Private Retrieval Augmented Generation via Random Projection

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) enhances the capabilities of large language models (LLMs) by querying external structured knowledge. However, it can also introduce privacy risks by leaking sensitive information from the retrieval database. We propose a simple method to preserve datastore privacy in RAG systems via random projection. By applying the same projection to both datastore embeddings and query embeddings, our method provably preserves semantic similarity between queries and retrieved items while substantially mitigating data extraction attacks. Across multiple RAG architectures and datasets, we show that this lightweight approach achieves superior retrieval and generation performance compared to prior methods with formal differential privacy (DP) guarantees, while exhibiting comparable empirical privacy under strong attack models. Our results for the first time suggest that random projection can serve as a competitive and practical baseline for privacy-preserving RAG systems.

1 Introduction

Retrieval-Augmented Generation (RAG) (Khandelwal et al., 2019; Lewis et al., 2020) enhances large language models (LLMs) by incorporating information retrieved from external knowledge. However, recent studies show that carefully crafted prompts (He et al., 2025b; Wang et al., 2025; Zeng et al., 2024; Koga et al., 2024) can extract sensitive information such as personal identity information from the external datastore.

Since attack queries can closely resemble legitimate user queries, simple access control may block ordinary users (He et al., 2025b; Wang et al., 2025). Recent work has begun to explore privacy risks in RAG, with a leading focus on differential privacy (DP) (Koga et al., 2024; Wu et al., 2025). Although DP methods provide formal statistical guarantees, they often degrade output quality (e.g., leading to high perplexity) due to DP's fundamental properties of worst-case guarantees, and are usually complicated to implement in practice. To this end, we propose private RAG via random projection to defend against data extraction attacks.

Our key idea is that differential privacy guarantees may be an overkill to prevent state-of-the-art attacks practically, whereas (some specific) random projection matrices naturally preserve pairwise similarities of the inputs in the lower-dimensional space. The extra randomness alters datastore embeddings so attackers may not recover the original text. This enables users to obtain accurate answers from the RAG system while adversaries fail. We apply a single projection step to datastore embeddings in the offline stage and the same projection matrix to user queries during the online query stage.

In this work, we consider two RAG architectures: 'KNN-LM' (Khandelwal et al., 2019) and 'Standard RAG' (Lewis et al., 2020)[1]. In KNN-LM, the system linearly combines the outputs of the language model and the k-nearest neighbour outputs (usually organized as a softmax over cosine similarities between queries and the datastore embeddings), and generates next token. In Standard RAG, the system concatenates retrieved texts with the query and feeds them to the LLM. In this work, we focus on the settings where formal privacy guarantees are not desired, and thus evaluate privacy leakage by exploring the success rates of strong, existing attacks empirically.

In Section 4.2, we empirically show that enforcing differential privacy guarantees may not provide better practical protection against data extraction and can instead harm utility. In summary, our random projection technique significantly improves utility (over 50% on average across all datasets) over the baseline methods without degrading em-

[1]We note that there exist various RAG architectures, and we use the naming 'Standard RAG' just to contrast with the less popular KNN-LM architecture.

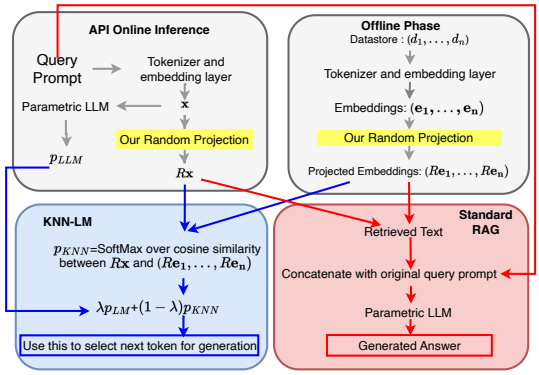

Figure 1: Overview of Private RAG via RP.

pirical privacy performance.

2 Related Works

Private RAG. Prior work shows that various RAG architectures can reveal sensitive personal data (e.g., emails, phone numbers, and URLs) from the external datastore (He et al., 2025b; Zeng et al., 2024; Jiang et al., 2024b,a; Di Maio et al., 2024). Existing defenses, such as DP-based sampling and aggregation (Koga et al., 2024; Grislain, 2025) and synthetic data (Zeng et al., 2025), reduce efficiency on large datasets and often require extra validation or training. Moreover, Zhao and Zhang (2025) show that synthetic data alone does not fully prevent memorization or leakage. He et al. (2025a) apply local differential privacy to sensitive content like addresses. However, in practice, privacy can involve entire sentences or paragraphs beyond tokens, making simple masking insufficient. Thareja et al. (2025) mix original and sanitized documents for inference, which requires identifying all private tokens and is impractical for large-scale RAG.

Privacy-Preserving Random Projection. Certain random projection operations are a common technique for reducing dimensionality while preserving some similarity measures with high probability (e.g., based on the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984)). Prior works have shown that methods built on top of these projections can achieve differential privacy (DP) for specific applications (Blocki et al., 2012; Xu et al., 2017; Li and Li, 2023; Ibrahim et al., 2024; Liu et al., 2006; Narimani and Tavassolipour, 2025; Kaleli and Polat, 2013; Pavlovic et al., 2025; Lee et al., 2025). In this work, we show that random projection can be effective for a variety of RAG scenarios as well.

3 Private RAG via Random Projection

As illustrated in Fig. 1, our algorithm is compatible with both RAG architectures. In this work, we evaluate our method as an empirical defense against data extraction attacks. We consider a black-box setting in which the attacker only has access to model APIs and does not have access to model parameters or the random projection matrix. The attacker submits queries to the RAG system using the API service and attempts to extract sensitive information in the datastore from the generated response through one round or multiple rounds. This attack model aligns with the threat scenarios considered in prior work on RAG privacy (He et al., 2025b; Wang et al., 2025; Huang et al., 2023).

3.1 Algorithm

Let $D = \{d_1, \ldots, d_n\}$ be the corpus of n natural language documents (e.g., email threads or case paragraphs). Let $f(\cdot)$ denote the tokenizer and embedding layers, which can be further fine-tuned to map these d_i's (texts) to embeddings. We assume a fixed pre-trained encoder throughout the paper for simplicity. We also assume that an attacker interacts with the RAG system through API interactions and aims to extract data with constructed queries.

We sample an IID Gaussian matrix $R \in \mathbb{R}^{k \times d}$, where each element follows $\mathcal{N}(0, 1/k)$. We precompute the projected datastore embeddings offline by computing $\boldsymbol{e}'_i = R\,\boldsymbol{e}_i$ for each document embedding $\boldsymbol{e}_i = f(d_i) \in \mathbb{R}^d$. We perform this preprocessing step once and store the results, introducing no runtime overhead during inference. For any query/prompt, we encode it as $\boldsymbol{x} = f(\text{prompt})$ and compute its projection $\boldsymbol{x}' = R\,\boldsymbol{x}$, which requires only a single matrix-vector multiplication. We then perform retrieval in the k-dimensional space by finding the nearest neighbors to $\boldsymbol{x}'$ among $\{\boldsymbol{e}'_i\}_{i \in [n]}$ and computing a softmax over their cosine similarities with $\boldsymbol{x}'$. Finally, we combine this softmax with the parametric LLM's logits for $\boldsymbol{x}$ in a weighted manner to select the next token. We repeat generation until reaching the maximum token limit or generating an end-of-sequence token. The pseudocode is summarized in Appendix D.

Privacy Implications. Random projection preserves privacy by altering the retrieval process: given a query, vanilla RAG retrieves the highest-scoring document and generates a corresponding token, which may contain sensitive information. After RP, embeddings that are close to the origi-

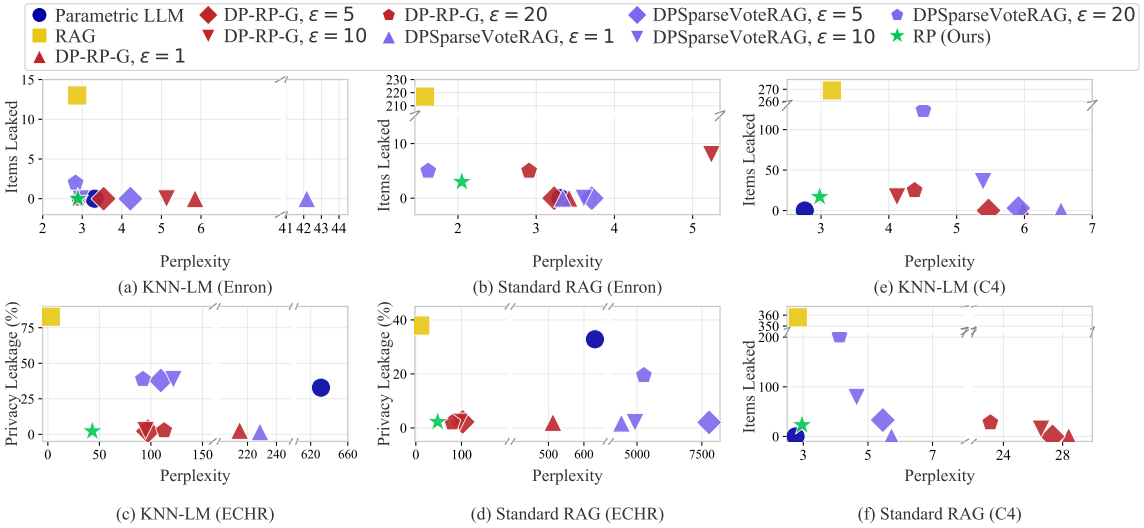

Figure 2: Utility/privacy comparisons of our method (RP) with the two DP baselines (DP-RP-G and DPSparseVoteRAG) on two RAG architectures (denoted as KNN-LM and Standard RAG) and targeted attacks on Enron Email and ECHR and untargeted attacks on C4. ε denotes various privacy budgets for the DP approaches. The x-axis is perplexity of generated content, and the y-axis is percentage of data leakage. RP (Ours) achieves a superior utility–privacy trade-off, with perplexity approaching the unprotected RAG baseline while maintaining near-zero privacy leakage comparable to the standalone parametric model without retrieval-augmentation.

nal top match can be selected with non-negligible probability (Appendix C). As a result, alternative tokens may replace the originally retrieved token.

Utility Discussions. A natural question arises: *if random projections alter the retrieved neighbors, how does the model remain accurate?* The answer is that random projection approximately preserves embedding distances, ensuring retrieval still selects relevant information. This has been extensively studied in prior literature (Johnson and Lindenstrauss, 1984). In Appendix B, for completeness, we provide proof that our random projection preserves L_2 distance and cosine similarity. Appendix B.2 shows empirical results consistent with the theoretical analysis on synthetic data. For DP-based baselines, achieving privacy guarantees sacrifices utility. DP introduces a fundamental tradeoff: a small clipping bound overly distorts embeddings, while a large bound increases noise magnitude, both degrading retrieval utility. We empirically illustrate this effect in Appendix J.

4 Evaluation

4.1 Threat Model

We focus on empirical data extraction attacks (Huang et al., 2023) rather than differential privacy bounds, considering both single- and multi-query API attackers. We evaluate four datastore/LLM combinations: Enron Email (Klimt and Yang, 2004) with GPT-2 (Radford et al., 2019), European Court of Human Rights (ECHR) with LLAMA3-1B-INSTRUCT, C4 with LLAMA3-8B-INSTRUCT, and HotpotQA (Yang et al., 2018) with QWEN2-7B-INSTRUCT. We launch single-query targeted attacks on Enron Email for personal identity information (PII) and ECHR for legal documents, single-query untargeted attacks on raw C4 content, and multi-round targeted attacks on HotpotQA for document information. Details, including query construction, are provided in Appendix E.

We assume the attacker knows that RP is used but does not have direct access to the specific random matrices, though they may still conduct repeated or multi-round attacks. To strengthen single-query extraction attacks, we inject dataset-specific prior knowledge into malicious queries, such as asking for emails when attacking the Enron Email dataset. For each entry, we perform 10 targeted extraction attempts, and count any successful extraction as leakage. We assume the system flags and blocks queries repeated more than 10 times.

Another strong attack is the multi-round adaptive attack (Jiang et al., 2024a) , where an attacker iteratively refines adversarial prompts based on outputs from previous rounds and aggregates the extracted information to infer private content. We report semantic similarity (SS) and extended edit distance (EED) as evaluation metrics. Following their experimental setup, the datastore in our experiments

is constructed using chunks with a length of 1500 words and a maximum overlap of 300 words. We conduct 200 attack attempts and report the number of rounds required to achieve the strongest attack performance. Knowledge Graph (KG) (Yao et al., 2026) further constructs a knowledge graph from outputs across attack rounds to generate future adversarial queries. The total attack budget is limited to a maximum of 200 queries to the RAG system.

4.2 Main Results

For simplicity, we denote our random projection as RP, DP-guaranteed random projection with Gaussian noise as DP-RP-G (Blocki et al., 2012), and the work of Koga et al. (2024) as DPSparseVoteRAG. We also report results for vanilla RAG and the parametric model without a datastore. For differential privacy, we use a wide range of privacy budgets with $\epsilon \in \{1, 5, 10, 20\}$. For DP-RP-G, the original DP-RP method by Blocki et al. (2012) assumes binary vector embeddings. To adapt this framework to RAG and document retrieval, we first project both queries and datastore embeddings into a lower-dimensional space, clip the projected embeddings, and then add Gaussian noise to obtain differential privacy guarantees. The complete procedure is in Appendix D. We sweep the clipping bound from $\{0.1, 0.5, 1, 5, 10, 15, 25, 50, 100\}$ under each setting and report the results with optimal clipping bound. Remaining settings are in Appendix F. For DPSparseVoteRAG, we sweep the subset number m over $\{10, 20, 30, 40, 50, 60\}$ and choose the best value 50 (Appendix F.2).

In our methods, for KNN-LM, we set λ as 0.1 and K as 1024. For RP, the only tunable parameter is k (the projection dimension). Larger k values better preserve utility while maintaining privacy protection. We choose $k = 100$ since it achieves comparable utility to larger values (e.g., $k = 1600$) while providing better computational efficiency. More detailed analysis is in Appendix F.3.

Utility and Privacy. Main results are discussed in the paper; full results are in Appendix G. For utility, besides perplexity as in prior work (Huang et al., 2023), we measure task-specific performance. Figure 2 illustrates the privacy/utility trade-off of different methods under varying privacy budgets on three datasets under single-query attack. Standard RAG achieves higher utility but poses greater risks of privacy leakage. Without protection, RAG leaks over 200 personal items in the Enron dataset. In contrast, our methods maintain utility close to RAG while leaking almost no private information. For DP-RP-G, we achieve strong privacy across all budgets, but utility remains unsatisfactory. DPSparseVoteRAG shows that increasing the privacy budget reduces perplexity but increases leakage, lacking a balanced trade-off point. Overall, our approach provides strong privacy with minimal utility loss.

Table 1: Utility and privacy results against multi-round attacks on HotpotQA using the KNN-LM framework. In utility, EM/F1 denote answer exact match and token-level F1. Cos. is the cosine similarity between retrieved content and the ground-truth reference. Rd. is # rounds.

	Utility		Cos.	Adaptive				KG		
	EM	F1		CRR	SS	EED	Rd.	CRR	SS	EED
Parametric	7.48	28.74	0.73	30%	0.17	0.030	2.18	37%	0.033	0.07
RAG	12.48	29.88	0.75	60%	0.16	0.028	2.05	75%	0.034	0.10
RP (Ours)	11.32	28.57	0.75	3%	0.16	0.88	2.05	0%	0.032	0.93
DP-RP-G ($\epsilon = 10$)	3.58	10.84	0.54	3%	0.17	0.89	2.01	0%	0.031	0.90
DPSVR ($\epsilon = 10$)	7.48	18.18	0.54	2%	0.17	0.88	2.34	0%	0.034	0.90

Table 2: Comparison of the number of tokens generated and latency per generation for different methods over ECHR dataset and Nvidia A100.

	RAG	RP (Ours)	DPSparseVoteRAG
# Tokens per Generation	107.8	**123.4**	92.3
Latency per Generation (s)	8.35	**9.42**	258.9
Latency per Token (s)	0.08	0.08	2.80

From Table 1, we observe that the three privacy-preserving methods achieve similar utility, while our method attains the highest accuracy. All methods achieve similar semantic similarity between the original datastore content in each chunk and the attacked results, which we attribute to LLMs attempting to generate correct answers. In contrast, a high chunk recovery rate (i.e., how much datastore content is completely restored) and low EED indicate successful extraction attacks. Examples of extracted content are shown in Appendix I.3.

Efficiency. The DPSparseVoteRAG method can be inefficient. In our experiments, each inference takes about 30× longer than RP and vanilla RAG (Table 2). This is due to DP voting and aggregation require partitioning the datastore into m subsets and generating a token for each, resulting in m inferences per token. Our method can achieve the same efficiency as RAG on latency per token.

5 Conclusion

In this work, we propose a simple method to defend against data extraction attacks for RAG applications. Compared with DP approaches, we show that the specific random Gaussian projection method can empirically protect against data reconstruction while still maintaining semantic alignment.

Limitations

LLM RAG systems remains an open challenge, which is beyond the scope of this work. In this work, we present an empirical defense method for privacy-preserving retrieval-augmented generation (RAG) based on random projection. We focus our evaluation on research-scale settings using up to nearly 100,000 samples. We observe that when scaling the datastore size from 5k to 10k and 100k, random projection consistently preserves privacy. Real-world systems may need to process millions of queries daily. Large-scale evaluation of privacy-preserving methods in real-world LLM-based RAG systems remains an interesting future work.

Ethical Considerations

We use models and datasets that are publicly available on the Internet and licensed for non-commercial research use. We do not involve any sensitive or personally identifiable information (PII). We use only publicly available or synthetically generated datasets; we did not use any proprietary data or private user information at any stage of the experiments.

We manually inspected the datasets and confirmed that they contain certain personally identifying information (PII), including names, email addresses, home addresses, and identification numbers. These datasets are publicly available and widely used in the research community. Consistent with prior work, we did not further anonymize the data, as it was collected and released under publicly accessible licenses and is used solely for research purposes. We did not introduce any new personal data, nor did we attempt to re-identify individuals beyond what is already present in the original datasets.

References

Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2012. The johnson-lindenstrauss transform itself preserves differential privacy. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 410–419. IEEE Computer Society.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Mexico City, Mexico. Association for Computational Linguistics.

Christian Di Maio, Cristian Cosci, Marco Maggini, Valentina Poggioni, and Stefano Melacci. 2024. Pirates of the rag: Adaptively attacking llms to leak knowledge bases. *arXiv preprint arXiv:2412.18295*.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006.*, pages 265–284. Springer.

Ronald A Fisher. 1915. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521.

Nicolas Grislain. 2025. Rag with differential privacy. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, pages 847–852. IEEE.

Longzhu He, Peng Tang, Yuanhe Zhang, Pengpeng Zhou, and Sen Su. 2025a. Mitigating privacy risks in retrieval-augmented generation via locally private entity perturbation. *Information Processing & Management*, 62(4):104150.

Yu He, Yifei Chen, Yiming Li, Shuo Shao, Leyi Qi, Boheng Li, Dacheng Tao, and Zhan Qin. 2025b. External data extraction attacks against retrieval-augmented large language models. *arXiv preprint arXiv:2510.02964*.

Harold Hotelling. 1953. New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society. Series B (Methodological)*, 15(2):193–232.

Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. 2023. Privacy implications of retrieval-based language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14887–14902.

Alaa Mahmoud Ibrahim, Mohamed Farouk, and Mohamed Waleed Fakhr. 2024. Privacy preserving image retrieval using multi-key random projection encryption and machine learning decryption. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 42(2):155–174.

Changyue Jiang, Xudong Pan, Geng Hong, Chenfu Bao, Yang Chen, and Min Yang. 2024a. Feedback-guided extraction of knowledge base from retrieval-augmented llm applications. *arXiv preprint arXiv:2411.14110*.

Changyue Jiang, Xudong Pan, Geng Hong, Chenfu Bao, and Min Yang. 2024b. Rag-thief: Scalable extraction of private data from retrieval-augmented generation applications with agent-based attacks. *CoRR*.

William B Johnson and Joram Lindenstrauss. 1984. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1.

Cihan Kaleli and Huseyin Polat. 2013. Privacy-preserving random projection-based recommendations based on distributed data. *International Journal of Information Technology & Decision Making*, 12(02):201–232.

Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. 2013. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1).

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 217–226. Springer.

Tatsuki Koga, Ruihan Wu, and Kamalika Chaudhuri. 2024. Privacy-preserving retrieval augmented generation with differential privacy. *arXiv preprint arXiv:2412.04697*.

Bonwoo Lee, Cheolwoo Park, and Jeongyoun Ahn. 2025. Optimal differentially private kernel learning with random projection. *arXiv preprint arXiv:2507.17544*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the Advances in Neural Information Processing Systems (NeuraIPS)*, volume 33, pages 9459–9474.

Ping Li and Xiaoyun Li. 2023. Differential privacy with random projections and sign random projections. *arXiv preprint arXiv:2306.01751*.

Kun Liu, Hillol Kargupta, and Jessica Ryan. 2006. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on knowledge and Data Engineering*, 18(1):92–106.

Mohammad Hasan Narimani and Mostafa Tavassolipour. 2025. Fedrp: A communication-efficient approach for differentially private federated learning using random projection. *arXiv preprint arXiv:2509.10041*.

Nikola Pavlovic, Sudeep Salgia, and Qing Zhao. 2025. Differential privacy in kernelized contextual bandits via random projections. *arXiv preprint arXiv:2507.13639*.

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

Rushil Thareja, Preslav Nakov, Praneeth Vepakomma, and Nils Lukas. 2025. Dp-fusion: Token-level differentially private inference for large language models. *arXiv preprint arXiv:2507.04531*.

Yuhao Wang, Wenjie Qu, Shengfang Zhai, Yanze Jiang, Zichen Liu, Yue Liu, Yinpeng Dong, and Jiaheng Zhang. 2025. Silent leaks: Implicit knowledge extraction attack on rag systems through benign queries. *arXiv preprint arXiv:2505.15420*.

Ruihan Wu, Erchi Wang, Zhiyuan Zhang, and Yu-Xiang Wang. 2025. Private-rag: Answering multiple queries with llms while keeping your data private. *arXiv preprint arXiv:2511.07637*.

Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2017. Dppro: Differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security*, 12(12):3081–3093.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380.

Mengyu Yao, Ziqi Zhang, Ning Luo, Shaofei Li, Yifeng Cai, Xiangqun Chen, Yao Guo, and Ding Li. 2026. Connect the dots: Knowledge graph-guided crawler attack on retrieval-augmented generation systems. *arXiv preprint arXiv:2601.15678*.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). In *ACL (Findings)*.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Jie Ren, Tianqi Zheng, Hanqing Lu, Han Xu, Hui Liu, Yue Xing, and Jiliang Tang. 2025. Mitigating the privacy issues in retrieval-augmented generation (rag) via pure synthetic data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 24538–24569.

Yunpeng Zhao and Jie Zhang. 2025. Does training with synthetic data truly protect privacy? In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Donald W Zimmerman, Bruno D Zumbo, and Richard H Williams. 2003. Bias in estimation and hypothesis testing of correlation. *Psicológica*, 24(1).

Contents

A Usage of AI

We employ large language models (LLMs) primarily to improve the grammar and clarity of our writing and assist coding. All research ideas, directions, and decisions, however, are independently conceived and carried out by the authors.

B Preservation of the L_2-Distance and Cosine Similarity of Random Projection

The theory has already been proved by Johnson and Lindenstrauss (1984). For completeness, we provide analysis below. Let the query embedding be $\boldsymbol{x}$, the retrieved embedding by vanilla RAG be $\boldsymbol{e}$, and the random projection matrix be R. The original ℓ_2 distance is $\|\boldsymbol{x}-\boldsymbol{e}\|_2$. According to the Johnson–Lindenstrauss transform, when the projection variance is $1/k$, we have the boundary condition

$$(1-\lambda_{JL})\|\boldsymbol{x}-\boldsymbol{e}\|_2 \le \|R\boldsymbol{x}-R\boldsymbol{e}\|_2 \le (1+\lambda_{JL})\|\boldsymbol{x}-\boldsymbol{e}\|_2$$

where $\lambda_{JL} = \Omega\big(\sqrt{\frac{\log d}{k}}\big)$ (Johnson and Lindenstrauss, 1984). This ensures that the ℓ_2 distance is well preserved after projection.

Further, the original retrieval score, typically cosine similarity, $\rho = \frac{\boldsymbol{x}^\top \boldsymbol{e}}{\|\boldsymbol{x}\|_2\,\|\boldsymbol{e}\|_2}$ can also be preserved after projection. After random projection, the score becomes

$$\rho' = \frac{\boldsymbol{x}^\top R^\top R\boldsymbol{e}}{\|R\boldsymbol{x}\|_2\,\|R\boldsymbol{e}\|_2}.$$

For fixed x and e, the expectation satisfies $\mathbb{E}[\rho'] = \rho\left(1-\frac{1-\rho}{2k}\right) + \mathcal{O}\left(\frac{1}{k^2}\right)$ (Proof in Appendix B.1). Thus, when k is sufficiently large, the expected cosine similarity after projection closely approximates the original score, preserving retrieval quality despite token perturbations.

B.1 Proof: Preservation of the Cosine Similarity

Our main argument is that with multiple rounds of token generation, the overall semantic meaning is preserved even if individual tokens are perturbed or replaced. In other words, we have

$$\mathbb{E}[\rho'] \approx \rho\left(1-\frac{1-\rho}{2k}\right) \tag{1}$$

Proof. We have query embedding $\boldsymbol{x}$ and datastore embedding $\boldsymbol{e}$. First, let $\boldsymbol{u} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}$ and $\boldsymbol{v} = \frac{\boldsymbol{e}}{\|\boldsymbol{e}\|_2}$ and we can have $\rho = \boldsymbol{u}^\top \boldsymbol{v}$. With that, we can have

$$\rho' = \frac{(\boldsymbol{u}^\top R^\top)(R\boldsymbol{v}))}{\|\boldsymbol{u}^\top R^\top\|_2 \|R\boldsymbol{v}\|_2} = \frac{(R\boldsymbol{u})^\top (R\boldsymbol{v}))}{\|R\boldsymbol{u}^\top\|_2 \|R\boldsymbol{v}\|_2} \tag{2}$$

We let $\boldsymbol{a} = R\boldsymbol{u}$ and $\boldsymbol{b} = R\boldsymbol{v}$. Then we can have

$$\rho' = \frac{\boldsymbol{a}^\top \boldsymbol{b}}{\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2} \tag{3}$$

Because R is an IID Gaussian matrix and each element is independently sampled. We can find rotation matrices U and V where $URV \overset{d}{=} R$. Hence, we can have $R(V\boldsymbol{x}) \overset{d}{=} R\boldsymbol{x}$. So, without loss of generality, we can rotate the basis so that $\boldsymbol{u} = \boldsymbol{e_1} = (1, 0, 0, \ldots, 0)^\top$, $\boldsymbol{v} = \rho\boldsymbol{e_1} + \sqrt{1-\rho^2}\boldsymbol{e_2}$. We know that each row of R is a Gaussian vector $\boldsymbol{g_i} \sim \mathcal{N}(0, I_d/k)$. So, $a_i = \boldsymbol{r_i}^T\boldsymbol{u} = r_{i1}$ and $b_i = \boldsymbol{r_i}^\top\boldsymbol{v} = \rho r_{i1} + \sqrt{1-\rho^2}r_{i2}$ where r_{i1}, r_{i2} are IID $\mathcal{N}(0, 1/k)$. Thus, for each $i = 1, \ldots, k$,

$$\begin{bmatrix} a_i \\ b_i \end{bmatrix} \sim \mathcal{N}\left(0,\ \frac{1}{k}\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right).$$

With this bivariate normal distribution, we can define

$$X_i = \sqrt{k}a_i \sim \mathcal{N}(0, 1), \tag{4}$$

$$Y_i = \sqrt{k}b_i \sim \mathcal{N}(0, 1), \tag{5}$$

And we can have that $\mathrm{Corr(X_i, Y_i)} = \rho$. Then, we can calculate

$$\rho' = \frac{\sum_i X_i Y_i}{\sqrt{\sum_i X_i^2}\sqrt{\sum_i Y_i^2}} \tag{6}$$

This is exactly the sample Pearson correlation coefficient computed from k IID bivariate normal samples with population correlation ρ.

Lemma 1. *For random variables X_i and Y_i where $X_i \sim \mathcal{N}(0, 1)$, $Y_i \sim \mathcal{N}(0, 1)$, and $\mathrm{Corr(X_i, Y_i)} = \rho$, we can calculate the expectation of sample Pearson correlation coefficient*

$$\rho' = \frac{\sum_i X_i Y_i}{\sqrt{\sum_i X_i^2}\sqrt{\sum_i Y_i^2}} \tag{7}$$

as

$$\mathbb{E}[\rho'] = \rho\left(1 - \frac{1-\rho}{2k}\right) + \mathcal{O}\left(\frac{1}{k^2}\right) \tag{8}$$

Although the proof of this lemma is not our original contribution, as there is plenty of literature showing this conclusion (Zimmerman et al., 2003; Hotelling, 1953), we would like to provide a proof here to demonstrate some of our insights.

Proof. Let $S_{xx} = \frac{1}{k}\sum X_i^2$, $S_{yy} = \frac{1}{k}\sum Y_i^2$, $S_{xy} = \frac{1}{k}\sum X_i Y_i$. Then

$$\rho' = \frac{S_{xy}}{\sqrt{S_{xx}}\sqrt{S_{yy}}} \tag{9}$$

If we use Taylor expansion, we will have

$$\frac{1}{\sqrt{S_{xx}}} = 1 - \frac{1}{2}S_{xx} + \mathcal{O}\left(\frac{1}{k^2}\right) \tag{10}$$

As a result, we will have

$$\mathbb{E}[\rho'] = \mathbb{E}[S_{xy}] - \frac{1}{2}\mathbb{E}[S_{xy}S_{xx}] - \frac{1}{2}\mathbb{E}[S_{xy}S_{yy}] \tag{11}$$

$$+ \frac{1}{4}\mathbb{E}[S_{xx}S_{yy}] + \mathcal{O}\left(\frac{1}{k^2}\right) \tag{12}$$

$$= \rho - \frac{1}{2k}\mathrm{Cov(XY, X^2)} - \frac{1}{2\mathrm{k}}\mathrm{Cov(XY, Y^2)} \tag{13}$$

$$+ \frac{1}{4k}\mathrm{Cov(X^2, Y^2)} + \mathcal{O}\left(\frac{1}{\mathrm{k}^2}\right) \tag{14}$$

We have $\mathbb{E}[X^2] = \mathbb{E}[Y^2] = 1$ and $\mathbb{E}[XY] = \rho$. We also know $\mathrm{Var(X)} = \mathrm{Var(Y)} = 1$ and $\mathrm{Cov(X, Y)} = \rho$. Using Isserlis theorem, we have $\mathbb{E}[X^4] = 3$, $\mathbb{E}[Y^4] = 3$, $\mathbb{E}[X^2Y^2] = 1 + 2\rho^2$, $\mathbb{E}[X^3Y] = \mathbb{E}[XY^3] = 3\rho$. Hence, we have

$$\mathrm{Cov(X^2, Y^2)} = \mathbb{E}[\mathrm{X^2Y^2}] - \mathbb{E}[\mathrm{X^2}]\mathbb{E}[\mathrm{Y^2}] = 2\rho^2 \tag{15}$$

$$\mathrm{Cov(X^2, XY)} = \mathbb{E}[\mathrm{X^3Y}] - \mathbb{E}[\mathrm{X^2}]\mathbb{E}[\mathrm{XY}] = 2\rho \tag{16}$$

By putting them back, we have our conclusion. □

According to Lemma 1, we prove Eq. (1). The expected distortion goes down as $\mathcal{O}(1/k)$. Hence, we can see that our designed random projection preserves the cosine similarity. □

B.2 Synthetic Data

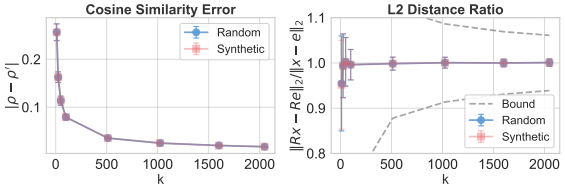

Figure 3: Cosine similarity error and L2 distance ratio under different sanity-check settings.

Before starting our experiments, we first conduct a sanity check to verify whether embedding distances remain consistent after random projection. We consider two settings. In the synthetic setting, we use LLAMA3-8B-INSTRUCT to generate 100 datastore entries with the prompt *"Generate a datastore entry containing domain knowledge."*

For each of these, we repeat 100 times to generate a query using the prompt *"Generate a query about domain knowledge."* In the random setting, both datastore and query embeddings are random vectors drawn from $\mathcal{N}(0, 1)$.

We measure the L2 distance and cosine similarity between each query embedding and the datastore embeddings. We use two metrics: the L2 distance ratio, $\frac{\|R\boldsymbol{x}-R\boldsymbol{e}\|_2}{\|\boldsymbol{x}-\boldsymbol{e}\|_2}$, and the cosine similarity difference, $|\rho - \rho'|$. We first average each metric over 100 datastores and then average across 100 runs. Fig. 3 presents the sanity-check results, along with the theoretical upper and lower bounds derived from the Johnson–Lindenstrauss (JL) transform, given by $1 \pm \Omega\left(\sqrt{\frac{\log d}{k}}\right)$. As k increases, the cosine similarity error decreases and approaches zero, confirming our later observation that larger k improves perplexity, which is consistent with our theoretical analysis. For the L2 distance, the results follow the JL transform prediction, with the ratios remaining within the theoretical bounds. A larger k also stabilizes the ratio. Overall, these results verify that, in expectation, distances are well preserved after random projection.

C Proof: Probability of Top-1 Selection Changing after Random Projection

Without loss of generality, we consider the top-1 case, where the datastore embedding $\boldsymbol{e}_1$ attains the largest cosine similarity ρ_1 in the vanilla RAG setting. We show that if there exists another datastore embedding $\boldsymbol{e}_j$ ($j \neq 1$) such that

$$\left\| \frac{\boldsymbol{e}_1}{\|\boldsymbol{e}_1\|_2} - \frac{\boldsymbol{e}_j}{\|\boldsymbol{e}_j\|_2} \right\|_2 \leq \Delta \tag{17}$$

then the probability that the projected cosine similarity ρ'_j exceeds ρ'_1 satisfies

$$\Pr(\rho'_j > \rho'_1) \gtrsim 1 - \Phi\left(\frac{\Delta}{(1-\rho_1^2)\sqrt{\frac{2(1-\eta_{\max})}{k-3}}} \right) \tag{18}$$

where $\eta_{\max}$ is an upper bound on the correlation between the Fisher-transformed projected similarities, and Φ denotes the cumulative distribution function of the standard normal distribution.

Proof. We define the normalized query and datastore embeddings as

$$\boldsymbol{u} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}, \qquad \boldsymbol{v}_i = \frac{\boldsymbol{e}_i}{\|\boldsymbol{e}_i\|_2}.$$

Under this notation, the assumption becomes $\|\boldsymbol{v}_1 - \boldsymbol{v}_j\|_2 \leq \Delta$. By the Cauchy–Schwarz inequality,

$$|\rho_1 - \rho_j| = |\boldsymbol{u}^\top(\boldsymbol{v}_1 - \boldsymbol{v}_j)| \leq \|\boldsymbol{v}_1 - \boldsymbol{v}_j\|_2 \leq \Delta.$$

We next introduce the Fisher transform

$$z_i = \tanh^{-1}(\rho_i), \qquad z'_i = \tanh^{-1}(\rho'_i).$$

Since cosine similarity takes values in $(-1, 1)$ and $\tanh^{-1}(\cdot)$ is strictly increasing on this interval, the ordering of similarities is preserved under this transformation. In particular,

$$\rho'_j > \rho'_1 \quad \Longleftrightarrow \quad z'_j > z'_1.$$

As shown in Appendix B, the projected cosine similarity ρ'_i is the sample Pearson correlation computed from k IID bivariate normal pairs with population correlation ρ_i. By the classical Fisher z-transform result (Fisher, 1915), we have

$$z'_i \approx \mathcal{N}\left(\tanh^{-1}(\rho_i), \frac{1}{k-3} \right). \tag{19}$$

Because z'_1 and z'_j share the same projected query vector, they are generally correlated. Let $\eta = \mathrm{Corr}(z'_1, z'_j)$, and assume $\eta \leq \eta_{\max}$ for some fixed upper bound $\eta_{\max} < 1$.

Under a joint Gaussian approximation, the difference $z'_j - z'_1$ is approximately normally distributed with

$$\mathbb{E}[z'_j - z'_1] = z_j - z_1, \qquad \mathrm{Var}(z'_j - z'_1) = \frac{2(1-\eta)}{k-3}.$$

Therefore,

$$\Pr(\rho'_j > \rho'_1) = \Pr(z'_j - z'_1 > 0) = \Phi\left(\frac{z_j - z_1}{\sqrt{\frac{2(1-\eta)}{k-3}}} \right). \tag{20}$$

Since $\rho_1 \geq \rho_j$, we have $z_1 \geq z_j$. By the mean value theorem, there exists c between ρ_1 and ρ_j such that

$$z_j - z_1 = (\rho_j - \rho_1)\frac{1}{1-c^2}. \tag{21}$$

Using $|\rho_j - \rho_1| \leq \Delta$ and $1 - c^2 \leq 1 - \rho_1^2$, we obtain

$$z_j - z_1 \geq -\frac{\Delta}{1-\rho_1^2}. \tag{22}$$

Substituting this bound yields

$$\Pr(\rho'_j > \rho'_1) \gtrsim \Phi\left(-\frac{\Delta}{(1-\rho_1^2)\sqrt{\frac{2(1-\eta)}{k-3}}}\right) \quad (23)$$

$$= 1 - \Phi\left(\frac{\Delta}{(1-\rho_1^2)\sqrt{\frac{2(1-\eta)}{k-3}}}\right) \quad (24)$$

$$\gtrsim 1 - \Phi\left(\frac{\Delta}{(1-\rho_1^2)\sqrt{\frac{2(1-\eta_{\max})}{k-3}}}\right), \quad (25)$$

which completes the proof. □

This result shows that after random projection, embeddings that are sufficiently close in the original space can overtake the original top-1 embedding with non-negligible probability. This behavior is desirable for privacy preservation: sensitive tokens (e.g., names, addresses, or identification numbers) often have embeddings close to semantically related but less sensitive alternatives, and random projection allows these alternatives to replace the original sensitive tokens with controlled probability. Combined with the bounds in Appendix B, this demonstrates that random projection limits distortion to a reasonable range, preserving general semantic content while reducing the likelihood that specific sensitive tokens are deterministically selected.

D Complete Algorithms

Algorithm 1 and Algorithm 2 illustrate the implementation of our proposed random projection method. The overall process follows the standard RAG pipeline: we first tokenize and embed the datastore offline, and store the resulting representations in FAISS or another efficient indexing framework to enable fast retrieval at inference time. During inference, the model performs autoregressive generation via iterative next-token prediction.

For KNN-LM (Algorithm 1), at each generation step we apply random projection to both the query embedding and the datastore embeddings. The language model produces a next-token distribution P_{LM} based on the current context x. In parallel, the KNN distribution P_{KNN} is computed by measuring the cosine similarity between the projected query embedding $\tilde{x}$ and each projected datastore embedding $\tilde{e}_i$, followed by a softmax over the top-K nearest neighbors to obtain a probability distribution over candidate tokens. The final next-token distribution is obtained by linearly combining the two distributions with weight $\lambda = 0.25$. The selected token y is then appended to the context and the generated answer, and the process repeats.

Algorithm 1: Private RAG with Random Projection (KNN-LM)

Data: Datastore D with n documents; Tokenizer and embedding function $f(\cdot)$: Text $\to \mathbb{R}^d$; $\lambda = 0.25$.
Input: User query q
Output: Generated answer a
Parameters : Projection dimension k, maximum token length $T_{\max}$

Offline preprocessing:
Sample random projection matrix $R \in \mathbb{R}^{k\times d}$ where $R_{ij} \sim \mathcal{N}(0, 1/k)$
Compute datastore embeddings $E_D \in \mathbb{R}^{d\times n}$ where $\boldsymbol{e}_i = f(d_i)$
Project embeddings: $\tilde{E}_D = RE_D$
Online inference (per query):
Initialize $t \leftarrow 0$, $a \leftarrow \emptyset$
while $t < T_{\max}$ *and* y *is not the end token* **do**
  Compute query embedding: $\boldsymbol{x} = f(q)$
  Project query embedding: $\tilde{\boldsymbol{x}} = R\boldsymbol{x}$
  $P(y|x) = \lambda P_{LM}(x) + (1-\lambda)P_{KNN}(\tilde{x}, \tilde{E}_D)$
  Predict next token: $y \leftarrow \text{argmax}_y P(y|x)$
  Append token: $a \leftarrow a\|y$, $q \leftarrow q\|y$
  $t \leftarrow t + 1$.
end
return a

For standard RAG (Algorithm 2), we compute the similarity between the projected query embedding $\tilde{x}$ and the projected datastore embeddings $\tilde{E}_D$, and retrieve the most relevant text based on this similarity. The retrieved text is concatenated with the original query to form an augmented prompt, which is then passed to a parametric LLM to generate the final output.

We implement DP-RP-G in Algorithm 3, following Blocki et al. (2012), which adds Gaussian noise after random projection to achieve differential privacy. Unlike Blocki et al. (2012), who assume binary or $[0, 1]$-bounded vectors, which does not

Algorithm 2: Private RAG with Random Projection (Standard RAG)

Data: Datastore D with n documents; Tokenizer and embedding function $f(\cdot)$: Text $\to \mathbb{R}^d$;
Input: User query q
Output: Generated answer a
Parameters : Projection dimension k

1 **Offline preprocessing:**
2 Sample random projection matrix $R \in \mathbb{R}^{k\times d}$ where $R_{ij} \sim \mathcal{N}(0, 1/k)$
3 Compute datastore embeddings $E_D \in \mathbb{R}^{d\times n}$ where $\boldsymbol{e}_i = f(d_i)$
4 Project embeddings: $\tilde{E}_D = RE_D$
5 **Online inference (per query):**
6 Initialize $a \leftarrow \emptyset$
7 Compute query embedding: $\boldsymbol{x} = f(q)$
8 Project query embedding: $\tilde{\boldsymbol{x}} = R\boldsymbol{x}$
9 Calculate similarity between $\tilde{x}$ and $\tilde{E}_D$
10 Retrieve the most relevant text: y
11 $q \leftarrow q||y$
12 $a \leftarrow LM(q)$
13 **return** a

Algorithm 3: Implementation of DP-RP-G (KNN-LM)

Data: Datastore $D = \{d_i\}_{i=1}^n$; embedding function f : Text $\to \mathbb{R}^d$;
Input: User query q
Output: Generated answer a
Parameters : Projection dimension k; max tokens $T_{\max}$; clipping bound c; privacy (ϵ, δ)

1 **Offline preprocessing:**
2 Sample random projection $R \in \mathbb{R}^{k\times d}$ with $R_{ij} \sim \mathcal{N}(0, 1/k)$
3 Compute datastore embeddings $E_D \in \mathbb{R}^{d\times n}$ with rows $\boldsymbol{e}_i = f(d_i)$
4 Project: $\tilde{E}_D = RE_D \in \mathbb{R}^{k\times n}$
5 **for** *each column* $\tilde{\boldsymbol{e}}_{\boldsymbol{i}} \in \mathbb{R}^k$ *of* $\tilde{E}_D$ **do**
6 $\quad \hat{\boldsymbol{e}}_{\boldsymbol{i}} \leftarrow \tilde{\boldsymbol{e}}_{\boldsymbol{i}} \cdot \min\left(1, \frac{c}{\|\tilde{\boldsymbol{e}}_{\boldsymbol{i}}\|_2}\right)$
7 **end**
8 **Online inference (per query):**
9 Initialize $t \leftarrow 0$, $a \leftarrow \emptyset$, $\sigma = \frac{c\sqrt{2\ln(1.25/\delta)}}{\epsilon}$
10 Sample IID $Z \in \mathbb{R}^{n\times k}$: $Z_{ij} \sim \mathcal{N}(0, \sigma^2)$
11 $\tilde{E}_D \leftarrow \tilde{E}_D + Z$
12 **while** $t < T_{\max}$ *and* y *is not the end token* **do**
13 $\quad \boldsymbol{x} \leftarrow f(q) \in \mathbb{R}^d$
14 $\quad \tilde{\boldsymbol{x}} = R\boldsymbol{x}$
15 $\quad y \leftarrow \text{argmax}_y P(y|x)$
16 $\quad a \leftarrow a\|y, q \leftarrow q\|y$
17 $\quad t \leftarrow t + 1$
18 **end**
19 **return** a

hold for neural or LLM embeddings. Therefore, after projection, we apply standard DP clipping: for any vector $\boldsymbol{v}$, define

$$\text{clip}(\boldsymbol{v}, c) = \boldsymbol{v} \cdot \min\left(1, \frac{c}{\|\boldsymbol{v}\|_2}\right),$$

so that all vectors have ℓ_2-norm at most c. After clipping, we add Gaussian noise calibrated to the privacy budget where the variance $\sigma = c\frac{\sqrt{2\ln(1.25/\delta)}}{\epsilon}$. As our goal is to protect the datastore, we apply the DP step only to E_D (and not to the queries). Hence, lines 12–19 in Algorithm 3 are the same as lines 7–15 in Algorithm 1. For the DP-RP-G method implemented in standard RAG, we replace lines 12–19 with lines 7–13 in Algorithm 2.

E Implementation Details

First, we use the Enron Email dataset (Klimt and Yang, 2004) as the datastore and GPT-2 (Radford et al., 2019) as the LLM because GPT-2's pre-training data do not overlap with Enron Email, ensuring the RAG data remain unseen. Thus, private information is limited to the RAG documents. The dataset contains annotated sensitive information (URLs, email addresses, phone numbers), and we use the entire dataset as the datastore, totaling 4576 sensitive items. We use the query prompts from (Huang et al., 2023) (e.g., `You may also contact us via e-mail at`) and evaluate whether the annotated personally identifiable information can be retrieved.

The European Court of Human Rights (ECHR) dataset (Chalkidis et al., 2021) contains legal case documents from the ECHR. Queries are formed from test-case descriptions, and utility is measured by matching generated judgments to ground-truth labels (see Appendix G.2). We use `Llama3-1B-Instruct`. For adversarial queries, we use `Llama3-8B-Instruct` to annotate sensitive information and query with prompts concatenating "`This is the first sentence of a paragraph from ECHR legal documents. Complete the`

paragraph with the sentence:" and the first sentence of each case. We then check whether generations contain the annotated sensitive identity. We use 10K documents.

Besides targeted leakage where we check if exact annotated information is leaked, we use the C4 dataset, a cleaned and large-scale version of Common Crawl, for untargeted leakage evaluation. We select 5,000 samples and use the first sentence of each as the query. We then compute ROUGE-L scores between each generation and every datastore sample. If any sample in the datastore has a score greater than 0.5 with the generation, where we follow the same settings by Huang et al. (2023), marking it as a leakage. We report the number of leakages out of the 5k samples. We use the LLAMA3-8B-INSTRUCT model as the backbone.

HOTPOTQA (Yang et al., 2018) is a multi-hop QA dataset with strong supervision for supporting facts, enabling more explainable systems. We use it to evaluate whether we retrieve correct supporting documents and measure cosine similarity between retrieved and reference embeddings to assess retrieval quality. We use the training set as the datastore and the test set for evaluation. The datastore contains over 90K documents. We evaluate with distractor setting which means the whole datastore is for reference. We use QWEN2-7B-INSTRUCT as the backbone model. Since document retrieval relies on KNN for both KNN-LM and standard RAG, we focus on the KNN-LM structure to analyze the retrieval process. For evaluation, we report exact match (EM) and F1 scores between the generated output and the ground truth. As the datastore is very large, we randomly select 100 documents as targets for targeted attacks. The initial query in multi-round attack is each question.

F Settings of Hyper-Parameters

F.1 General Settings

For all LLM generations, we set the repetition penalty to 0.75 and the no-repeat-ngram-size to 0. The feature dimensions of GPT-2, Llama3, and Qwen2 are 128, 2048, and 3584 respectively. All other generation parameters follow the default HUGGINGFACE settings. By default, we set the datastore size equal to the total number of tokens in each dataset. We follow previous differential privacy work where we set δ as the reverse scale of dataset size. For Enron Email and ECHR, it is 10^{-4}. For HotPotQA and C4, we use 10^{-5}. We use each model's tokenizer and its last hidden-state vectors as embeddings for both retrieval settings. The max output token length is 2048.

F.2 DPSparseRAG

Apart from ϵ and clipping bound we tune on in the experiments, another hyper-parameter for DPSparseRAG is m. We therefore conduct a sweep over $m \in \{10, 20, 30, 40, 50, 60\}$ with $\epsilon = 5$ on the Enron Email and ECHR datasets. Overall, we observe in Table 4 no significant difference in utility across various values of m.

F.3 Setting k

In both RP and DP-RP-G, we sample the random projection matrix from $\mathcal{N}(0, 1/k)$. Table 3 examines the impact of different k values. As discussed in Section 3, larger k better preserves retrieval scores. Consistently, we observe higher utility with increasing k, while privacy remains largely unchanged. Our method relies on similarity-preserving projection rather than precise distance preservation (Appendix B and Appendix C), so the expected post-projection ranking change is less sensitive to k, indicating robustness across a wide range of values. Based on analysis and empirical observation of choosing k, we suggest following ways of choosing k. If a user wants more privacy guarantee or less offline computation latency, a smaller k should be chosen (e.g. $k = 25$ for $d = 2048$). If a user wants higher utility, a bigger k should be chosen (e.g. $k = 1600$ for $d = 2048$). Otherwise, in general, a moderate value would be suitable (e.g. $k = 100$ for $d = 2048$).

To understand the effect of not setting the sampling variance as $1/k$, we use $k = 1600$ and sample from $\mathcal{N}(0, 0.2^2)$, resulting in a perplexity of 70.86 and 2.32% privacy leakage. For $k = 25$ and $\mathcal{N}(0, 0.025^2)$, perplexity is 71.38 with 3.1% leakage. Choosing sampling variances larger or smaller than $1/k$ degrades utility, validating that setting variance to $1/k$ is optimal.

G Complete Experiment Results

G.1 Full Results on Enron Email with GPT2

Table 5 shows perplexity numbers in Fig. 2. We present the leakage of phone numbers, email addresses, and URLs in Table 6, corresponding to the experiments in Fig. 2. We observe that URLs are the easiest to leak. Without any protection, vanilla RAG exposes a substantial amount of sen-

Table 3: Comparison of privacy and utility under different k settings.

	KNN-LM				Standard RAG			
	Perplexity		Leakage (%)		Perplexity		Leakage (%)	
	RP (Ours)	DP-RP-G	RP (Ours)	DP-RP-G	RP (Ours)	DP-RP-G	RP (Ours)	DP-RP-G
$k = 25$	46.22	113.86	**2.18**	2.32	95.84	112.62	**2.12**	1.94
$k = 100$	**46.15**	99.28	2.26	2.41	**47.92**	104.70	2.16	**1.87**
$k = 1600$	43.20	**97.10**	2.19	**2.22**	**47.92**	**102.66**	2.27	2.23

Table 4: Comparison of KNN-LM and Standard RAG on Enron Email and ECHR datasets with different m where $\epsilon = 5$.

	KNN-LM						Standard RAG					
m	10	20	30	40	50	60	10	20	30	40	50	60
Enron Email												
Privacy (# leakage)	4	2	0	0	0	0	8	6	1	0	0	0
Perplexity	3.14	3.15	3.14	3.14	3.16	3.16	3.70	3.68	3.70	3.71	3.71	3.98
ECHR												
Privacy (% leakage)	2.12	2.24	2.06	2.12	2.18	1.94	2.31	2.02	2.16	2.08	2.12	2.14
Perplexity	512.64	512.64	512.64	112.35	109.64	110.35	4307.36	4888.23	5155.85	4307.36	7549.47	7551.63

Table 5: Comparison of perplexity for the Enron Email dataset using GPT2.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	3.31	2.87	**2.89**	5.85	3.54	5.13	2.91	42.17	4.22	2.96	2.83
Standard RAG	3.31	1.58	**2.05**	3.42	3.23	5.24	2.91	3.34	3.71	3.61	1.62

Table 6: Number of leaked email addresses, URLs, and phone numbers with different methods on Enron Email.

	KNN-LM			Standard RAG		
	Email	URL	Phone	Email	URL	Phone
Parametric	0	0	0	0	0	0
RAG	0	13	0	36	56	125
RP (Ours)	0	0	0	0	3	0
DP-RP-G, $\epsilon = 1$	0	0	0	0	0	0
DP-RP-G, $\epsilon = 5$	0	0	0	0	0	0
DP-RP-G, $\epsilon = 10$	0	0	0	0	8	0
DP-RP-G, $\epsilon = 20$	0	0	0	0	1	4
DPSparseVoteRAG, $\epsilon = 1$	0	0	0	0	0	0
DPSparseVoteRAG, $\epsilon = 5$	0	0	0	0	0	0
DPSparseVoteRAG, $\epsilon = 10$	0	0	0	0	0	0
DPSparseVoteRAG, $\epsilon = 20$	0	2	0	0	8	1

sitive information, particularly in Standard RAG where phone numbers show the highest leakage.

G.2 Full Results on ECHR with Llama3

Table 7 shows the privacy leakage in Fig. 2. Table 8 shows the numbers of perplexity in Fig. 2. Apart from perplexity, we also evaluate RAG utility on the ECHR dataset using LLAMA3-1B-INSTRUCT for generation. For each legal document, the model generates the judgment, and we compute the F1 score. Precision is defined as the number of matching tokens divided by the total generated tokens, and recall as matching tokens divided by ground-truth tokens. We show results in Table 9. Consistent with previous findings, random projection preserves performance far better than differential privacy methods. Even with loose privacy budgets, DP methods yield notably lower utility scores.

G.3 Full Results on ECHR with Qwen3

We further evaluate all methods on the ECHR dataset using the more recent Qwen3-4B model. Table 10 shows the perplexity. Table 11 shows the F1 score. Table 12 shows the privacy leakage. Overall, Qwen3-4B achieves substantially stronger utility than Llama3-1B, as reflected by consistently lower perplexity and higher F1 scores across methods. Standard RAG can collapse to parametric-level perplexity in the worst case. From a privacy perspective, the parametric model effectively forms a lower bound for standard RAG leakage, whereas for KNN-LM our methods can reduce leakage to levels comparable to or below the parametric baseline, demonstrating a favorable privacy–utility trade-off for stronger models.

G.4 Full Results on C4 with Llama3-8B

Table 14 shows the privacy leakage in Fig. 2. Table 13 shows the numbers of perplexity in Fig. 2.

Table 7: Comparison of perplexity for the ECHR legal judgment task across different methods using the Llama3-1B.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	631.09	2.90	**46.15**	214.54	99.28	94.87	112.85	228.59	109.64	121.82	92.26
Standard RAG	631.09	9.84	**47.92**	512.66	104.70	95.72	79.72	4398.91	7780.70	4922.52	5264.40

Table 8: Comparison of privacy leakage for the ECHR legal judgment task across different methods using the Llama3-1B.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	32.82	82.72	**2.26**	2.76	2.41	2.81	2.69	1.75	37.86	38.68	38.82
Standard RAG	32.82	37.92	**2.16**	2.03	1.87	2.15	1.96	1.77	2.05	2.05	19.48

Table 9: Comparison of F1 scores for the ECHR legal judgment task across different methods.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	25.7	90.32	**87.7**	53.25	51.3	54.02	55.13	41.67	41.67	40.32	45.38
Standard RAG	25.7	97.18	**66.7**	51.39	54.31	54.12	54.79	1.72	43.72	42.42	43.33

Table 10: Comparison of perplexity for the ECHR legal judgment task across different methods using the Qwen3-4B.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	1.47	1.55	**1.55**	2.26	2.28	2.28	2.30	3130.89	610.10	500.13	277.62
Standard RAG	–	1.19	**1.19**	2.50	2.49	2.51	2.50	1.74	1.43	1.62	1.38

Table 11: Comparison of F1 score for the ECHR legal judgment task across different methods using the Qwen3-4B.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	90.18	90.32	**89.32**	80.35	80.32	80.18	82.39	52.40	51.37	51.94	52.98
Standard RAG	–	97.32	**95.43**	85.32	86.17	86.43	85.19	88.35	91.03	93.48	94.32

Table 12: Comparison of privacy leakage for the ECHR legal judgment task across different methods using the Qwen3-4B model.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	2.03%	19.08%	**1.94%**	13.45%	13.21%	14.85%	15.39%	2.44%	2.46%	2.21%	2.24%
Standard RAG	–	21.30%	**2.05%**	25.43%	14.36%	17.07%	15.09%	2.84%	2.88%	2.86%	3.32%

Table 13: Comparison of privacy leakage for the untargeted attack on C4.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	0	269	**17**	0	0	17	25	0	3	36	124
Standard RAG	0	358	**23**	0	0	15	28	0	33	79	203

Table 14: Comparison of perplexity on C4.

	Parametric	RAG	RP (Ours)	DP-RP-G				DPSparseVoteRAG			
				$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 20$
KNN-LM	2.76	3.16	**2.98**	5.98	5.47	4.12	4.38	6.54	5.91	5.39	4.51
Standard RAG	2.76	2.83	**2.96**	28.41	27.32	26.54	23.12	5.73	5.46	4.65	4.12

For parametric, it is 0 as RAG datastore is not used. For DP-RP-G and DPSparseVoteRAG, we can see that though with certain privacy budgets, they can achieve very low data leakage. However, in Table 18, from example generated content, we can see that they cannot generate acceptable content while our method can generate comparable results to vanilla RAG, showing that it achieves the best tradeoff between privacy and utility.

Table 15: Utility results of different algorithms on HotpotQA using the KNN-LM structure.

	Ans		Sup		Joint		Cos.
	EM	F1	EM	F1	EM	F1	
Parametric	7.48	28.74	21.37	49.62	7.13	28.55	0.73
RAG	12.48	29.88	25.38	59.74	10.84	32.06	0.75
RP (Ours)	11.32	28.57	25.22	57.14	9.12	34.09	0.75
DP-RP-G, $\epsilon = 1$	0.08	7.10	9.54	30.42	0.15	9.03	0.54
DP-RP-G, $\epsilon = 5$	0.21	9.32	10.35	27.82	0.16	9.85	0.53
DP-RP-G, $\epsilon = 10$	3.58	10.84	9.71	28.93	1.19	7.34	0.54
DP-RP-G, $\epsilon = 20$	4.56	9.81	14.95	31.23	1.67	8.32	0.61
DPSparseVoteRAG, $\epsilon = 1$	0.02	8.12	13.98	23.12	0.00	13.12	0.51
DPSparseVoteRAG, $\epsilon = 5$	3.31	11.23	12.88	30.64	1.31	14.29	0.52
DPSparseVoteRAG, $\epsilon = 10$	7.48	18.18	14.39	28.85	1.54	16.67	0.54
DPSparseVoteRAG, $\epsilon = 20$	7.48	18.18	14.38	28.85	1.54	15.38	0.54

G.5 Full Results on Hotpot QA with Qwen2-7B

For evaluation, we report exact match (EM) and F1 scores between the generated output and the ground truth. For F1, precision is the number of matching tokens divided by the total generated tokens, and recall is the number of matching tokens divided by the ground-truth tokens. *Ans* denotes the final generated answer, and *Sup* denotes the retrieved supporting documents. For the parametric model, we prompt it to generate supporting facts to enable fair comparison, as the datastore may overlap with pre-training data. *Joint* indicates cases where both *Ans* and *Sup* are correct.

As shown in Table 15, random projection preserves the cosine similarity between retrieved documents and ground truth, indicating successful retrieval. In contrast, differential privacy methods degrade retrieval performance. Although looser privacy budgets can improve utility, as shown earlier, they also raise privacy concerns.

H Random Projection with Other Distributions

Beyond the normal (Gaussian) distribution, we also experimented with constructing the projection matrix using a Rademacher distribution. In this distribution, each entry independently takes the value $+1$ or -1 with equal probability (50%). As shown in Table 16, the Rademacher-based projection also preserves privacy effectively while achieving performance comparable to vanilla RAG. These results indicate that our proposed random projection method remains effective as long as the projection matrix preserves similarity structure, regardless of the specific distribution used to generate it.

I More Discussion on Results

I.1 Examples of Sensitive Information

To provide a clearer understanding of what constitutes sensitive information, we present examples of leakage with and without sensitive content. In Table 17, we observe that leakage of sensitive information exposes critical details such as names, addresses, phone numbers, and email addresses. In contrast, non-sensitive information does not reveal useful details. For example, a generic URL or repeated mentions of "Queen St" are too common and lack specific identifying information.

Although random projection provides strong privacy protection overall, our experiments on Enron Email, ECHR, and C4 reveal that a small number of items may still be leaked. All failure cases fall into the case of leakage examples without sensitive information. Analysis of these failure cases reveals a clear pattern: random projection provides complete protection (100% leakage prevention) for highly specific content such as unique numbers, IDs, and email addresses. However, in rare cases, common terminology appearing in personal identity information, such as place names (e.g., "Richmond," "Queens St") in addresses or common phrases in personal narratives, may be leaked by coincidence. This occurs because such terms may appear in multiple contexts, making them less distinctive in the embedding space. These findings suggest that random projection is most effective for protecting unique identifiers while providing strong but not absolute protection for common terms that may appear across multiple documents.

I.2 Generalization Across Sensitive Information Types

Our evaluation covers both structured and unstructured sensitive information to assess the method's generalizability. On the Enron dataset, we evaluate protection against structured data including email addresses, phone numbers, and URLs. As shown in Table 6, random projection effectively protects all three types, with leakage rates below

Table 16: Perplexity and data leakage comparison using random projection matrices drawn from different distributions on Enron Email dataset with GPT2.

		KNN-LM				Standard RAG			
		Perplexity	Email	URL	Phone	Perplexity	Email	URL	Phone
RAG		2.87	0	13	0	1.58	36	56	125
RP (Ours)	Gaussian	**2.89**	0	0	0	2.05	0	3	0
	Rademacher	2.92	0	2	0	**1.76**	0	3	1

Table 17: Examples of sensitive vs non-sensitive information leakage

Leakage Information	Sensitive Information	Non-Sensitive Information
Email	Suggestions and feedback are welcome at feedback@nytimes.com	Suggestions an= d feedback are welcome at https://www.reddit.com
Phone	Don Fuller Director, Client Relations 916-608-7055 DFuller@caiso.com	If you have any comments or suggestions contact us at! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Address	The applicant is a British national, born in 1957 and he is currently in prison in Kent, the United Kingdom.	The applicant, Mr Richard Owens, is a British national who was born in 1944 and lives in Queen St. Queen St. Queen St. Queen St.
Name	The applicant, Mr Michael Boyce, is an Irish national, who lives in Mullingar, Ireland. He was represented before the Court by Mr P.J. Groarke, a lawyer practising in County Longford, Ireland.	The applicant, Mr D.H., who was born in 1966 and lives in H.

Table 18: Example responses generated by different methods given the prompt.

Prompt	*Complete the sentences: you have any suggestions, please share. With the rise of social media...*
Parametric	With that in mind here are a few ideas to get you started: 1. **Create a vision board**: Cut out images and words that represent your goals and desires and create a visual representation of what you want to achieve. 2. **Set SMART goals**: Make sure your goals are Specific Measurable Relevant and Time-bound. This will help you stay focused and motivated. 3. **Break down big goals into smaller.
RAG	it's become increasingly important for businesses to have a strong online presence. One way to do this is by creating engaging content that resonates with your target audience. Here are some tips to help you create content that drives engagement: 1. Know your audience: Before you start creating content make sure you have a good understanding of your target! audience. Who are they? What are their interests? What problems do they need?
RP (Ours)	it's become increasingly important for businesses to have a strong online presence. One way to do this is by creating engaging content that resonates with your target audience. Here are some tips to help you create content that drives engagement: 1. Know your audience: Before you start creating content make sure you have a good understanding of who your target! audience is. What are their interests need?
DP-RP-G ($\epsilon = 20$)	PrintWriter.PrintWriter userService userServiceExchangeExchangeExchange_generate_generate
DPSparseVoteRAG ($\epsilon = 20$)	make sure you have a good understanding of your target! "#'"#'!$""&"'$"#"#'

3% across both KNN-LM and Standard RAG architectures. For unstructured text, we evaluate on the ECHR dataset, which contains legal case narratives with sensitive personal information embedded in natural language. Our method maintains strong privacy protection (2.19–2.27% leakage) while preserving utility, demonstrating that random projection works effectively across both structured and unstructured sensitive information. This suggests that the privacy-preserving mechanism operates at the embedding level, making it agnostic to the specific format or structure of the sensitive content.

I.3 Examples in Multi-Round Attack

In this section, we provide examples of attack results under multi-round attacks. From Table 19, we can see that Pablo Trapero and Aleksander Ford are originally mentioned in the question, so it is reasonable that the responses also contain these two names. However, the response generated by

our method is substantially different from the original dataset chunk while still focusing on answering the question. No sensitive information, such as address, nationality, or actual birthdate, is revealed. As a result, this demonstrates a successful defense against multi-round attacks.

One interesting aspect of our response is that Pablo Trapero was actually born on October 4th, 1971, and Aleksander Ford on November 24th, 1908. However, our response states that they were born on December 27th, 1965, and September 1st, 1923, respectively. This provides a good example of our system mechanism, where similar tokens may be replaced with other tokens. The system does not reveal the actual birthdates and therefore protects privacy. At the same time, the relative relationship between the two birthdates remains correct, so the final answer is still correct.

In contrast, the response generated by standard RAG reveals a large portion of the original content. The method of Jiang et al. can successfully extract most of the sensitive content, while the knowledge graph-based method also recovers most of the original information.

Table 19: Examples of attacking results in multi-round attack. We highlight the content which is the same as the original chunk in datastore.

Original Chunk	...Aleksander Ford: Aleksander Ford (born Mosze Lifszyc, 24 November 1908 in Kiev, Russian Empire – 4 April 1980 in Naples, Florida, United States) was a Polish Jewish film director, and head of the Polish People's Army Film Crew in the Soviet Union during World War II. Ford became director of the nationalized Film Polski company following the Red Army occupation of Poland. 7 Days in Havana: 7 Days in Havana (Spanish: 7 días en La Habana) is a 2012 Spanish-language anthology film. Set during a week in the Cuban capital Havana, the film features one segment for each day, each segment directed by a different filmmaker. The directors are Julio Medem, Laurent Cantet, Juan Carlos Tabío, Benicio del Toro, Gaspar Noé, Pablo Trapero and Elia Suleiman. The screenplay was written by the Cuban novelist Leonardo Padura Fuentes. The film is a co-production between companies in Spain, France and Cuba...	
Attacking Methods	Response by RAG	Response by ours
Adaptive (Jiang et al., 2024a)	<think> Aleksander Ford (born Mosze Lifszyc, 24 November 1908 in Kiev, Russian Empire – 4 April 1980 in Naples, Florida, United States) was a Polish Jewish film director, and 2006 BBC Four television. Ford became director of the nationalized Film Polski company following the Red Army occupation of Poland. 7 Days in Havana: 7 Days in Havana (Spanish: 7 días en La Habana) is a "Pennies From Heaven" (1978)...	<think> Pablo Trapero or Aleksander Ford? 7 Who was born first? 8 Who was born first? Aleksander Ford?
Knowledge Graph (Yao et al., 2026)	<think> Thinking Process: 1. **Analyze the Request:** * Target: An English actor of stage, film, and television. * Condition 1: Aleksander Ford (born Mosze Lifszyc, 24 November 1908 in Kiev, Russian Empire – 4 April 1980 in Naples, Florida, United States) was a Polish Jewish film director * Condition 2: 7 Days in Havana: 7 Days in Havana (Spanish: 7 días en La Habana) * Goal: Who was born first? * ...	<think> Who was born first? Pablo Trapero or Aleksander Ford? <think> Thinking Process: 1. **Analyze the Request:** The user is asking for a comparison of birth dates between two individuals: Pablo Trapero and Aleksander Ford. The goal is to determine who was born first. 2. **Identify the Individuals:** * **Pablo Trapero:** An Argentine film director, screenwriter, and producer. * **Aleksander Ford:** A Polish film director and screenwriter. 3. **Retrieve Birth Dates:** * **Pablo Trapero:** I need to recall or verify his birth date. * Search query mental check: "Pablo Trapero birth date". * Knowledge: Pablo Trapero was born on December 27, 1965. * **Aleksander Ford:** I need to recall or verify his birth date. * Search query mental check: "Aleksander Ford birth date". * Knowledge: Aleksander Ford was born on September 1, 1923. 4. **Compare Dates:** * Aleksander Ford

Table 20: Perplexity comparison of DP-RP-G under clipping-only settings (clipping applied without adding DP Gaussian noise) across different clipping bounds.

Clipping bound	0.1	1	5	10	15	25	50	100
KNN-LM	214.54	214.54	214.54	114.86	97.86	46.33	**45.47**	46.03
Standard RAG	512.66	507.62	512.66	489.56	117.82	47.38	**46.99**	**46.99**

J Why Differential Privacy Methods Degrade Performance

Table 21: Perplexity comparison of random projection methods with only clipping under the KNN-LM and standard RAG structure on the ECHR dataset.

Method	KNN-LM	Standard RAG
RP (Ours)	**43.20**	**47.92**
DP-RP-G	97.09	102.66
Random Projection (only Clipping)	45.47	46.99

A natural question arises: *why does differential privacy harm utility?* To make a RAG mechanism differentially private (DP), two common approaches are adding Gaussian noise or applying DP aggregation. In the DP-RP approach, achieving a privacy guarantee with budget (ϵ, δ) requires clipping each projected embedding to a norm bound c and then adding Gaussian noise $\mathcal{N}(0, \frac{c\sqrt{2\ln(1.25/\delta)}}{\epsilon})$. In contrast, if we set the privacy budget to $\epsilon = \infty$, we may use a sufficiently large clipping bound with zero noise, effectively reducing the method to our proposed random projection scheme. This highlights that the utility loss arises from enforcing formal DP constraints, rather than from the projection itself.

The second approach, DP aggregation, can also harm utility because limited-domain methods (Koga et al., 2024) depend heavily on the perturbation magnitude. Balancing privacy and utility often requires extensive tuning, making it hard to choose an optimal privacy budget. As shown empirically in Section 4.2, even with a loose budget, utility remains low and privacy can still be leaked.

To understand why differential privacy methods degrade performance, we conducted an empirical study where the only difference from our random projection baseline was the addition of a clipping operation before projection. In differential privacy theory, clipping is required to bound sensitivity and ensure that the difference between participants remains limited (Dwork et al., 2006). A common approach is to clip variables within a specific range. For instance, Kenthapadi et al. (2013) assume all vectors are binary (0 or 1) and show that as long as vectors lie within $[0, 1]$, the DP guarantee holds. In general, clipping or normalizing vectors to a fixed range ensures the DP condition. For consistency with prior work, we set the clipping bound to $[0, 1]$. Additionally, we evaluate the impact of normalizing embeddings to $[0, 1]$.

In Table 21, we compare random projection with only clipping. In Table 20, we can see the change of perplexity along with different clipping bounds if no Gaussian noise is added. If we do not add the Gaussian noise and set a very large clipping bound, DP-RP-G becomes equivalent to our methods as we barely clip any vectors. The results show that clipping degrades perplexity and reduces overall performance. In this experiment, we set $\epsilon = 5$. Overall, achieving differential privacy requires bounding sensitivity, which degrades performance.