Toward Privacy-preserving Text Embedding Similarity with Homomorphic Encryption

Donggyu Kim*[1] Garam Lee*[2] Sungwoo Oh*[1]
KB Kookmin Bank[1], CryptoLab[2]
{donggyukimc, david.oh0126}@gmail.com, garamlee@cryptolab.co.kr

Abstract

Text embedding is an essential component to build efficient natural language applications based on text similarities such as search engines and chatbots. Certain industries like finance and healthcare demand strict privacy-preserving conditions that user's data should not be exposed to any potential malicious users even including service providers. From a privacy standpoint, text embeddings seem impossible to be interpreted but there is still a privacy risk that they can be recovered to original texts through inversion attacks. To satisfy such privacy requirements, in this paper, we study a Homomorphic Encryption (HE) based text similarity inference. To validate our method, we perform extensive experiments on two vital text similarity tasks. Through text embedding inversion tests, we prove that the benchmark datasets are vulnerable to inversion attacks and another privacy preserving approach, $d\chi$-privacy, a relaxed version of Local Differential Privacy method fails to prevent them. We show that our approach preserves the performance of models compared to that the baseline has degradation up to 10% of scores for the minimum security.

1 Introduction

Recently, various industries provide enhanced user experiences through natural language processing (NLP) applications. AI assistants such as Amazon's Alexa and Google Assistant are representative examples that help users to achieve their purposes with a wide range of intentions. To build such complex applications, it is common to utilize machine-learned text representations, i.e., text embeddings to infer similarities between texts (Cer et al., 2018). Text embeddings facilitate the efficient implementations of various NLP functions like document search (Karpukhin et al., 2020), intent decision (Humeau et al., 2020), and dialogue response selection (Gu et al., 2020) by leveraging precomputed embeddings for real-time applications. However, such usage of text embeddings poses emerging privacy risks so-called *inversion attacks* that recover the original texts from embeddings (Song and Raghunathan, 2020).

No.	Query Text
1	***I'm 13***. Can I buy supplies at a pet store without a parent/adult present?
2	I earn ***$75K***, have ***$30K in savings, no debt, rent from my parents*** who are losing home. Should I buy home now or save?
3	How do I ***fold side-income into our budget*** so my husband doesn't know?

Table 1: Examples of query text containing sensitive information from FIQA-2018 dataset. Sensitive texts are marked with red color.

User texts such as *Know-Your-Customer*[1] inquiries in the finance domain frequently contain privacy-sensitive data. The sensitive data include not only personal information which can identify users, but also their assets and clues or intentions about their future behaviors (Wheatley et al., 2016; Schwartz and Solove, 2011). Table 1 shows example texts with information that causes infringements on user's privacy if they are leaked to unauthorized users. We define malicious users without authorization for user's privacy information into two categories. First, external malicious users perform attacks from outside of services by accessing data or servers. Second, in certain domains that require strict privacy preservation such as finance, the data access from internal malicious users even including service providers should be prevented.

In this paper, we propose Homomorphic

*Equal contribution

[1]https://en.wikipedia.org/wiki/Know_your_customer

Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), pages 25 - 36

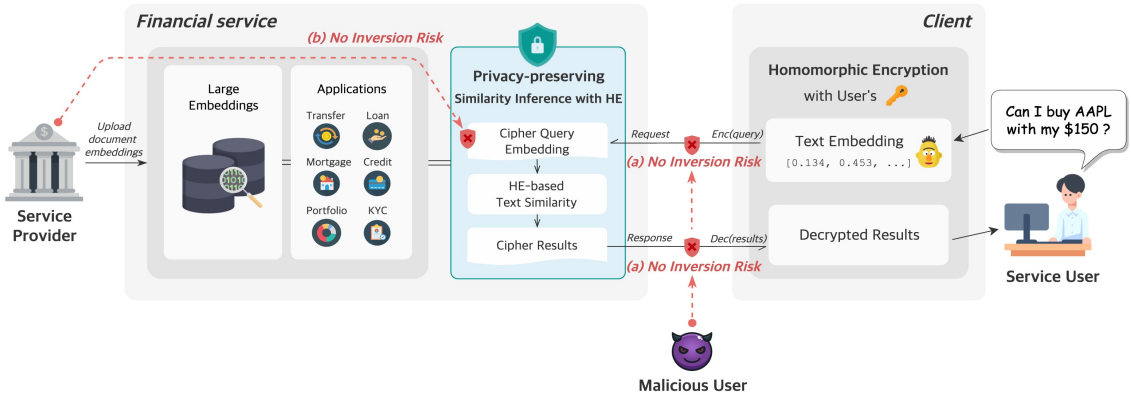

Figure 1: An example of text embedding based finance service with privacy preservation. Our Homomorphic Encryption method protects user data from *(a) external* and *(b) internal* malicious user attacks.

Encryption (HE) based text similarity inference secure from inversion attacks by both external and internal malicious users. It is possible to satisfy such rigorous privacy-preserving mechanism because HE approach enables all computations to be performed without decryption of the data (Cheon et al., 2017). Other cryptographic technologies do not meet the requirements because they need server-side decryption for computation (Acar et al., 2018). Another candidate method to resolve the above problem like d_χ-privacy, a variant of Local Differential Privacy (LDP) should consider a privacy-utility trade-off (Qu et al., 2021) in our tasks.

Figure 1 shows an example of text embedding based financial services using our HE method to protect user's query embedding from inversion attacks. First, a large number of server-side text embeddings such as documents for search were precomputed and uploaded to a centralized server in advance. Here, we assume that server-side text embeddings are not encrypted since service providers can access the database. Service users generate a public key and a secret key for homomorphic encryption. Then they convert their query texts into embeddings and encrypt them using HE with their public key. When the users send the encrypted query embedding to the server, no external malicious user can access the original data because of encryption. As a result, we can protect an inversion attack at *(a)*. Once the encrypted query embeddings reach the server, services can perform inference without decrypting the data due to our HE-based similarity function. During the inference, the service provider cannot extract any information from encrypted data because the HE secret key is owned by the user only. Therefore, the inversion attack point *(b)* is secure. Finally, the server sends the still encrypted result securely, and the user decrypts the result with the secret key.

We perform extensive tests using well-known benchmarks on two text similarity tasks, semantic textual similarity and text retrieval. The results on inversion attacks indicate that text embeddings can be easily recovered to original texts. Furthermore, we observe our d_χ-privacy baselines are not suitable to prevent such attacks completely while maintaining the performance of models. Specifically, it loses up to 10% of scores at minor noise settings and still shows information leakage. In contrast, our method guarantees the protection from inversion attacks and do not hurt performances. To summarize, our contributions are:

- We demonstrate that well-known bechmarks and pretrained text embedding models are vulnerable to inversion attacks.
- We implement HE based text similarity functions that can precisely approximate original performance while preventing any potential information leakage.
- Through extensive experiments, we prove that our method achieves complete privacy-preserving similarity tasks without hurting the performance.

2 Related Work

2.1 Text similarity with embeddings

Measuring text similarity is a fundamental functionality for many NLP applications. To overcome the limitation of lexical matching (Robertson and Zaragoza, 2009) such as TF-IDF and BM25, it is common to convert natural language text into text embeddings (Reimers and Gurevych, 2019) capturing the semantic meaning of texts as a form of vectors because it can represent rich contextual information. Using text embeddings, the similarity between texts can be interpreted as distances between data points in a vector space. These properties facilitate efficient computations of large-scale text similarity inference because embeddings can be precomputed and used in real-time applications without inference considering many parameters (Karpukhin et al., 2020). In practice, large amounts of texts such as search documents are precomputed whereas real-time data from users such as short search-queries require the embedding process on the fly. The relevancy between query and documents can be calculated with similarity functions such as cosine similarity or dot product. Formally (1), given text embeddings for query and document, E_q and E_d, the similarity between query q and each document d is computed with a similarity function:

$$sim = funct(E_q(q), E_d(d)) \tag{1}$$

2.2 Privacy-preserving in NLP

Although homomorphic computation basically takes numerical data as its input, much recent research shows attempts to apply HE to text data (Lee et al., 2022; Chen et al., 2022). However, these works mainly consider encrypted *classification* tasks on text embeddings. In this study, we focus on the text embedding based text similarity applications. Compared to classification task settings, the service scenario using text similarity is more suitable to take the advantage of using HE. This is because huge text embeddings are stored in a centralized server and users need to send query texts to the server to get inference results. In the process of it, they want their queries, which may contain sensitive information, not to be exposed to the server and still receive a response as expected.

The authors in (Feyisetan et al., 2020) proposed d_χ-privacy for privacy-preserving approach on textual data. However, their method requires to select a privacy parameter ϵ very carefully. Our baseline experiment results using d_χ-privacy showed low performance. The work in (Xiong et al., 2022) proposed how to evaluate a privacy risk on text data using semantic correlation. Our HE-based method using CKKS provides a practically complete security in terms of this privacy risk assessment since it ensures a 128-bit level of security and no information leakage occurs without decryption using a secret key. Other prior HE-based works such as (Yu et al., 2017) and (Nautsch et al., 2018) do not compute cosine similarity directly because the HE schemes they use do not support bootstrapping.

3 Method

In this section, we propose our Homomorphic Encryption (HE) based method to protect text data privacy. To achieve this, our goal is to approximate text similarity functions for given text embeddings in an encrypted state using the CKKS scheme. Formally, similar to the definition in (1), we implement encrypted similarity function $funct_*$ that computes the encrypted similarity result sim_* from encrypted text embeddings in order to achieve $sim_* \approx sim_{(1)}$ as much as possible.[2]

$$sim_* = funct_*(Enc(E_q(q)), E_d(d))$$

3.1 Homomorphic Encryption : CKKS Scheme

Homomorphic Encryption is a cryptographic primitive that can support computations on encrypted data without decryption. After performing computations in encrypted state, the decrypted output is the same as if we performed the computations in plaintext.

We adopt the CKKS scheme (Cheon et al., 2017, 2018a, 2019) that supports *approximate* arithmetic operations over encrypted real-valued vectors. While other HE schemes

[2] Asterisks(*) indicate a ciphertext or a computation in ciphertext.

such as BGV (Brakerski et al., 2011) and BFV (Fan and Vercauteren, 2012) can be applied for computations over integers, the fourth generation HE scheme, CKKS supports encrypted computations over real and complex numbers. This advantage provides scalability of encrypted computation to many applications in the real world. More details of the CKKS scheme can be found in (Cheon et al., 2017).

CKKS is a *leveled* HE scheme (Lee et al., 2022). This implies that a given ciphertext has a bounded depth to perform operations; the number of operations we can perform repeatedly is limited due to noise increase in computation. If we multiply two ciphertexts of level l, the output is a ciphertext of level $l-1$, which means the remained number of operations is reduced by 1. For this reason, we need a unique operation called *bootstrapping* to resolve this level reduction. The bootstrapping operation refreshes a ciphertext increasing its level higher so the number of possible operation times increases. The following HE operations are available over ciphertexts of given real-valued vectors pt_1 and pt_2 in plaintext.

- Add(ct_1, ct_2): output a ciphertext of pt_1+pt_2, where $+$ is the slot-wise addition.
- Mult(ct_1, ct_2): output a ciphertext of $pt_1 \odot pt_2$, where $\odot$ is the slot-wise multiplication.
- Bootstrap(ct_1): output a ciphertext of pt_1 at refreshed level.

In addition, it is worth to note that homomorphic operations can be performed on a plaintext and a ciphertext together as the operands of operations (Carpov and Sirdey, 2015). We can take the advantage of a plaintext-ciphertext operation because the noise increase is less than that of between both ciphertext operation. This flexibility enables us to consider various user scenarios depending on what to be protected.

For our tasks, we adopt the cosine similarity as our relevance score. Recall the cosine similarity of two vectors is defined as follows:

$$\begin{aligned}\cos\theta &= \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \\ &= \frac{u_1v_1+\cdots+u_nv_n}{\sqrt{(u_1^2+\cdots+u_n^2)\times(v_1^2+\cdots+v_n^2)}}\end{aligned}$$

where u and v are n-dimensional vectors and θ is the angle between them, which indicates how close they are. Since HE supports addition and multiplication only, it is essential to approximate an arbitrary operation with an appropriate polynomial. In our task, the approximation we need is the square root inverse function. To implement this, we apply Newton's method (Panda, 2021) of the following form to approximate the square root inverse in an encrypted state.

$$y_{n+1} = \frac{1}{2}y_n(3-xy_n^2)$$

The input domain of the function is $1 \le x \le 2^{22}$ and precision is 3×10^{-7}. For each iteration, the polynomial equation is updated recursively. Note that the function converges with an initial value y_0 satisfying $|1-xy_0^2|<1$. Here is a brief error analysis of the approximation:

$$\begin{aligned}1-xy_{n+1}^2 &= \frac{1}{4}xy_n^2(3-xy_n^2)^2 \\ &= (1-xy_n^2)^2(1-\frac{xy_n^2}{4}) \\ &\vdots \\ &= (1-xy_0^2)^{2^{n+1}}\prod_{k=0}^{n}(1-\frac{xy_k^2}{4})^{2^{n-k}}\end{aligned}$$

where n denotes the number of iterations.

Inference In a real-world scenario for HE based similarity inference, the workflow requires the procedure for en/decryption of data. Procedure 1 describes how a client and a server can communicate in the process of a document search service while achieving privacy-preserving. One might concern that the most relevant document index with decrypted at the end might imply information about the query. To resolve this concern, a client can generate random indices and send the target index with them to the server.

Security Lastly, we emphasize that our HE parameters ensure 128-bit security level, which implies 2^{128} operations are required to recover the plaintext from a ciphertext with the current best algorithm (Cheon et al., 2022). Thus, a homomorphically encrypted ciphertext is securely protected and cannot be revealed without access to the secret key for decryption.

Procedure 1 Find most relevant document

Initialize

D //*service documents for search*
E_d, E_q //*text embedding models*
$funct_*$ //*HE based similarity function*
$D_{emb} \leftarrow E_d(D)$

Client

1: Generate a public key pk and a secret key sk
2: $Q_{text} \leftarrow$ *User input query text*
3: $Q_{emb} \leftarrow E_q(Q_{text})$
4: $Q_{emb*} \leftarrow Enc_{pk}(Q_{emb})$
5: $Q_{emb*}, pk \rightarrow Server$

Server

6: **return** $sim_* \leftarrow funct_*(Q_{emb*}, D_{emb})$ with pk

Client

7: $sim \leftarrow Dec_{sk}(sim_*)$ with sk
8: $index \leftarrow argmax(sim)$
9: $index \rightarrow Server$

Server

10: **return** $document \leftarrow D[index]$

4 Experiments

4.1 Text Similarity Tasks

To evaluate our approach, we consider two text similarity task settings: **STS** (Semantic textual similarity) and **Text retrieval**. We provide the brief descriptions on the tasks.

- **STS (Semantic textual similarity):** The task assesses the ability to inference the semantic similarity of given text pairs. Specifically, we measure the correlation between ground truth labels judged by human, and similarity scores predicted by models. Following previous studies (Reimers and Gurevych, 2019), we consider a set of seven well-known semantic textual similarity datasets, STS 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). Each dataset has a set of text pairs and the corresponding ground truth labels indicating semantic relevances. We compute the cosine similarity between text embeddings and measure the correlation between the similarities and the ground truth labels. Following Gao et al. (2021), we utilize Spearman's correlation evaluation script from the SentEval toolkit[3] (Conneau and Kiela, 2018).

- **Text Retrieval:** The task computes list-wise relevance scores i.e. dot product between a query and documents to be searched. The documents are sorted according to the scores and the task assesses text retrieval quality based on the rank of correct documents. Following the recent works (Gao and Callan, 2021; Santhanam et al., 2022), we evaluate text retrieval performances with the BEIR benchmark (Thakur et al., 2021), which aims to evaluate zero-shot retrieval performance of text embedding models. We consider five datasets: FiQA-2018, NFCorpus, ArguAna, SCIDOCS, and SciFact. Each dataset contains domain-specific text data. For instance, FiQA-2018 consists of finance search queries which are representative examples of privacy-sensitive texts.

We use publicly open text embedding models without additional fine-tuning to demonstrate that our approach can be applied generally to any existing text embedding models. For STS and text retrieval, we use *SimCSE*[4] and *DistilBERT*[5] checkpoints from huggingface transformers (Wolf et al., 2020) as our backbone models, respectively. More details about evaluation settings can be found in Appendix A.

4.2 Privacy-Preserving Baseline

1. **Plaintext:** The results from text embeddings without privacy-preserving schemes are obvious counterparts to be compared with privacy-preserved ones. In the rest of this paper, we denote them as *plaintext*. The common objective of our method and other privacy-preserving baselines is to precisely approximate the performances of *plaintext* while preventing the exposure of original information.
2. **d_χ-privacy:** Following Qu et al. (2021) and Lee et al. (2022), we consider d_χ-privacy, which is a relaxed variant of noise-based local differential privacy (LDP) methods as our baseline. The method prevents information leakage of text embeddings

[3] https://github.com/facebookresearch/SentEval

[4] https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased

[5] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v3

	SentEval							BEIR benchmark				
	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	FiQA-2018	NFCorpus	ArguAna	SCIDOCS	SciFact
Plaintext	**0.1617**	**0.1381**	**0.1493**	**0.1496**	**0.1033**	**0.2489**	**0.1325**	**0.6281**	**0.4731**	**0.0797**	**0.1556**	**0.6098**
$\eta = 175$	0.0492	0.0546	0.0466	0.0441	0.0321	0.0699	0.0232	0.5718	0.3237	0.0694	0.1132	0.4806
$\eta = 150$	0.0407	0.0505	0.0425	0.0392	0.0318	0.0592	0.0262	0.5715	0.3175	0.0638	0.1154	0.5101
$\eta = 125$	0.0333	0.0406	0.0344	0.0296	0.0254	0.0465	0.0178	0.5603	0.3225	0.0560	0.0984	0.4756
$\eta = 100$	0.0248	0.0280	0.0235	0.0211	0.0152	0.0350	0.0114	0.5060	0.2245	0.0494	0.0823	0.4175
$\eta = 75$	0.0139	0.0141	0.0115	0.0101	0.0075	0.0190	0.0040	0.4347	0.1827	0.0375	0.0558	0.2844
$\eta = 50$	0.0031	0.0027	0.0019	0.0016	0.0017	0.0049	0.0007	0.3204	0.0910	0.0249	0.0283	0.1439

Table 2: **Performance of Text Embedding Inversion.** Black-box inversion on text embeddings with text data from SentEval and BEIR benchmark. We report the F1 scores of multi-label classifiers predicting words in original text from given text embeddings.

	SentEval							BEIR benchmark				
Name	STS12	STS13	STS14	STS15	STS16	STSB	SICK-R	FiQA-2018	NFCorpus	ArguAna	SCIDOCS	SciFact
#Texts	3,108	1,500	3,750	3,000	1,186	1,379	4,927	648	323	1,406	1,000	300
#Avg words	6.33	6.48	6.09	5.88	5.96	5.76	4.94	6.48	2.59	108.38	7.80	9.12

Table 3: **Statistics of evaluation text data.** #Texts indicates the number of sentence pairs and queries. #Avg words show the number of average words per sentence and query.

through the noise injection privatization. For a given embedding x and sampled noise N, the privatized embedding is $P(x) = x + N$. We sample $N \in \mathbb{R}^n$ by $N = rp$ where r is sampled from the Gamma distribution $\Gamma(n, \frac{1}{\eta})$ and p is sampled from the uniform distribution $\mathbb{B}^n$. Same as Lee et al. (2022), we measure performances at six noise levels (η = 175, 150, 125, 100, 75, 50). Lower η indicates higher noise to embeddings and better privacy-preserving.

4.3 Text Embedding Inversion

We investigate inversion risk existing in text similarity tasks. Song and Raghunathan (2020) suggests two methods for embedding inversion attack, namely, white-box and black-box inversion. We choose **black-box inversion** since it assumes that an attacker only can access text embeddings but no access to model itself. This property is suitable for our privacy-preserving concerns on the applications which utilize precomputed embeddings for text similarity inference. Black-box inversion, in a nutshell, trains a multi-label classifier which takes text embeddings as inputs and predicts words in original texts.

$$\max_{\phi} \sum_{s \in S} \sum_{w \in W} \log p_{\phi}(w|E(s))$$

Formally, for any pretrained text embedding model E, we train an inversion model ϕ by maximizing the log-likelihood where S and W are a set of training sentences and a set of words in a sentence, respectively.

Implementation As an inversion model, we use a simple 1-layer MLP which shows enough performance to extract meaningful information from given text embeddings in our test. To train the inversion model, we sample sentences from BookCorpus (Zhu et al., 2015) and take the train-split texts from benchmark datasets. To choose the best checkpoints and the thresholds for classifiers, we also make the validation data by using BookCorpus and the development split of benchmark datasets. For more detail settings on text embedding inversion test, see Appendix B.

Result Table 2 shows the performances (F1 measurement) of inversion models. We measure F1 scores for the extracted words filtered by the best threshold selected by validation data. Note that our HE approach is not included in the test because it provides complete security (see Section 3.1). First, we can find the inversion models successfully extract original texts from plaintext embeddings. However, compared to typical classification tasks, the models show poor performances (less than 0.5 point) on overall F1 scores (except for FIQA-2018 and SciFact). This is because the model should perform extreme multi-label classification (Chalkidis et al., 2019) with a large number of classes i.e. the vocab size, which is roughly 20,000 words. We can see that inversion models show worst performance on the ArguAna retrieval dataset, it is because ArguAna consists of search queries much longer than other datasets (presented in Table 3).

Original text from **FIQA-2018**: *15 year mortgage vs 30 year paid off in 15*	
Plaintext	vs, year, mortgag, 30, paid, 15
$\eta = 175$	vs, paid, mortgag, year, loan, 30
$\eta = 150$	vs, mortgag, paid, year, pay, 15
$\eta = 125$	vs, paid, month, bore, 30, pay
$\eta = 100$	vs, mortgag, 30, paid, pay
$\eta = 75$	vs, 30, paid, spare, mortgag, tore
$\eta = 50$	paid, vs, common, pay, hunch
Original text from **STS-B**: *a man is singing and playing a guitar*	
Plaintext	guitar, sing, man, play, fluid
$\eta = 175$	guitar, play, man, banana, trampolin, sing
$\eta = 150$	guitar, play, man, banana, trampolin, afghanistan
$\eta = 125$	guitar, play, banana, man, afghanistan, nadia
$\eta = 100$	guitar, banana, afghanistan, play, nadia, afghan
$\eta = 75$	guitar
$\eta = 50$	-

Table 4: **Result of Text Embedding Inversion with texts from FIQA-2018 and STS-B.** The words with red color are correctly predicted ones.

Meanwhile, inversion models show different overall performances on STS and text retrieval benchmarks. This might be due to two factors, which are: 1) SentEval and BEIR benchmark have different domains of texts i.e. sentences on common domain and search queries for diverse domains such as finance science; 2) most importantly, they use their own text embedding models, *SimCSE* and *DistilBERT*. Even though the overall performance on STS does not reach that of text retrieval, inversion models still extract unneglectable amount of original information. At lowest noise value ($\eta = 175$), the model loses more than half of its performance in the plaintext setting. After that, the results clearly show that the more noise we add the less original information extracted. When a noise reaches the highest value ($\eta = 50$), inversion model shows F1 scores less than 0.01 point on all STS datasets. On the other hand, for text retrieval, the model still has moderate performances (greater than 0.3 point at most).

Qualitative Analysis We analyze two examples of text embedding inversion in order to provide an qualitative analysis. We bring two examples of texts from FIQA-2018 and STS-B shown in Table 4. We enumerate extracted words up to Top-6 words ordered by their likelihood scores. We set the number of top words based on the analysis from Table 3, which shows the number of average tokens on most datasets is about 6. We first see the example from FIQA-2018. From the plaintext embedding, the inversion model successfully extracts a set of words (*vs*, *year*, *mortgag*, *30*, *paid*, *15*) that represent the semantic information of original text and have no false positive words. After we add lowest noise ($\eta = 175$) to the embedding, the model starts to confuse semantically related words (*mortgag* $\rightarrow$ *loan*, *paid* $\rightarrow$ *pay*). At highest noise ($\eta = 50$), the model starts to extract completely unrelated words such as *common*, *hunch*. We can observe similar patterns on the example of STS-B. The inversion model extracts all important words (*guitar, sing, man, play*) and only one false positive word (*fluid*) from the plaintext embedding. After we maximize the noise, the model failed to extract any words (filtered by a threshold). By using d_χ-privacy, it is possible to alleviate the embedding inversion but not enough to prevent it completely. These results demonstrate how vulnerable text embeddings are in terms of potential information leakages. For more examples on other datasets, see Appendix C.

4.4 Text Similarity Evaluation

Implementation We implemented HE methods with the full residual number system (RNS) of the CKKS scheme (Cheon et al., 2018b) that supports bootstrapping on GPU. We utilized approximation of the square root inverse with a vector-vector multiplication in STS and a vector-matrix multiplication to compute dot products in Text Retrieval. Note that we only encrypt query texts in retrieval settings since we suppose a situation where documents are open to the public and need not to be protected. Similar to computing cosine similarity, other similarity functions like dot product can be easily implemented with additions and multiplications in an encrypted state. For detailed descriptions of our HE parameters, refer to Appendix D.

Semantic Textual Similarity Table 5 shows the performance results on STS datasets. To accurately validate the approximation performance of our HE method, we report the Spearman's correlation scores displaying floating point numbers up to seven decimal point. At the first step of noise ($\eta = 175$), the noise-based perturbation, d_χ privacy loses about 10% of plaintext performance in average. It shows the largest drop at STS-12 (75.2961809

	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg
Plaintext	75.2961809	84.6670451	80.1894789	**85.3988064**	**80.8192094**	84.1348744	80.3869902	81.5561381
$\eta = 175$	51.7546246	74.0075826	67.1210473	81.6210128	69.9514644	78.9265977	78.1727766	71.6507294
$\eta = 150$	48.2063251	71.9283803	64.6684534	80.4148958	67.3609882	77.2761905	77.0743311	69.5613663
$\eta = 125$	44.3021020	69.2406938	61.6506406	78.6018538	63.8953096	74.9027321	75.2612661	66.8363711
$\eta = 100$	39.9374092	65.3637801	57.6159417	75.5333165	58.9898118	71.1861685	72.0707709	62.9567427
$\eta = 75$	34.4587947	58.1591253	50.9218122	69.1092061	51.0846282	64.1191090	65.5976507	56.2071895
$\eta = 50$	25.1756964	41.6619297	36.5166211	52.4056402	35.6048580	47.1511518	49.2968972	41.1161135
HE (Ours)	**75.2984575**	84.6670451	**80.1894864**	85.3988015	80.8192093	**84.1348781**	**80.3870014**	**81.5564113**
diff. w plaintext	0.0022766	-	0.0000075	-0.0000049	-0.0000001	0.0000036	0.0000112	0.0003277

Table 5: **Performance of Semantic Textual Similarity task.** We report Spearman's correlation scores using the SentEval toolkit. At the bottom of the table, we show the gap between Plaintext results and HE approximations. Values in bold denote better scores.

	FiQA-2018	NFCorpus	ArguAna	SCIDOCS	SciFact
Plaintext	0.2569705	**0.2564896**	**0.4261360**	0.1332835	**0.5378220**
$\eta = 175$	0.2384545	0.2568275	0.4177632	0.1263631	0.5305757
$\eta = 150$	0.2327629	0.2533514	0.4136059	0.1252494	0.4925842
$\eta = 125$	0.2262708	0.2521192	0.4073824	0.1204473	0.5075745
$\eta = 100$	0.2142368	0.2478810	0.3909309	0.1112396	0.4824138
$\eta = 75$	0.1850581	0.2281045	0.3484871	0.0938708	0.4300155
$\eta = 50$	0.1071001	0.1670855	0.2334603	0.0499934	0.2653896
HE (Ours)	0.2569705	0.2564895	0.4259367	0.1332835	0.5378219
diff. w plaintext	-	-0.0000001	-0.0001993	-	-0.0000001

Table 6: **Performance of Text Retrieval task.** We report nDCG@10 scores.

$\rightarrow$ 51.7546246). After the representation of text embeddings are highly collapsed with large noise ($\eta = 50$), the average correlation scores down to the half of the original score (81.5561381 $\rightarrow$ 41.1161135). On the other hand, we can see that our HE method preserves the performance of plaintext almost completely. The method lose scores less than 10^{-5} point from plaintext (at most 0.0000049 point on STS-15). We can also observe the increase of scores at STS-12, STS-14, STS-B and SICK-R datasets. This happens to occur because the noises during encryption may influence in a positive way to compute the scores. As a result, in terms of average score, plaintext and our approach have almost the same scores (less than 10^{-3} point difference between them). In particular, the average absolute deviation between the plaintext cosine similarity scores and the ciphertext cosine similarity scores is from 3.89×10^{-8} in STS15 (lowest) to 5.08×10^{-8} in STS12 (highest).

Text Retrieval Table 5 shows the experimental results on text retrieval datasets. We report nDCG(Normalized Discounted Cumulative Gain) scores. Different from the results on STS datasets, the d_χ-privacy method shows relatively robust performances on text retrieval. We can observe little performance degradation (less than 5%) on most datasets (except for FIQA-2018). Even if we increase noise further, it still maintain small degradation (less than 10%). We think theses differences comes from their evaluation metrics (spearman's correlation and nDCG). Since the correlation measures the difference between ground truth similarities and predicted ones, the noise directly affects the final correlation scores although the noise is small. In contrast, nDCG is measured by the rank of documents which remain the same if the noise only affects to the relevance scores but not to the rank of documents. However, at the last noise step ($\eta = 50$), the scores drop under 50% of original scores similar to the results on STS datasets. On the other hand, same as the STS result, our HE method maintains plaintext performance with little degradation (at most 0.0001993 point on ArguAna). More precisely, the average absolute deviation between the dot-products in plaintext and those in ciphertext lies from 3.67×10^{-8} in SciFact (lowest) to 3.94×10^{-8} in NFCorpus (highest).

5 Conclusions

In this paper, we proposed homomorphic encryption based text similarity inference with text embeddings. With our method, users can utilize text embedding based services without revealing the original text, which can be recovered through inversion attacks as we demonstrated in the experiment 4.3. Extensive experiments 4.4 on two text similarity tasks proved that our approach does not harm the performance of models. In contrast, the d_χ-privacy baselines fail to achieve protection from inversion attacks without performance degradation. We hope that this work lays

the groundwork for the secure usage of text embeddings in privacy-sensitive industries like finance and more future works on the practical usage of our HE approach by resolving the current limitations.

Limitations

Our HE-based methods report that a vector-vector multiplication in STS takes roughly 30 to 40 ms per text on average. For text retrieval, a vector-matrix multiplication per query takes approximately 0.6 to 0.7 seconds against 1,000 documents in our benchmark datasets on average. The computation time increases linearly depending on the number of documents. Since operations in an encrypted state are computationally expensive, efficiency need to improve in computing time to provide document search services over large amounts of corpora for a practical use.

For efficient search with text embedding similarities, modern applications equip with approximate search frameworks like faiss[6]. Such method becomes more crucial when handling open-domain search corpus like Wikipedia (larger than 5 million of documents). Since the HE implementation in this paper focuses on relatively simple similarity functions like cosine similarity, it is non-trivial to be directly incorporated with existing frameworks and algorithms that utilize complex data structures and operations like hashing and graph-based search. Therefore, one of our future works will be the research on the implementation of the HE based efficient search methods.

Acknowledgements

We thank our anonymous reviewers for their constructive comments. Lee was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2022-0-01047, Development of statistical analysis algorithm and module using homomorphic encryption based on real number operation].

[6] https://github.com/facebookresearch/faiss

References

Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, 51(4).

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2011. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Paper 2011/277. https://eprint.iacr.org/2011/277.

Sergiu Carpov and Renaud Sirdey. 2015. A compression method for homomorphic ciphertexts. *IACR Cryptol. ePrint Arch.*, 2015:1199.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2019. Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. THE-X: Privacy-preserving transformer inference with homomorphic encryption. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3510–3520, Dublin, Ireland. Association for Computational Linguistics.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2018a. Bootstrapping for approximate homomorphic encryption. In *Advances in Cryptology – EUROCRYPT 2018*, pages 360–384, Cham. Springer International Publishing.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2018b. A full RNS variant of approximate homomorphic encryption. In *International Conference on Selected Areas in Cryptography*, pages 347–368. Springer.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2019. A full rns variant of approximate homomorphic encryption. In *Selected Areas in Cryptography – SAC 2018*, pages 347–368, Cham. Springer International Publishing.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, Cham. Springer International Publishing.

Jung Hee Cheon, Yongha Son, and Donggeon Yhee. 2022. Practical FHE parameters against lattice attacks. *Journal of the Korean Mathematical Society*, 59(1):35–51.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Junfeng Fan and Frederik Vercauteren. 2012. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Paper 2012/144. `https://eprint.iacr.org/2012/144`.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 178–186. ACM.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. volume 9, pages 169–178.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, pages 2041–2044. ACM.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Garam Lee, Minsoo Kim, Jai Hyun Park, Seung-won Hwang, and Jung Hee Cheon. 2022. Privacy-preserving text classification on BERT embeddings with homomorphic encryption. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3169–3175, Seattle, United States. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Andreas Nautsch, Sergey Isadskiy, Jascha Kolberg, Marta Gomez-Barrero, and Christoph Busch. 2018. Homomorphic encryption for speaker recognition: Protection of biometric templates and vendor model parameters. pages 16–23.

Samanvaya Panda. 2021. *Principal Component Analysis Using CKKS Homomorphic Scheme*, pages 52–70.

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Natural language understanding with privacy-preserving bert. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management*, CIKM '21, page 1488–1497, New York, NY, USA. Association for Computing Machinery.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Paul M. Schwartz and Daniel J. Solove. 2011. The PII Problem: Privacy and a New Concept of Personally Identifiable Information. *NYUL Rev.*, 86:1814–1894.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 377–390, New York, NY, USA. Association for Computing Machinery.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Spencer Wheatley, Thomas Maillart, and Didier Sornette. 2016. The extreme risk of personal data breaches and the erosion of privacy. *The European Physical Journal B*, 89(1):1–12.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ping Xiong, Lin Liang, Yunli Zhu, and Tianqing Zhu. 2022. Pritxt: A privacy risk assessment method for text data based on semantic correlation learning. *Concurrency and Computation: Practice and Experience*, 34(5):e6680.

Xiaojie Yu, Xiaojun Chen, and Jinqiao Shi. 2017. Vector based privacy-preserving document similarity with lsa. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pages 1383–1387.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning

books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Experimental Settings for Text Similarity Tasks

Hyperparameters for text embedding models are shown in Table 7. We only report the parameters necessary for inference because we do not fine-tune the models at all. The models used for two text similarity tasks have differences on 1) their pooling strategies which decides the way to aggregate transformer hidden states to single text embedding, 2) the similarity functions to calculate the relevancy between text embeddings.

Hyperparams	SimCSE	DistilBERT
Pooling strategy	[CLS]	mean
Max sequence length	512	512
Embedding size	768	768
Similarity	cosine	dot product

Table 7: Hyperparams for text embedding model test.

B Experimental Settings for Text Embedding Inversion

Hyperparameters for inversion model are shown in Table 8. We sample 100k sentences from BookCorpus as train data. We choose the best threshold parameter based on the results with thresholds from 0.6 to 1.0 with an interval 0.05. As a result, 0.90 and 0.95 thresholds are selected for SentEval and BEIR benchmark, respectively. To build the vocabulary for inversion model predictions, we tokenize given texts with spacy[7] and postprocess them by removing stopwords and normalizing words with lemmatization[8].

Hyperparams	Inversion Model
Learning rate	0.001
Max epoch	100
Batch size	64
Hidden size	768
Threshold range	[0.6, 1.0]

Table 8: Hyperparams for inversion model training and test.

[7] https://spacy.io/
[8] https://www.nltk.org/index.html

C Text Embedding Inversion Results

Table 9 shows additional text embedding inversion results from NFCorpus, SCIDOCS, SciFact, and SICK-R.

Original text from **NFCorpus:**	
Do Cholesterol Statin Drugs Cause Breast Cancer?	
Plaintext	cholesterol, cancer, caus, statin, drug, breast
$\eta = 175$	cholesterol, caus, statin, cancer, breast, exact
$\eta = 150$	cholesterol, cancer, statin, breast, drug, caus
$\eta = 125$	cholesterol, cancer, statin, breast, caus
$\eta = 100$	cholesterol, breast, caus, cancer, statin
$\eta = 75$	cholesterol, cancer, statin, breast, drug, induc
$\eta = 50$	cholesterol, cancer, statin, breast, soar
Original text from **SCIDOCS:**	
Digital image forensics: a booklet for beginners	
Plaintext	beginn, digit, imag, begin, twelv
$\eta = 175$	digit, beginn, photograph, slowli, fascin, pictur
$\eta = 150$	digit, beginn, photograph, fool, examin, pictur
$\eta = 125$	beginn, digit, photograph, photo, imag, dive
$\eta = 100$	photograph, pictur, imag, digit, beginn, studi
$\eta = 75$	photograph, digit, absorb, prod, fascin
$\eta = 50$	drawer, manual, photo, examin, lectur, memor
Original text from **SciFact:**	
0-dimensional biomaterials show inductive properties.	
Plaintext	biomateri, dimension, induct, properti
$\eta = 175$	dimension, biomateri, induct, properti, note
$\eta = 150$	induct, dimension, biomateri, feminin, tight, close
$\eta = 125$	biomateri, dimension, properti
$\eta = 100$	biomateri, dimension, induct, element, feminin, announc
$\eta = 75$	induct, scrub, tentat, dealt, project, show
$\eta = 50$	dimension, agon, daze, biomateri, induct, darren
Original text from **SICK-R:**	
A black dog on a leash is walking in the water	
Plaintext	dog, black, collar
$\eta = 175$	black, bella
$\eta = 150$	black, bella
$\eta = 125$	black, bella
$\eta = 100$	black, bella
$\eta = 75$	mall, daypack
$\eta = 50$	-

Table 9: Result of Text Embedding Inversion

D Our HE parameter selection

For STS, we selected CKKS parameter whose dimension $N = 2^{16}$ and its modulus q is 2^{1555}. For text retrieval, since dot products only require additions and multiplications, we select a ciphertext parameter preset for Somewhat Homomorphic Encryption (Gentry, 2009) for efficiency in computation. We choose a parameter set where dimension N is 2^{13} so each ciphertext block consists of $2^{13-1} = 4,096$ slots and its modulus $q \approx 2^{217}$ guarantees a 128-bit security level under SparseLWE-estimator.