping can accurately reconstruct the geometry of the target embedding space, maintaining retrieval accuracy and demonstrating strong generalization. Figure 4a illustrates that even as the fitting ability improves, Rouge-L and BLEU remain significantly lower than directly exposing raw embeddings. This suggests that our embedding perturbation is highly structured and effectively mitigates the risk of EIAs. **These trends are consistent across models and datasets, confirming that STEER enhances security while preserving robustness and generalizability across different deployment settings.**

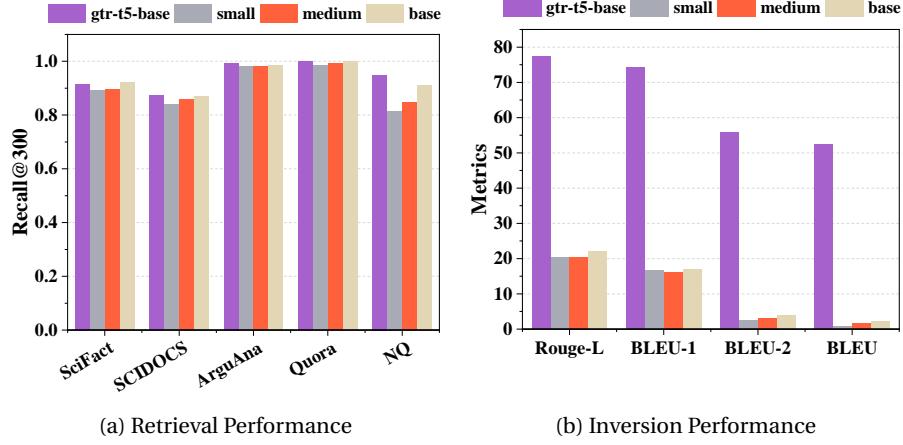


Figure 4: The Performance of Nonlinear Models with Varying Fitting Capacities.

**Comparison between Different Anonymization Levels and STEER.** We leveraged three different anonymization levels (Level 1/2/3), with Level 1 being the strictest; details are provided in Appendix A . Figure 5 shows that anonymization severely degrades retrieval accuracy by removing key query information. For example, on SCIDOCs and NQ, Recall@100 under Level 1 drops below 0.4—far below our method. Even Level 3 fails to restore acceptable performance. **This highlights that while anonymization avoids exposing raw text, it causes irreversible semantic loss. In contrast, STEER preserves semantic content without revealing the original text, maintaining high retrieval performance across multiple datasets.**

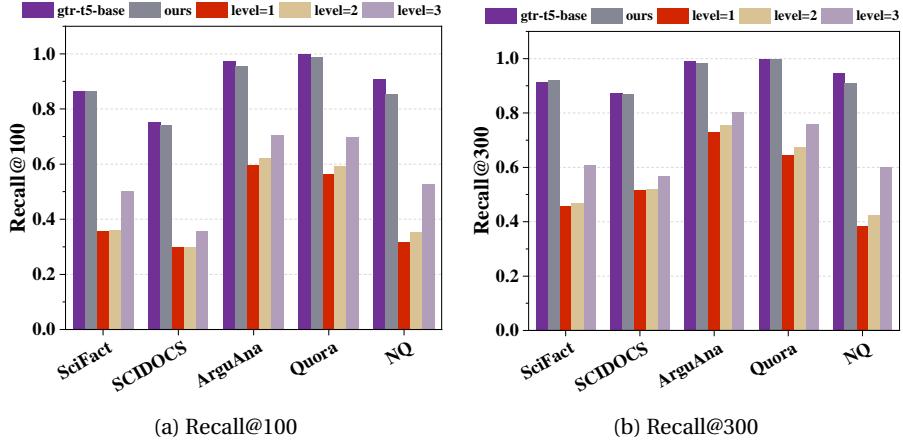


Figure 5: The Retrieval Performance under Varying Levels of Desensitization.

**Comparison between Different SnD Noise Levels and STEER.** In the SnD scheme, it is crucial to select the noise strength to balance privacy and utility. Thus, we adopted different noise levels and measured both retrieval accuracy and inversion risk. As shown in Figure 6, stronger noise leads to lower BLEU and Rouge-L scores, indicating reduced inversion risk, but also results in a sharp decline in Recall@k. In contrast, our STEER (dashed lines) achieves significantly better retrieval performance under comparable inversion risk levels. This suggests that, **compared**

**to the unstructured perturbations used in SnD, STEER provides more stable and effective privacy protection without the need for random noise, while still preserving retrieval accuracy.**

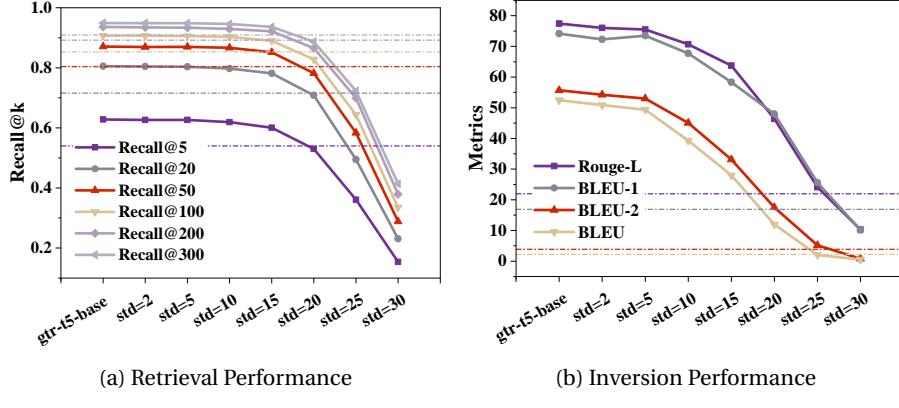


Figure 6: SnD under Varying Noise Levels.

## 5 Discussion

Based on the threat model defined in our work, text embeddings stored in VDB are still generated by the embedding model of the VDB service provider. This embedding model used for retrieval remains under the service provider's control, while users only employ local models to generate approximate embeddings to mitigate EIAs. Naturally, this raises a question: *If users are capable of deploying the embedding model locally to generate both the embeddings of VDB and query embeddings, would security risks persist?* Recent studies proposed by Jha et al. (2025) and Zhang et al. (2025) solve this question. Their embedding inversion methods can still infer and extract text information even if the embedding model is deployed locally. We believe that even in such a new threat model, STEER can still resist EIAs to a certain extent. This is because all embeddings obtained by the attacker are approximate embeddings. The approximate embedding space does not belong to any existing embedding model, making it difficult for attackers to reconstruct text information.

## 6 Conclusion

In this paper, we propose STEER, a novel and privacy-preserving vector retrieval framework designed to protect user query text privacy. During the setup phase, a local embedding model is deployed on the user's device, and a set of non-sensitive, general-purpose texts is used to generate embedding pairs from both the local and server-side models. In the transformation phase, the mapping function between embedding spaces is computed locally using the embedding pairs prepared in the setup phase. When a query is issued, STEER first encodes it using the local model, and then applies the learned transformation to generate an approximate embedding in server's embedding space. The retrieval is finally performed within the original VDB using this approximate embedding. Extensive experiments demonstrate that STEER has no need to transmit query text to the server while resisting EIAs. Crucially, it preserves the semantic integrity of the embeddings to maintain retrieval performance. Our work demonstrates that the deviation introduced by transformation defends against EIAs while significantly improving the retrieval accuracy over current methods.

## References

- Nishant Balepur, Jie Huang, and Kevin Chang. Text fact transfer. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4745–4764, 2023. [7](#)
- Song Bian, Zhou Zhang, Haowen Pan, et al. He3db: An efficient and elastic encrypted database via arithmetic-and-logic fully homomorphic encryption. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2930–2944, 2023. [2](#)
- Tianyu Chen, Hangbo Bao, Shaohan Huang, et al. The-x: Privacy-preserving transformer inference with homomorphic encryption. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL Findings)*, pp. 3510–3520, 2022. [3](#)

- Yiyi Chen, Heather Lent, and Johannes Bjerva. Text embedding inversion security for multilingual language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7808–7827, 2024. 3
- Yiyi Chen, Qiongkai Xu, and Johannes Bjerva. Algen: Few-shot inversion attacks on textual embeddings using alignment and generation. *arXiv:2502.11308*, 2025. 2
- Yihang Cheng, Lan Zhang, Junyang Wang, Mu Yuan, and Yunhao Yao. Remoterag: A privacy-preserving llm cloud rag service. *arXiv:2412.12775*, 2024. 3
- Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, et al. Mmteb: Massive multilingual text embedding benchmark. *arXiv:2502.13595*, 2025. 1
- Wenqi Fan, Yujuan Ding, Liangbo Ning, et al. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 6491–6501, 2024. 3
- Alexandra Henzinger, Emma Dauterman, Henry Corrigan-Gibbs, et al. Private web search with tiptoe. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, pp. 396–416, 2023. 2
- Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X Morris. Harnessing the universal geometry of embeddings. *arXiv:2505.12540*, 2025. 10
- Vladimir Karpukhin, Barlas Onguz, Sewon Min, et al. Dense passage retrieval for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020. 7
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 2019. 6
- Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, et al. Rethinking the role of token retrieval in multi-vector retrieval. *Advances in Neural Information Processing Systems (NIPS)*, 36:15384–15405, 2023. 3
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems (NIPS)*, 33:9459–9474, 2020. 1, 3, 7
- Dacheng Li, Hongyi Wang, Rulin Shao, et al. Mpcformer: Fast, performant and private transformer inference with mpc. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 3
- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, 2004. 7
- Wen-jie Lu, Zhicong Huang, Zhen Gu, et al. Bumblebee: Secure two-party inference framework for large transformers. *Cryptology ePrint Archive*, 2023. 3
- Peihua Mai, Ran Yan, Zhe Huang, et al. Split-and-denoise: Protect large language model inference with local differential privacy. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 34281–34302, 2024. 2, 3, 7
- Kelong Mao, Zheng Liu, Hongjin Qian, et al. Rag-studio: Towards in-domain adaptation of retrieval augmented generation through self-alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP Findings)*, pp. 725–735, 2024. 1
- John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, et al. Text embeddings reveal (almost) as much as text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 12448–12460, 2023. 3, 8
- Jianmo Ni, Chen Qu, Jing Lu, et al. Large dual encoders are generalizable retrievers. *arXiv:2112.07899*, 2021. 6
- Dimitrios P Panagoulias, Maria Virvou, and George A Tsirhrintzis. Augmenting large language models with rules for enhanced domain-specific interactions: The case of medical diagnosis. *Electronics*, 13(2):320, 2024. 3
- Kishore Papineni, Salim Roukos, Todd Ward, et al. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002. 7
- Boci Peng, Yun Zhu, Yongchao Liu, et al. Graph retrieval-augmented generation: A survey. *arXiv:2408.08921*, 2024. 1
- Ori Ram, Yoav Levine, Itay Dalmedigos, et al. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics (TACL)*, 11:1316–1331, 2023. 3
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. 6

- Weijia Shi, Sewon Min, Michihiro Yasunaga, et al. Replug: Retrieval-augmented black-box language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 8364–8377, 2024. [3](#)
- Tobin South, Guy Zuskind, Robert Mahari, and Thomas Hardjono. Secure community transformers: Private pooled data for llms, 2023. [3](#)
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 27, 2014. [2](#)
- Nandan Thakur, Nils Reimers, Andreas Ruckle, et al. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks (NIPS)*, volume 1, 2021. [6](#)
- Peng Xia, Kangyu Zhu, Haoran Li, et al. Rule: Reliable multimodal rag for factuality in medical vision language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1081–1093, 2024. [1](#)
- Wenhai Yu, Dan Iter, Shuohang Wang, et al. Generate rather than retrieve: Large language models are strong context generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023a. [7](#)
- Zhiyuan Yu, Yuhao Wu, Ning Zhang, et al. Codeipprompt: Intellectual property infringement assessment of code language model. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 40373–40389, 2023b. [2](#), [3](#), [7](#)
- Collin Zhang, John X Morris, and Vitaly Shmatikov. Universal zero-shot embedding inversion. *arXiv:2504.00147*, 2025. [10](#)
- Guy Zyskind, Tobin South, and Alex Pentland. Don’t forget private retrieval: Distributed private similarity search for large language models. In *Proceedings of the Workshop on Privacy in Natural Language Processing (PrivateNLP)*, pp. 7–19, 2024. [2](#)

## Appendix

### A Definition of Desensitization Levels

To evaluate the impact of privacy-preserving preprocessing, we adopt a rule-based desensitization framework that categorizes identifiable spans into three semantic sensitivity levels: High, Medium, and Low. The classification is based on named entity types and pattern-based matching. The anonymization level determines the extent of perturbation applied to the input query.

**Level 1 (Strict).** All entities labeled as High, Medium, or Low sensitivity are anonymized. This includes personally identifiable information (PII) such as email addresses, phone numbers, ID numbers, IP addresses, and financial tokens (High), names, organizations, locations, and online handles (Medium), as well as generic numeric or nominal expressions (Low).

**Level 2 (Moderate).** Only Medium and Low sensitivity entities are processed. Highly sensitive entities such as email addresses or identity numbers are retained to preserve more semantic content while still reducing surface identifiability.

**Level 3 (Light).** Only Low sensitivity tokens, such as dates, quantities, and common nouns are anonymized. This minimally alters the semantic structure and serves as a lower bound in the privacy–utility trade-off.

Each sensitivity level is handled using a distinct anonymization operator. High-sensitivity spans are replaced with placeholders. Medium-sensitivity spans are partially masked. Low-sensitivity spans are replaced with generic semantic tokens. The desensitization pipeline integrates SpaCy-based named entity recognition and regular expression matching and is applied line-by-line to all input queries during evaluation.

### B Model Size of Mapping Functions

To study how differences in the fitting capacity of the mapping function affect retrieval performance and security robustness, we design four alignment models with varying capacities. The model sizes are shown in Table 3.

Model		Size
Linear	\	2.3MB
Nonlinear	Small	5.7MB
	Medium	12.33MB
	Base	105.51MB

Table 3: Model Size.

## C Case Study: Examples of EIAs

Examples of EIAs under nonlinear setting (base) with all-MiniLM-L12-v2 are as follows:

**Original:** what happened to regina's mother on switched at birth

**Inversion:** when she drank on her, she woke up to find that she was going to be dead when she was told she was going to switch mom

**Original:** where is fe best absorbed in the body

**Inversion:** body the GF concentration allows the absorption of iron at a favourable level. Citing the literature on the surface of the skin as a

**Original:** is kermit the frog part of sesame street

**Inversion:** Chuck Bruno character Beet Street. Sesame plays a minor part in the a.k.a. 'Kung Fu Panda