ALGEN: Few-shot Inversion Attacks on Textual Embeddings via Cross-Model Alignment and Generation

Yiyi Chen[1], **Qiongkai Xu**[2]*, **Johannes Bjerva**[1]
[1]Department of Computer Science, Aalborg University, Copenhagen, Denmark
[2]School of Computing, FSE, Macquarie University, Sydney, Australia
yiyic@cs.aau.dk, qiongkai.xu@mq.edu.au, jbjerva@cs.aau.dk

Abstract

With the growing popularity of Large Language Models (LLMs) and vector databases, private textual data is increasingly processed and stored as numerical embeddings. However, recent studies have proven that such embeddings are vulnerable to inversion attacks, where original text is reconstructed to reveal sensitive information. Previous research has largely assumed access to millions of sentences to train attack models, e.g., through data leakage or nearly unrestricted API access. With our method, *a single data point* is sufficient for a partially successful inversion attack. With as little as 1k data samples, performance reaches an optimum across a range of black-box encoders, without training on leaked data. We present a Few-shot Textual Embedding Inversion Attack using Cross-Model **AL**ignment and **GEN**eration (**ALGEN**), by aligning victim embeddings to the attack space and using a generative model to reconstruct text. We find that **ALGEN** attacks can be effectively transferred across domains and languages, revealing key information. We further examine a variety of defense mechanisms against **ALGEN**, and find that none are effective, highlighting the vulnerabilities posed by inversion attacks. By significantly lowering the cost of inversion and proving that embedding spaces can be aligned through one-step optimization, we establish a new textual embedding inversion paradigm with broader applications for embedding alignment in NLP.[1]

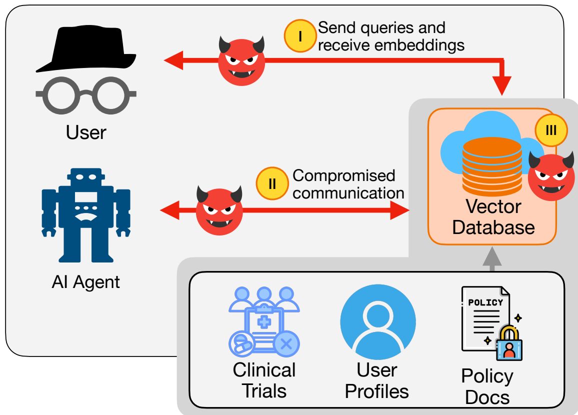


Figure 1: An illustration of inversion attacks on textual embeddings stored in a vector DB, in scenarios where (I) a user exploits API access to extract excessive embeddings to train attack model; (II) a generative AI agent's interaction channel with the DB is compromised; (III) the DB is misconfigured by an insider to expose private embeddings.

1 Introduction

Large Language Models (LLMs) such as OpenAI's GPT series (Radford, 2018; Radford et al., 2019; Brown et al., 2020; OpenAI et al., 2024) and Claude from Anthropic,[2] have become essential across a wide range of applications, extending far beyond natural language processing (NLP). These models are deeply integrated into people's daily lives and business operations, e.g., powering search engines, virtual assistants, and content generation. A critical component enabling the efficiency of these applications is vector databases (DB), which allow for fast and scalable retrieval and processing of high-dimensional vector representations. Companies such as Pinecone and Weaviate, provide vector DB services and build AI services on top of them.[3] In a recent Google whitepaper on generative AI agents (Wiesinger et al., 2025), vector DBs are considered one of the essential components enabling such agents through external sources. Retrieval-augmented generation (RAG) is another common use case in leveraging vector DBs to generate more diverse and factually grounded responses (Lewis et al., 2020).

While applications such as these benefit from

*Corresponding author.
[1]We open-source our code https://github.com/siebeniris/ALGEN.
[2]https://www.anthropic.com/claude
[3]https://www.pinecone.io, https://weaviate.io

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24330–24348
July 27 - August 1, 2025

vector DBs, the potential security and privacy risks permeate the process. Fig. 1 illustrates three separate threat scenarios where a vector DB can be exploited to expose private and sensitive information: (I) a malicious user exploits the model API to extract embeddings to train an attack model; (II) when an AI agent interacts with the DB, a malicious attacker can compromise the communication channel to intercept sensitive data; (III) a misconfigured vector DB may expose private data, either through access vulnerabilities or insider threats. The attacker can train an attack model (e.g. an embedding-to-text generator) to reconstruct text from intercepted embeddings, which might contain sensitive, private, or proprietary information. This so-called *embedding inversion attack* poses significant risks and potential harm.

Previous work has demonstrated the feasibility and detrimental effects of inversion attacks (Song and Raghunathan, 2020; Li et al., 2023). However, either a massive amount of intercepted (victim) embeddings and their texts are required for training an attack model (Morris et al., 2023), or the attack is conducted under white-box settings, where the model parameters and architecture are known to the attacker (Song and Raghunathan, 2020). Moreover, inversion attacks have been demonstrated to threaten multiple languages, especially lower-resource ones (Chen et al., 2024a,b).

We propose a Few-shot Textual Embedding Inversion Attack using Cross-Model **AL**ignment and **GEN**eration (**ALGEN**), to first align victim embeddings to the attack embedding space, and then reconstruct text from the aligned embeddings using the generative attack model. In contrast to previous work, we investigate inversion attacks using a small handful of samples – e.g., a Rouge-L score of 10 can be reached by leveraging *a single leaked data point*. Our work makes the following main contributions:

- We propose and verify the effectiveness of a novel few-shot inversion attack, which drastically reduces the cost and complexity of such attacks, making them plausible real-world threats.
- We demonstrate the transferability of the inversion attack across various languages, models and domains.
- We examine several established defense mechanisms, none of which are successful mitigation strategies for this attack, highlighting the new security and privacy vulnerabilities of embeddings in vector databases.

2 Related Work

2.1 Textual Embedding Inversion Attacks

Textual embedding inversion attack aims to learn the inversion function that reconstructs the original textual inputs given their embeddings. Song and Raghunathan (2020) demonstrates that it is possible to recover over half of the input words from a text embedding without preserving their order. Li et al. (2023) starts to treat the inversion attacks as a generation task, generating coherent and contextually similar sentences compared to the original text. Morris et al. (2023) adopts an iterative approach to train the attack model by parameterizing attack and hypothesis embeddings based on decoded text from the previous step, which results in exact matches between original and reconstructed text in certain settings. Huang et al. (2024) implements adversarial training to align victim embeddings to attack embeddings, making them not differentiable. Chen et al. (2024a,b) expand inversion attacks beyond English embeddings to multilingual spaces, leveraging linguistic typology to investigate inversion attack performance, finding that certain languages are particularly vulnerable.

However, all existing works in embedding inversion attacks require an enormous amount of data leakage to train the generative attack models, such as 100k samples for Li et al. (2023), 1-5 million for Morris et al. (2023); Chen et al. (2024a,b) and 8k for Huang et al. (2024). In comparison, our proposed approach **ALGEN** does not require training an attack model on leaked/private embeddings, and the inversion attack succeeds with few leaked data, we additionally experiment on multiple languages.

2.2 Embedding Alignment

Embedding alignment has continuously progressed in NLP with the development of embedding representations and LLMs. In the early stages, a common approach involved independently training monolingual word vectors (Mikolov et al., 2013b) and then learning a mapping between source and target language embeddings using a bilingual dictionary (Mikolov et al., 2013a; Smith et al., 2017; Artetxe et al., 2017). When this mapping is restricted to an orthogonal linear transformation, the optimal word pair alignment can be computed in closed form (Artetxe et al., 2016; Schönemann, 1966). In contrast, Lample et al. (2018) introduce an unsupervised method for aligning word embedding spaces, incorporating cross-domain similarity adaptation to address the hubness problem.

With the advancement of contextualized embeddings since the emergence of LLMs such as BERT (Devlin, 2018), the focus shifted to the alignment of contextual word representations (Schuster et al., 2019; Aldarmaki and Diab, 2019; Wang et al., 2019b; Alqahtani et al., 2021; Cao et al., 2020; Jalili Sabet et al., 2020). Moreover, sentence embedding alignment has been operated in lifelong relation extraction with a linear transformation (Wang et al., 2019a), aligning encoders in different languages to evaluate crosslingual transfer (Conneau et al., 2018), aligning unsupervised multilingual sentence embeddings across languages to build parallel corpus (Kvapilíková et al., 2020), building parallel data for machine translation (Krahn et al., 2023), and further to improve information retrieval (Bhattarai et al., 2025; Yadav and McMillan, 2024).

In comparison, **ALGEN** aligns sentence embeddings from different models to conduct embedding inversion attacks, but it can also be applied in embedding alignment in general.

2.3 Mitigating Embedding Inversion Attacks

Most research on textual embedding inversion focuses on attacks (Li et al., 2023; Huang et al., 2024; Chen et al., 2024a). While Song and Raghunathan (2020) adopt an adversarial training approach to mitigate the risks of inversion attacks, this method is ineffective for defending textual embeddings in black-box settings. To defend against inversion attacks while maintaining embedding utility in downstream tasks, Morris et al. (2023) propose inserting Gaussian noise as a defense mechanism. Expanding inversion attacks into multilingual space using the same method, Chen et al. (2024b) find that Gaussian noise effectively protects monolingual embeddings but is less effective for multilingual ones.

Differential privacy (DP) limits the impact of individual element (Dwork et al., 2014), and has been shown to preserve the privacy of the extracted representation from text, when applied during model training (Lyu et al., 2020). To ensure sequence-level metric-based local DP, which can be employed during inference, a sentence embedding

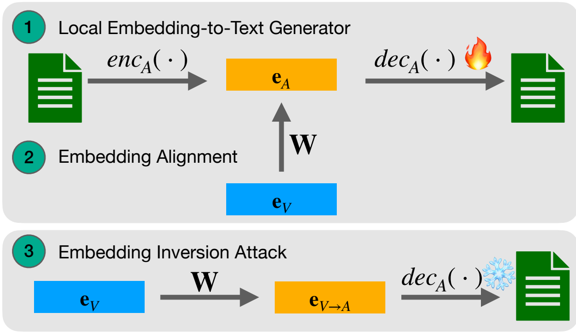


Figure 2: Three steps for Few-shot Inversion Attack, (1) Train a Local Embedding-to-Text Generation Model; (2) Transform victim embeddings $\boldsymbol{e}_V$ to the attack embeddings space A with matrix $\boldsymbol{W}$; and (3) Textual embedding inversion attack.

sanitization pipeline has been developed, maintaining non-private task accuracy and effectively thwarting privacy threats of membership inference attacks (Du et al., 2023). Pertinent to vector DBs, Watermarking EaaS with Linear Transformation (WET) introduces a method that applies linear transformations to embeddings to implant watermarks to counter paraphrasing vulnerabilities (Shetty et al., 2024).

In this work, we examine these defenses against **ALGEN**, and discuss the potentials and challenges of defending embeddings from inversion attacks.

3 Methodology

We explore a situation in which a malicious attacker gains access to a *limited set of embeddings*, and attempts to reconstruct private and sensitive text data. We propose **ALGEN** to circumvent the disadvantage of scarce data and leverage a pretrained encoder-decoder to align the victim embeddings to the attack space and reconstruct the texts. We note the victim and attack embedding spaces as V and A, respectively. As illustrated in Fig. 2, the framework consists of three steps: (1) we train an embedding-to-sequence generation model by fine-tuning a pre-trained decoder $dec_A(\cdot)$; (2) we align the embeddings from a black-box victim encoder $\boldsymbol{e}_V$ to attack embeddings in attacker's model space $\boldsymbol{e}_A$; (3) the attacker leverages the capability of the generation model $dec_A(\cdot)$ with the embedding alignment model $\boldsymbol{W}$ to reconstruct the original text from $\boldsymbol{e}_V$ to $\boldsymbol{e}_{V\to A}$ and finally to text.

3.1 Local Embedding-to-Text Generator

To train the local embedding-to-text generator, we use a publicly available text corpus, noted as D_L. Given a sentence $x \in D_L$, and attack encoder $enc_A(\cdot)$, the token embeddings are obtained $\boldsymbol{H} = enc_A(x) \in \mathbb{R}^{s\times n}$ where s is the sequence length and n the embedding dimension. The sentence embeddings are computed through mean pooling the last hidden embeddings $\boldsymbol{e}_A = \sum_{j=1}^{s} \boldsymbol{m}_j \boldsymbol{H}_j / \sum_{j=1}^{s} \boldsymbol{m}_j \in \mathbb{R}^n$, where $\boldsymbol{m} \in \{0,1\}^s$ is the attention mask for the sequence x. Furthermore, L2 normalization is implemented on the sentence embeddings, i.e., $\boldsymbol{e}_A/\|\boldsymbol{e}_A\|$ as sentence embedding normalization has proven beneficial in avoiding overfitting and inducing faster convergence in fine-tuning (Aboagye et al., 2022). The pre-trained decoder $dec_A(\cdot)$, parameterized by θ, processes the embeddings to produce an output sequence $\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_s)$, where $\hat{x}_i$ are tokens in the predicted output sequence. We train the embedding-to-text generator by fine-tuning $dec_A(\cdot)$, by minimizing the cross-entropy loss:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{s} \log \mathbb{P}(\hat{x}_i | \hat{x}_{<i}, \boldsymbol{e}_A; \theta), \quad (1)$$

where $\mathbb{P}(\hat{x}_i | \hat{x}_{<i}, \boldsymbol{e}_A; \theta)$ is the probability of predicting token $\hat{x}_i$ given the previous tokens $\hat{x}_{<i}$ and the input sentence embeddings $\boldsymbol{e}_A$.

Notably, attackers can easily retrieve a large corpus D_L to train this generator model, as it operates independently of any victim models. The challenge of aligning victim semantics to target embedding space is addressed in the next subsection.

3.2 Embedding Space Alignment

Suppose there is a leaked data pair $(\boldsymbol{X}, \boldsymbol{E}_V)$, given that $\boldsymbol{X} \subseteq D_V$ is the victim dataset with $b \in \mathbb{N}$ samples, and embedding matrix $\boldsymbol{E}_V = enc_V(\mathbf{X})$, where enc_V is the black-box victim encoder. To align the victim embeddings to the attack space A, we obtain the embedding matrix $\boldsymbol{E}_A = enc_A(\mathbf{X})$ given the leaked text dataset, and seek a solution to solve the system

$$\boldsymbol{E}_V \boldsymbol{W} \approx \boldsymbol{E}_A, \quad (2)$$

and the best possible $\boldsymbol{W} \in \mathbb{R}^{m\times n}$, given $\boldsymbol{E}_V \in \mathbb{R}^{b\times m}$ and $\boldsymbol{E}_A \in \mathbb{R}^{b\times n}$, where n and m are the regarding embedding dimensions of victim and attacker embeddings. While there is no exact solution to the system, our approach is to minimize their deviation, $e = \boldsymbol{E}_A - \boldsymbol{E}_V \boldsymbol{W}$. Taking the square of the error by each sample, the objective is to minimize the following:

$$\min_{\boldsymbol{W}} \sum_{i=1}^{b} \|\boldsymbol{e}_A^i - \boldsymbol{e}_V^i \cdot \boldsymbol{W}\|^2. \quad (3)$$

The solution to this least squares loss is:

$$\boldsymbol{W} = (\boldsymbol{E}_V^T \boldsymbol{E}_V)^{-1} \boldsymbol{E}_V^T \boldsymbol{E}_A, \quad (4)$$

where $(\boldsymbol{E}_V^T \boldsymbol{E}_V)^{-1} \boldsymbol{E}_V^T$ is the *Moore-Penrose Inverse* of $\boldsymbol{E}_V$ (see the detailed derivation in Appendix A).

The aligned embedding from V to A is thus:

$$\boldsymbol{E}_{V\to A} = \boldsymbol{E}_V \boldsymbol{W}, \quad (5)$$

where $\boldsymbol{E}_{V\to A} \in \mathbb{R}^{b\times n}$. Implementing this alignment does not require any training; it is a one-step linear scaling. Moreover, aligning using D_V with a batch size b varying from 1 to 1,000, we observe that even with as few as 30 samples the Rouge-L score exceeds 20, and a reasonably successful attack can be initiated with only a single data point. The density distribution of the alignment transformation matrix $\boldsymbol{W}$'s weights of encoders remains consistent across different datasets (see Fig 5).

3.3 Textual Embedding Inversion Attack

Given the attack model, i.e. the local embedding-to-text generator $dec_A(\cdot)$ and the $\boldsymbol{e}_V$ to $\boldsymbol{e}_A$ alignment model, and a body of eavesdropped embeddings $\boldsymbol{E}_V$, we launch the inversion attack:

$$\hat{\boldsymbol{X}} = dec_A(\boldsymbol{E}_V \boldsymbol{W}). \quad (6)$$

4 Experimental Setup

4.1 LLMs

We use pretrained FLANT5 as the backbone to launch our attack modules, encoder $enc_A(\cdot)$ and decoder $dec_A(\cdot)$. For victim models, a variety of encoders are experimented on, including T5, GTR, MT5, MBERT and OpenAI text embedders TEXT-EMBEDDING-ADA-002 (ADA-2) and TEXT-EMBEDDING-3-LARGE (3-LARGE) (see the details of LLMs in Tabel 10).

4.2 Attack Experimental Setup

Datasets and Attack Model We train the embedding-to-text generator $dec_A(\cdot)$ by fine-tuning FLANT5-decoder, using the MultiHPLT English dataset (de Gibert et al., 2024) to explore few-shot inversion attacks. For multilingual inversion attacks, we utilize English, German, French, and Spanish datasets from mMarco (Bonifacio et al., 2021). In dataset, we split 150k samples (D_L) to train $dec_A(\cdot)$; and up to 1k samples (D_V) to derive the alignment metric $\boldsymbol{W}$ by aligning $\boldsymbol{e}_V$ to $\boldsymbol{e}_A$, as alignment samples; and 200 for evaluation.

Dataset	Clasisification	#Class	#Train	#Dev	#Test
SNLI	NLI	3	540,000	200	200
SST2	Sentiment	2	59,560	200	200
S140	Sentiment	2	1,599,798	200	200

Table 1: Statistics of Utility Task Datasets

Embedding-to-Text Generator Training We conduct a series of experiments varying the training set size (from 10k to 1M samples), learning rate and weight decay, then select the generator configuration that achieves the best Rouge-L score on the evaluation set. Eventually, to strike a balance of performance and data usage, we train $dec_A(\cdot)$, an embedding-to-sequence decoder, with a learning rate of $1e-4$ and weight decay $1e-4$ on AdamW optimizer (Loshchilov et al., 2017), batch size 128, and 150k data samples performs the best. We use Cross-entropy Loss for training the generator.

Evaluation Metrics **Rouge-L** (Lin, 2004) is used to measure the accuracy and overlap between ground truth text x and reconstructed text $\hat{x}$ based on n-grams. In addition, we report **Rouge1**, **BLEU1** and **BLEU2**. **Cosine Similarity (COS)** between the aligned victim embeddings $\boldsymbol{e}_{V\to A}$ and the attack embeddings $\boldsymbol{e}_A$ is calculated to evaluate the semantic similarity in the latent embedding space.

4.3 Defense Experimental Setup

We aim to evaluate how embeddings with defense mechanisms perform in downstream tasks.

Datasets We use SST2 (Socher et al., 2013), sentiment140 (S140) (Go et al., 2009) and SNLI (Bowman et al., 2015) in our experiments, which are curated to ensure a balanced distribution of labels. Table 1 shows the statistics of datasets.

Utility of Embeddings Using the embeddings from the victim encoders, we train multi-layer perceptron classifiers on the datasets and evaluate the accuracy (ACC) and F_1-score (F1) performance. We train each classifier 6 epochs, and select the best model with ACC on the dev dataset to evaluate the test dataset. There would be minimal difference in the utility performance between the protected and original embeddings, if a defense is successful.

	Victim	BLEU1	BLEU2	Rouge-L	Rouge1	COS
	$enc_A(\cdot)$	62.27	40.68	54.16	62.07	-
Vec2Text	T5 (Base)	21.47	9.07	**17.38**	19.52	0.4663
	Corrector	18.35	7.60	15.81	17.76	<u>0.4835</u>
	GTR (Base)	6.70	2.31	4.70	4.82	0.1911
	Corrector	13.42	2.79	**10.26**	12.31	<u>0.2725</u>
	MT5 (Base)	22.27	9.86	**17.21**	19.28	<u>0.7118</u>
	Corrector	18.73	7.79	15.98	17.82	0.6891
	MBERT (Base)	21.56	9.09	**16.97**	18.81	<u>0.5335</u>
	Corrector	18.45	7.48	15.99	18.10	0.5531
ALGEN	RANDOM	11.63	0.6	7.09	8.36	-0.0440
	T5	52.98	33.86	**45.75**	51.56	<u>0.9464</u>
	GTR	42.59	26.17	38.27	42.32	0.8879
	MT5	49.61	31	**43.35**	48.47	<u>0.9370</u>
	MBERT	47.06	28.66	39.9	45.04	0.9217
	OPENAI (ADA-2)	46.7	28.67	**41.45**	47.01	<u>0.9312</u>
	OPENAI (3-LARGE)	46.28	28.74	41.31	46.28	0.9066

Table 2: Inversion Attack Performances by victim models with 1,000 leaked data samples. The best Rouge-L scores are **bolded**, and the highest cosine similarities are <u>underlined</u>.

5 Few-shot Inversion Attacks

Each subsection aims to answer one Research Question (RQ).

5.1 How few Leaked Data do Attackers Need?

With only a single leaked data sample, our attack model manages to invert the victim embeddings, achieving a Rouge-L score of 10 across the encoders, as shown in Fig. 3. We use randomly generated embeddings as a baseline to verify that the aligned embeddings from **ALGEN** capture meaningful information. As shown in Table 2, all victim embeddings substantially outperform the RANDOM across metrics, validating our approach. Furthermore, when the number of leaked data samples increases until 1k, the inversion performance increases sharply, reaching 45.75 in Rouge-L and 0.9464 for cosine similarity for T5 embeddings. Notably, while GTR and T5 share the same tokenizer with the attack model, their inversion performances are not superior to others. Moreover, the inversion performance on proprietary OPENAI embeddings also reaches comparable performance, more than 41 in Rouge-L and 0.9 in cosine similarities for both ADA-2 and 3-LARGE, highlighting the risks posed by inversion attacks. As shown in the qualitative analysis in Table 11, some sentences can be inverted with almost an exact match. Furthermore, we conduct an ablation study with the size of alignment data samples and find that 1k

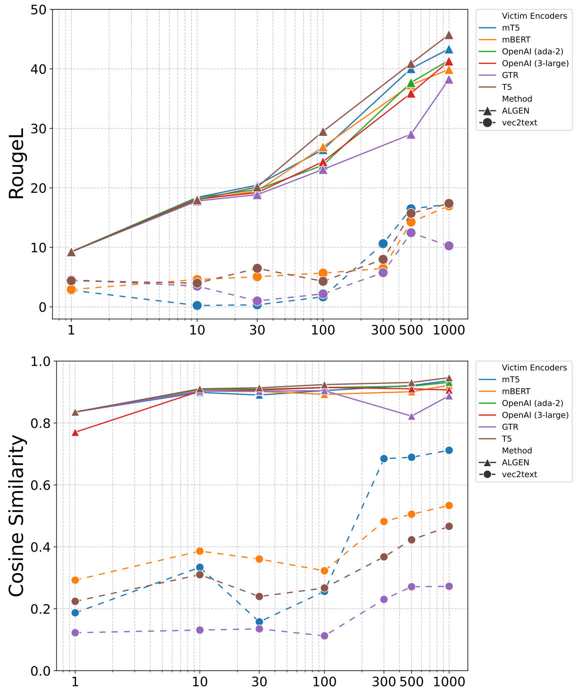


Figure 3: Inversion Performance in Rouge-L (Top) and Cosine Similarities (Bottom) by Victim Models and Alignment samples. Dashed lines are results of Vec2Text and solid lines are results of **ALGEN**.

Victim	λ	1	10	30	100	500	1000
MT5	0	9.27	18.40	20.46	26.43	40.04	43.35
	0.001	9.27	18.44	20.49	26.09	39.99	43.00
	0.01	9.27	18.45	20.60	25.95	39.77	42.99
	0.1	9.27	18.82	19.96	27.48	40.25	43.50
	1	9.27	13.99	18.82	20.86	35.30	38.55
	10	9.27	10.15	11.10	12.38	21.38	25.84
ADA-2	0	9.27	18.35	19.72	23.82	37.70	41.45
	0.001	9.27	18.39	19.81	24.45	37.72	41.23
	0.01	9.27	18.37	19.64	24.18	37.86	41.55
	0.1	9.27	18.27	19.48	24.80	38.77	41.75
	1	9.27	11.66	16.20	19.23	32.99	38.22
	10	9.27	8.86	10.50	10.73	20.10	21.55

Table 3: Inversion Performance in Rouge-L of the Victim models across regularization parameter λ by the Alignment samples.

alignment samples strike a balance between data size and performances (see Fig. 6 in Appendix. C).

We compare our method with Vec2Text (Morris et al., 2023), which trains two-step models (i.e., Base and Corrector) with iterative access to the victim encoder, and it requires training on embeddings from each encoder to invert the regarding embeddings. In comparison, our method only requires training one local attack model, and training does not involve specific victim encoders. Up to 1k samples, Vec2Text performance is much inferior compared to our method both in Rouge-L and cosine similarities, lower than 20 and 0.72, respectively, as detailed in Fig. 3 and Table 2.

We generally observe small alignment errors, with the cosine similarities between victim embeddings and attack embeddings consistently higher than 0.8 and near 1.0 when the number of alignment samples increases. The bottleneck of the performance of **ALGEN** is likely to lie in the decoding, as the trained attack decoder can only reach 54.16 in Rouge-L to invert the attack embeddings, as shown in Table 2, which is considered to be the upper bound of inversion attack performance.

5.2 Does Regularization further Increase Few-shot Inversion Vulnerabilities?

To experiment whether adding regularization in our solution $\boldsymbol{W}$ further improve the inversion attack performance, we add an L2 regularization term to Eq. 4, which give us

$$\boldsymbol{W} = (\boldsymbol{E}_V^T \boldsymbol{E}_V + \lambda \boldsymbol{I}_m)^{-1} \boldsymbol{E}_V^T \boldsymbol{E}_A, \qquad (7)$$

where m is the dimension of $\boldsymbol{E}_V$. We experimented with $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$, across a range of alignment samples $[1, 10, 30, 100, 500, 1000]$, and $\lambda = 0$ refers to no regularization.

As shown in Table 3, the best regularization results in Rouge-L provide only marginal improvements, which indicates the robustness and generalization of **ALGEN**.

5.3 Are Other Languages (More) Vulnerable?

Building on previous work on multilingual embedding inversion, we also investigate the impact of **ALGEN** on languages other than English. To achieve this, we trained local attack models in English, French, German, and Spanish. We further conduct crosslingual embedding inversion attacks, for example, applying the English-trained attack model to invert French textual embeddings. Table 5 summarizes the results of multilingual and crosslingual embedding inversion attacks. The first row shows the results of monolingual inversion on the attack embeddings, which serve as an upper bound of the inversion performances on this dataset. As expected, the monolingual inversion attacks perform better than the crosslingual ones.

Consistent with previous findings (Chen et al., 2024b), crosslingual inversion often reconstructs

Input	This business uses tools provided by **TripAdvisor [ORG]**(or one of its official Review Collection Partners) to encourage and collect guest reviews, including this one.
Reconstructed	This business uses tools provided by **TripAdvisor [ORG]** (or one of its official Review Collection Partners) to encourage and collect guest reviews, including this one
Input	Book your flights now from Hermosillo (**Mexico [GPE]**) to the most important cities in the world. The box below contains flights from Her.
Reconstructed	Book your flights now from Las Vegas (**Mexico [GPE]**) to the most important cities in the world. The box below contains flights from Las Vegas to

Table 4: Qualitative Analysis of In-Domain Inversion Results from OPENAI (ADA-2) embeddings with 1k alignment data samples and the attack model trained on MultiPHLT English dataset. The matched entities with their entity types are colored and **bolded** in **Input** and **Reconstructed**. The mismatched reconstructed texts are in grey colored box. [GPE]: Countries/cities/states; [ORG]:Organization.

	Attack Lang.	Victim Languages English	French	German	Spanish
$enc_A(\cdot)$		54.47	52.78	24.77	53.98
T5	**English**	29.29	12.57 (+2.58)	5.54 (+0.7)	13.31 (+0.33)
	French	8.67 (+3.37)	31.01	2.4 (+1.25)	13.72 (+0.84)
	German	15.72 (+0.09)	15.56 (+0.06)	13.04	16.63 (+0.09)
	Spanish	6.56 (+4.49)	11.33 (+1.05)	2.18 (+0.95)	31.83
GTR	**English**	19.22	10.45 (+1.34)	4.27 (+0.57)	10.17 (+1.09)
	French	4.78 (+2.85)	23.98	1.95 (+1.14)	12.16 (+0.42)
	German	11.13 (+0.48)	13.37 (-0.15)	8.48	14.32 (-0.09)
	Spanish	4.4 (+3.74)	10.5 (+1.98)	1.9 (+0.95)	20.88
MT5	**English**	25.48	12.29 (+1.64)	5.31 (+0.03)	12.61 (+0.87)
	French	8.66 (+2.79)	24.6	2.27 (+1.43)	13.1 (+0.48)
	German	15.54 (+0.06)	15.2 (-0.1)	10.09	15.59 (+0.02)
	Spanish	6.84 (+3.98)	11.62 (+1.35)	1.88 (+1.31)	24.05
MBERT	**English**	21.3	11.72 (+1.42)	4.46 (+0.3)	11.76 (+0.82)
	French	5.91 (+3.61)	22.79	1.87 (+1.71)	12.05 (-0.19)
	German	14.29 (+0.12)	14.16 (-0.05)	9.17	15.6 (+0.05)
	Spanish	5.17 (+4.14)	11.07 (+0.09)	1.75 (+0.57)	21.74
OPENAI (ADA-2)	**English**	24.18	11.62 (+1.32)	5 (+0.17)	11.36 (+0.95)
	French	6.97 (+3.73)	22.73	1.59 (+1.52)	12.73 (+0.07)
	German	14.74 (+0.11)	13.7 (+0.07)	9.63	14.33 (+0.05)
	Spanish	7.08 (+3.57)	9.82 (+1.39)	1.63 (+1.05)	21.25

Table 5: Crosslingual Embedding Inversion Performance in Rouge-L with $|D_V|$ =1k. The results in the brackets are the performance gain after translation.

text in a language other than the intended, usually English (the dominant language in most multilingual LLMs) or trained languages in the attack model, hindering the performance evaluation with string-match metrics. For example, a French text "*un composé organique qui ne contient que du carbone*", is reconstructed into English "*a chemical compound composed of carbon*". Rouge-L evaluation is more accurate when the inverted text is translated back into the target language (e.g., "*un composé chimique composé de carbone*"). To ensure fairness, we translate the inverted texts into their target languages using deep-translator.[4] After translation, Rouge-L scores improve across most models, with the most notable gains observed in the French-to-English scenario, as shown in Table 5. From an attacker's perspective, splitting words in English rather than in the victim's language is advantageous. Attackers can be assumed to be proficient in English, while the victim's language might be unintelligible to them. This further exacerbates the vulnerabilities of non-English languages.

5.4 Is Risk Transferable across Domains?

To evaluate the cross-domain transferability of **ALGEN**, we attack the embeddings on the mMarco English dataset using an attack model trained on MultiHPLT English data. Although the cross-domain inversion performance in Rouge-L is about 25% lower than that of in-domain attacks on mMarco (see Table 5), the results remain alarmingly high - with Rouge-L near 20 and BLEU1 near 31 across victim encoders.

Model	BLEU1	BLEU2	Rouge-L	Rouge1	COS
T5	31.4	5.72	21.76	29.6	0.9278
GTR	22.04	2.26	15.27	20.45	0.8442
MT5	30.08	4.82	19.33	26.94	0.9188
MBERT	26.68	3.57	17.16	23.88	0.9033
OPENAI (ADA-2)	27.62	4.63	19.07	26.40	0.9089

Table 6: Cross-Domain Inversion Attack with $|D_V|$=1k.

5.5 Does Inversion Recover Key Information?

Model	Overall	Product	ORG	GPE	Date	Time	Cardinal	Ordinal
T5	23.86	50	32.41	21.62	14.55	16.67	28.57	40.00
MT5	20.2	53.33	28.04	17.5	11.11	0	10.53	25.00
GTR	21.17	54.55	32.13	10.67	11.11	22.22	13.79	25.00
MBERT	19.68	49.12	28.03	12.35	15.09	33.33	6.90	33.33
OPENAI	22.45	53.33	34.13	16.47	13.79	0	14.55	37.50

Table 7: Named Entity Recognition in F1 scores for overall and top 10 entities in specific.

To examine whether **ALGEN** attacks real key information, we apply Named Entity Recognition[5] on input and reconstructed test data from MultiHPLT English dataset to calculate the current ratio

[4] https://github.com/nidhaloff/deep-translator

[5] https://github.com/explosion/spacy-stanza

of Named Entities in the reconstructed texts. Table 7 shows the results in F1 scores for overall and individual entities. In the qualitative analysis, as shown in Table 4 and 11, the input and reconstructed texts in the inversion attacks on OpenAI (ada-2) embeddings are compared, with named entities highlighted. These attacks reveal sensitive details, such as organization, country, and numbers, highlighting the risks of privacy disclosure by embeddings.

6 Defending Textual Embeddings

To explore defenses against **ALGEN**, we evaluate defense mechanisms designed to protect textual embeddings from adversarial attacks.

6.1 Defense Methods

WET We implement WET on textual embeddings, to examine whether it makes embeddings robust against inversion attacks, since it is effective in defending paraphrasing attacks (Shetty et al., 2024). A transformation matrix $\boldsymbol{T}$ is generated to transform the $\boldsymbol{e}_V$ into $\boldsymbol{e}_{WET}$ with $\boldsymbol{T} \cdot \boldsymbol{e}_V / \|\boldsymbol{T} \cdot \boldsymbol{e}_V\|$, to ensure that i) the original elements are discarded and only the transformed ones are retained; and ii) $\boldsymbol{T}$ is full-rank and well-conditioned to allow for accurate recovery of the original embeddings (see details of generating $\boldsymbol{T}$ in Appendix B.1).

Shuffling We randomly shuffle the embeddings with $\boldsymbol{e}_{V,\pi(i)}$, where π is a random permutation function that reorders the indices i along the hidden dimension.

Gaussian Noise Insertion We add Gaussian noises to $\boldsymbol{e}_V$ with $(\boldsymbol{e}_V + \lambda \cdot \epsilon)/\|\boldsymbol{e}_V + \lambda \cdot \epsilon\|, \epsilon \sim \mathcal{N}(0, 1)$ (Morris et al., 2023; Chen et al., 2024b) with $\lambda \in [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1]$.

Differential Privacy Du et al. (2023) adopts Purkayastha Mechanism (PurMech) and Normalized Planer Laplace (LapMech) on sentence embeddings to ensure metric-LDP (see details in Appendix B.2). We adopt the parameters from Du et al. (2023) to experiment with defending textual embeddings from inversion attacks. The privacy budgets $\epsilon \in [1, 4, 8, 12]$ are selected.

6.2 Results

As shown in Table 8, 12 and 13, WET and Shuffling have minimal impact on both inversion attack performance and the utility performance across victim models and datasets. The randomly generated embeddings are also used as a baseline. With Gaussian noise insertion, the bigger the noise λ, both performance in inversion and utility decrease.

Victim	Defense	Rouge-L↓	COS↓	ACC↑	F1↑
Random	-	12.42	0.0052	40.5	40.33
T5	-	42.89	0.9595	63	62.92
	WET	39.4	0.9562	63	62.92
	Shuffling	35.33	0.9599	63	62.92
	0.001	43.3	0.9595	66.5	66.38
	0.005	42.88	0.9601	63	62.55
	0.01	40.26	0.9537	69	68.57
	0.05	22.82	0.8459	59	58.61
	0.1	19.32	0.7918	48	47.6
	0.5	14.29	0.3853	38	37.92
	1	12.97	0.1543	30.5	30.54
GTR	-	37.39	0.9284	60.5	60.53
	WET	35.58	0.9289	53.5	53.1
	Shuffling	31.01	0.9280	60.5	60.58
	0.001	36.52	0.9279	60	60.06
	0.005	36.48	0.9299	63	62.91
	0.01	34.41	0.9218	60.5	60.59
	0.05	22.31	0.8160	46.5	46.42
	0.1	19.39	0.7701	50.5	49.98
	0.5	14.31	0.2920	30.5	29.97
	1	13.48	0.1659	30.5	30.57
mT5	-	37.98	0.9518	61	60.95
	WET	34.69	0.9479	55.5	55.12
	Shuffling	31.83	0.9515	60.5	60.44
	0.001	37.97	0.9519	61.5	61.47
	0.005	37.84	0.9493	55.5	55.06
	0.01	34.5	0.9438	60	59.96
	0.05	21.58	0.8292	54	53.52
	0.1	17.65	0.7578	36.5	36.45
	0.5	14.8	0.4482	36	35.17
	1	13.31	0.1468	35.5	34.85
mBERT	-	35.47	0.9423	57	57.05
	WET	34.35	0.9428	53.5	53.5
	Shuffling	30.29	0.9408	52	51.93
	0.001	35.82	0.9422	51.5	51.37
	0.005	36.18	0.9409	51	50.82
	0.01	34.18	0.9349	57	57.01
	0.05	22.29	0.8265	50.5	50.48
	0.1	18.65	0.7741	43	43.03
	0.5	13.76	0.3969	35	33.96
	1	13.11	0.1824	32.5	32.62
OpenAI (ada-2)	-	38.15	0.9433	74.5	74.78
	WET	31.74	0.9405	64.5	64.51
	Shuffling	33.91	0.9440	72	72.31
	0.001	37.94	0.9437	74.5	74.51
	0.005	37.63	0.9445	69	69.09
	0.01	35.59	0.9409	67	66.85
	0.05	20.59	0.8426	49.5	49.39
	0.1	18.72	0.8231	44	42.76
	0.5	14.75	0.4483	37	36.87
	1	11.94	0.1097	30.5	30.49

Table 8: The Inversion and Utility Performance on Classification Tasks on SNLI dataset with WET, Shuffling, Guassian Noise Insertion. From a defender's perspective, ↑ means higher are better, ↓ means lower are better.

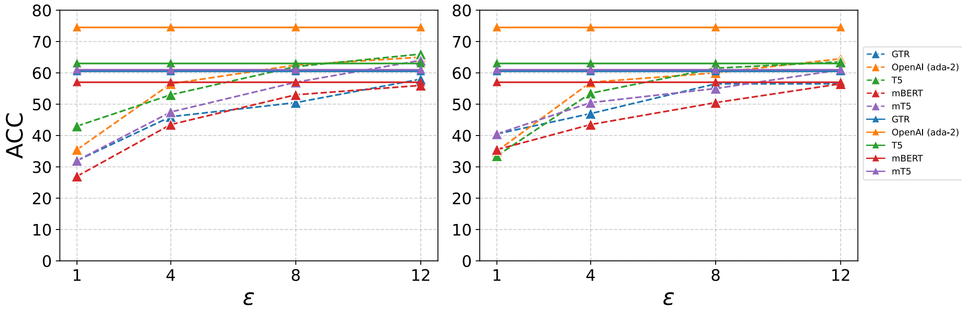


Figure 4: The Inversion and Utility Performance in Accuracy on Classification Tasks on SNLI dataset with local DP, across ϵ. The solid lines represent utility performance for non-private embeddings, while the dashed lines are for LDP-guaranteed embeddings.

Victim	ϵ	**Rouge-L**↓	**COS**↓	**Rouge-L**↓	**COS**↓
		LapMech		PurMech	
T5	1	11.58	0.0184	11.38	-0.0341
	4	12.02	-0.0171	11.52	-0.0095
	8	11.8	0.0510	11.96	0.0137
	12	11.64	0.0438	11.48	-0.0239
GTR	1	11.9	0.0327	11.86	0.0183
	4	11.78	-0.0193	11.82	0.0073
	8	11.89	-0.0217	11.51	0.0226
	12	10.92	-0.0277	11.33	-0.0314
MT5	1	12.3	0.0827	11.96	0.0143
	4	13.15	0.0696	13.17	0.0652
	8	11.89	-0.0148	12.09	0.0493
	12	13.44	0.1179	11.61	-0.0443
MBERT	1	12.37	-0.0026	11.33	0.0016
	4	11.84	-0.0431	11.58	-0.0019
	8	11.58	-0.0333	12.93	0.1003
	12	11	-0.0242	11.48	0.0271
OPENAI (ADA-2)	1	12.33	0.0198	11.15	-0.0020
	4	12.31	0.0133	11.87	-0.0417
	8	10.87	-0.0659	13.13	0.1078
	12	11.07	-0.0050	12.48	0.0447

Table 9: The Inversion Performance on Classification Tasks on SNLI dataset with Local DP. From a defender's perspective, ↓ means lower are better.

Using local DP, while the utility performance is preserved almost as the non-private embeddings with $\epsilon = 12$, as shown in Fig. 4, Tabel 14 and 15. However, the inversion performance still maintains more than 25% of the non-private embeddings in Rouge-L across encoders for both LapMech and PurMech, as detailed in Table 9, 14 and 15, posing security and privacy risks for the embeddings.

7 Discussion and Conclusion

In this work, we introduce and validate the effectiveness of a novel few-shot inversion attack, **ALGEN**, which drastically lowers the cost and complexity of such attacks on widely used vector databases. Our results show that the attack transfers effectively across domains and languages while revealing critical information. Moreover, its ability to align embeddings from different LLMs with minimal loss highlights its broad NLP applications, especially in cross-lingual embedding alignment. Finally, our evaluation of existing defense mechanisms reveals that none can adequately protect textual embeddings from inversion attacks while maintaining utility, highlighting significant security and privacy vulnerabilities.

Limitations

Our work does not propose a sufficient defense mechanism for **ALGEN**. Although we evaluated a number of existing defense mechanisms for textual embeddings, we found them to be ineffective against the proposed embedding inversion attack. The primary focus of this work is to expose the security vulnerabilities in embedding services and to inspire the development of future defense paradigms.

Computational Resources

We conduct experiments and train each text-to-embedding generator model on a single Nvidia A40 GPU, with the training process completing in three hours. Beyond this, **ALGEN** requires minimal GPU resources, making it a genuinely few-shot experimental setting.

Ethics Statement

We comply with the ACL Ethics Policy. The inversion attacks implemented in this paper can be misused and potentially harmful to proprietary embeddings. We discuss and experiment with potential mitigation and defense mechanisms, and we encourage further research in developing effective defenses in this attack space.

Acknowledgements

YC and JB are funded by the Carlsberg Foundation, under the Semper Ardens: Accelerate programme (project nr. CF21-0454). We further acknowledge the support of the AAU AI Cloud and express our gratitude to DeiC for providing computing resources on the LUMI cluster (project nr. DeiC-AAU-N5-2024085-H2-2024-28). YC expresses her gratitude to the Danish Ministry of Higher Education and Science for the EliteForsk travel grant and to the School of Computing at Macquarie University for hosting her research stay. Q. Xu acknowledges support from 2024 FSE Strategic Startup and FSE Travel Grant. We also thank the anonymous reviewers for their constructive feedback, which helped us improve this paper.

References

Prince Osei Aboagye, Yan Zheng, Chin-Chia Michael Yeh, Junpeng Wang, Wei Zhang, Liang Wang, Hao Yang, and Jeff M Phillips. 2022. Normalization of language embeddings for cross-lingual alignment. In *International Conference on Learning Representations*.

Hanan Aldarmaki and Mona Diab. 2019. Context-aware cross-lingual mapping. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3906–3911, Minneapolis, Minnesota. Association for Computational Linguistics.

Sawsan Alqahtani, Garima Lalwani, Yi Zhang, Salvatore Romeo, and Saab Mansour. 2021. Using optimal transport as alignment objective for fine-tuning multilingual contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3904–3919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Manish Bhattarai, Ryan Barron, Maksim E. Eren, Minh N. Vu, Vesselin Grantcharov, Ismael Ismael, Valentin Stanev, Cynthia Matuszek, Vladimir I Valtchinov, Kim Rasmussen, and Boian S. Alexandrov. 2025. HEAL: Hierarchical embedding alignment loss for improved retrieval and representation learning. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 205–214, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2021. mmarco: A multilingual version of the ms marco passage ranking dataset. *arXiv preprint arXiv:2108.13897*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations.

Yiyi Chen, Russa Biswas, Heather Lent, and Johannes Bjerva. 2024a. Against all odds: Overcoming typology, script, and language confusion in multilingual embedding inversion attacks. *arXiv preprint arXiv:2408.11749*.

Yiyi Chen, Heather Lent, and Johannes Bjerva. 2024b. Text embedding inversion security for multilingual language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7808–7827.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1116–1128, Torino, Italia. ELRA and ICCL.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Minxin Du, Xiang Yue, Sherman SM Chow, and Huan Sun. 2023. Sanitizing sentence embeddings (and labels) for local differential privacy. In *Proceedings of the ACM Web Conference 2023*, pages 2349–2359.

Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Robert M Gray et al. 2006. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239.

Yu-Hsiang Huang, Yuche Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. 2024. Transferable Embedding Inversion Attack: Uncovering Privacy Risks in Text Embeddings without Model Queries. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4193–4205, Bangkok, Thailand. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kevin Krahn, Derrick Tate, and Andrew C. Lamicela. 2023. Sentence embedding models for ancient greek using multilingual knowledge distillation.

Ivana Kvapilíková, Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Ondřej Bojar. 2020. Unsupervised multilingual sentence embeddings for parallel corpus mining. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 255–262, Online. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International conference on learning representations*.

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 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14022–14040, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov, Frank Hutter, et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. Differentially private representation for NLP: Formal guarantee and an empirical study on privacy and fairness. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2355–2365, Online. Association for Computational Linguistics.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

John X Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. 2023. Text embeddings reveal (almost) as much as text. *arXiv preprint arXiv:2310.06816*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large dual encoders are generalizable retrievers.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Anudeex Shetty, Qiongkai Xu, and Jey Han Lau. 2024. Wet: Overcoming paraphrasing vulnerabilities in embeddings-as-a-service with linear transformation watermarks. *arXiv preprint arXiv:2409.04459*.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019a. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 796–806, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019b. Cross-lingual BERT transformation for zero-shot dependency parsing. 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 5721–5727, Hong Kong, China. Association for Computational Linguistics.

Julia Wiesinger, Patrick Marlow, and Vladimir Vuskovic. 2025. Agents. Accessed: 2025-01-07.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Arihan Yadav and Alan McMillan. 2024. Mind the gap: A generalized approach for cross-modal embedding alignment. *arXiv preprint arXiv:2410.23437*.

A Derivation of Normal Equation

To determine the optimal alignment matrix $\boldsymbol{W}$, we aim to minimize a cost function J that quantifies the discrepancy between the attack embedding matrix $\boldsymbol{E}_A$ and the transformed victim embeddings $\boldsymbol{E}_{V\to A} = \boldsymbol{E}_V\boldsymbol{W}$:

$$\begin{aligned} J(\boldsymbol{W}) &= \frac{1}{2}(\boldsymbol{E}_A - \boldsymbol{E}_V\boldsymbol{W})^T(\boldsymbol{E}_A - \boldsymbol{E}_V\boldsymbol{W}) \\ &= \frac{1}{2}(\boldsymbol{E}_A^T\boldsymbol{E}_A - \boldsymbol{E}_A^T\boldsymbol{E}_V\boldsymbol{W} - (\boldsymbol{E}_V\boldsymbol{W})^T\boldsymbol{E}_A \\ &+ (\boldsymbol{E}_V\boldsymbol{W})^T\boldsymbol{E}_V\boldsymbol{W}) \\ &= \frac{1}{2}(\boldsymbol{E}_A^T\boldsymbol{E}_A - \boldsymbol{E}_A^T\boldsymbol{E}_V\boldsymbol{W} - \boldsymbol{W}^T\boldsymbol{E}_V^T\boldsymbol{E}_A \\ &+ \boldsymbol{W}^T\boldsymbol{E}_V^T\boldsymbol{E}_V\boldsymbol{W}). \end{aligned} \quad (8)$$

By calculating the derivatives of $J(\boldsymbol{W})$, we have

$$\begin{aligned} \nabla_{\boldsymbol{W}} J(\boldsymbol{W}) &= \frac{1}{2}\nabla_{\boldsymbol{W}}(\boldsymbol{E}_A^T\boldsymbol{E}_A - \boldsymbol{E}_A^T\boldsymbol{E}_V\boldsymbol{W} \\ &- \boldsymbol{W}^T\boldsymbol{E}_V^T\boldsymbol{E}_A + \boldsymbol{W}^T\boldsymbol{E}_V^T\boldsymbol{E}_V\boldsymbol{W}) \\ &= 2\boldsymbol{E}_V^T\boldsymbol{E}_V\boldsymbol{W} - 2\boldsymbol{E}_V^T\boldsymbol{E}_A. \end{aligned} \quad (9)$$

The optimized $\boldsymbol{W}$ is achieved when the derivative is equal to 0,

$$\boldsymbol{E}_V^T\boldsymbol{E}_V\boldsymbol{W} = \boldsymbol{E}_V^T\boldsymbol{E}_A. \quad (10)$$

Then, the matrix $\boldsymbol{W}$ that minimizes $J(\boldsymbol{W})$ is

$$\boldsymbol{W} = (\boldsymbol{E}_V^T\boldsymbol{E}_V)^{-1}\boldsymbol{E}_V^T\boldsymbol{E}_A. \quad (11)$$

B Defense Mechanisms

B.1 WET

$\boldsymbol{T}$ is constructed by adopting circulant matrices (Gray et al., 2006) to ensure that the transformation matrix is both full-rank and well-conditioned to allow for accurate pseudoinverse computation for recovering the original embeddings from watermarked embeddings (Shetty et al., 2024), refer to Shetty et al. (2024) for the complete algorithm for generating $\boldsymbol{T}$.

In detail, WET as a defense is applied to aligned embeddings with the equation 12, where $\boldsymbol{W}$ is the optimal solution for alignment, and $\boldsymbol{T}$ is invertible.

$$\begin{aligned} \text{Norm}(\boldsymbol{T}\boldsymbol{E}_{V\to A}) &= \text{Norm}(\boldsymbol{T}(\boldsymbol{E}_V\boldsymbol{W})) \\ &= \text{Norm}((\boldsymbol{T}\boldsymbol{E}_V)\boldsymbol{W}) \end{aligned} \quad (12)$$

Model	Huggingface	Architecture	#Languages	Dimension
FLAN-T5 (Chung et al., 2022)	google/flan-t5-small	Encoder-Decoder	60	512
GTR (Ni et al., 2021)	sentence-transformers/gtr-t5-base	Encoder	1	768
T5 (Raffel et al., 2023)	google-t5/t5-base	Encoder-Decoder	4	768
MT5 (Xue et al., 2021)	google/mt5-base	Encoder-Decoder	102	768
MBERT (Devlin et al., 2019)	google-bert/bert-base-multilingual-cased	Encoder	104	768
TEXT-EMBEDDING-ADA-002	OpenAI API	Encoder	100+	1536
TEXT-EMBEDDING-3-LARGE	OpenAI API	Encoder	100+	3072

Table 10: Details of LLMs and Embeddings.

B.2 (Local) Differential Privacy (DP)

As illustrated in Du et al. (2023), DP ensures that a randomized mechanism $\mathcal{M}$ behaves similarly on any two neighboring datasets $\mathcal{X} \simeq \mathcal{X}'$ differing in only one individual's contribution (e.g., a sequence). It is formally defined as follows:

Definition B.1. Let $\epsilon \geq 0$, $0 \leq \delta \leq 1$ be *two privacy parameters*. $\mathcal{M}$ fulfills $(\epsilon, \delta)-$DP, if $\forall \mathcal{X} \simeq \mathcal{X}'$ and any output set $\mathcal{O} \subseteq Range(\mathcal{M})$, $\Pr[\mathcal{M}(\mathcal{X}) \in \mathcal{O}] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(\mathcal{X}') \in \mathcal{O}] + \delta$.

If $\delta = 0$, then we say that $\mathcal{M}$ is $\epsilon-$DP or pure DP.

There are two popular DP settings, *central* and *local*. In central DP, a trusted curator can access the raw data of all individuals, apply a Mechanism $\mathcal{M}$ with random noise to ensure DP, and then release the perturbed outputs. Local DP (LDP) is ensured without the curator by letting individuals perturb their data locally before being shared. The local DP (Kasiviswanathan et al., 2011) is defined as follows:

Definition B.2. Let $\epsilon \geq 0$ be a *privacy parameter*. $\mathcal{M}$ is $\epsilon-$LDP, if for any two private inputs $\mathcal{X}$, $\mathcal{X}'$ and *any output set* $\mathcal{O} \subseteq Range(\mathcal{M})$, $\Pr[\mathcal{M}(\boldsymbol{X}) \in \mathcal{O}] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(\mathcal{X}') \in \mathcal{O}]$.

However, $\epsilon - LDP$ offers homogenous protection for all input pairs, which can be too stringent in certain scenarios. When ϵ is too small, the noisy outputs are useless for utility tasks.

Thus, Du et al. (2023) customizes heterogeneous privacy guarantees for different pairs of inputs, so called metric-based LDP, formally defined as follows:

Definition B.3. Let $\epsilon \geq 0$ be the *privacy parameter*, and d be a suitable distance metric for the input space. $\mathcal{M}$ satisfies ϵd-LDP, if for any two inputs $\mathcal{X}$, $\mathcal{X}'$ and any output set $\mathcal{O} \subseteq Range(\mathcal{M})$, $\Pr[\mathcal{M}(\boldsymbol{X}) \in \mathcal{O}] \leq e^{\epsilon d(\boldsymbol{X}, \boldsymbol{X}')} \cdot \Pr[\mathcal{M}(\boldsymbol{X}') \in \mathcal{O}]$.

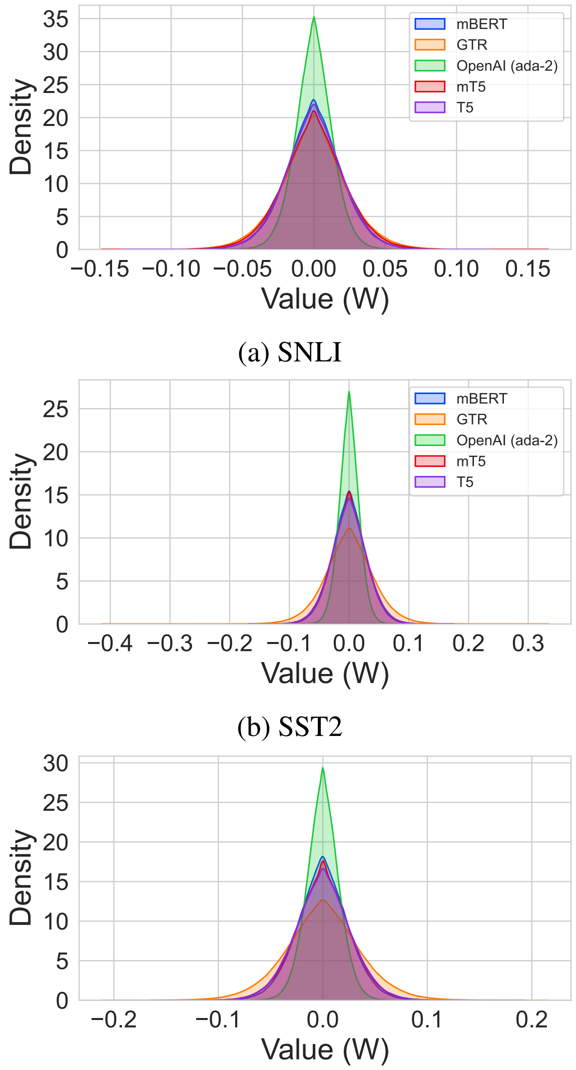
Purkayastha Mechanism with Purkayastha distribution and *Planar Lapalace Mechanism* with Euclidean metric, are thus proposed to ensure ϵd-LDP on embeddings. Refer to Du et al. (2023) for details in transforming embeddings with these mechanisms.

C Ablation Study of Leakage Data Size

The more data for alignment, the better the performance for **ALGEN**. However, after a certain amount of data, i.e., 3K, the increase in data samples does not boost the inversion performance. We conduct an ablation study of the sizes of leakage data for alignment in terms of inversion performances. Fig. 6 shows the inversion performance in Cosine Similarity (Top) and Rouge-L (Bottom) with the leakage data sizes from 1 to 8k. As shown in Fig. 6 (Top), embeddings for alignment are the perfect match for cosine similarities from 1 to 100 samples, then they decrease until they converge with the cosine similarities for aligned test embeddings for the corresponding encoder. In Fig. 6 (Top), while the inversion performances in Rouge-L increase with more data points for alignment until 8k across the encoders, the performance increases sharply from 1 to 1k. It becomes relatively stagnant from 2k to 8k. Considering the trade-off between inversion performance and the size of data samples, we choose 1k as the upper bound of the number of data samples for alignment to conduct thorough experimentation in this work.

Input	KOffice Project Reviews Starts: **1, 2, 3, 4, 5 [CARDINAL]** with comment only In chronological order from new to old - KOffice - OSDN Download.
Reconstructed	DevOps Project Reviews Starts: **1, 2, 3, 4, 5 [CARDINAL]** with comment only In chronological order from new to old - DevOp
Input	**TripAdvisor [ORG]** is proud to partner with **Travelocity, Expedia, Hotels.com, Agoda, Booking.com, Price [ORG]**.
Reconstructed	**TripAdvisor [ORG]** is proud to partner with **Booking.com, Expedia, Hotels.com, Travelocity, Agoda, Price [ORG]**
Input	Step **5 [CARDINAL]**: Utilize Windows System Restore to """"Undo"""" Recent System Changes Windows System Restore allows you to """"go back in.
Reconstructed	Step **5 [CARDINAL]**: Utilize Windows System Restore to """"Undo"""" Recent System Changes Windows System Restore allows you to """"go back in
Input	If you want to ensure you grab a bargain, try to book **more than 90 days [DATE]** before your stay to get the best price for a **Paris [GPE]**.
Reconstructed	If you want to ensure you grab a bargain, try to book **more than 90 days [DATE]** before your stay to get the best price for a **Paris [GPE]**
Input	The Fund's total amount for the Fund is limited to a maximum of **$4,000,000 [MONEY]**, and the Fund's total amount for each transaction is
Reconstructed	means an amount up to **USD 4,000,000 [MONEY]**, for the Winners, the exact amount is subject to the sole and final discretion of the Fund;
Input	**Microsoft [ORG]** is constantly updating and improving **Windows [PRODUCT]** system files that could be associated with 100street_bkg_bikini_bottom.swf.
Reconstructed	**Microsoft [ORG]** is constantly updating and improving **Windows [PRODUCT]** system files that could be associated with jabber-shp-src.jar. Sometimes

Table 11: Qualitative Analysis of In-Domain Inversion Results from OPENAI (ADA-2) embeddings with 1k alignment data samples and the attack model trained on MultiPHLT English dataset. The matched entities with their entity types are colored and **bolded** in **Input** and **Reconstructed**. The mismatched reconstructed texts are in grey colored box . [GPE]: Countries/cities/states; [ORG]:Organization.



(a) SNLI

(b) SST2

(C) S140

Figure 5: The Analysis of Alignment Transformation Weight ($\boldsymbol{W}$) on Victim Encoders on different datasets.

Victim	Defense	Rouge-L↓	COS↓	ACC↑	F1↑
RANDOM	-	6.57	-0.0108	56.50	56.02
T5	-	25.06	0.9359	89.5	89.5
	WET	24.52	0.9344	87	87
	Shuffling	20.34	0.9358	89	89
	0.001	25.15	0.9355	89.5	89.5
	0.005	24.59	0.9308	88.5	88.5
	0.01	22.06	0.9172	88	88
	0.05	12.11	0.7637	87.5	87.5
	0.1	9.57	0.7092	79	79
	0.5	7.03	0.3169	59	59
	1	7.05	0.1878	56.5	56.41
GTR	-	18.14	0.8823	85.5	85.5
	WET	15.69	0.9178	84	83.97
	Shuffling	14.69	0.8835	85.5	85.5
	0.001	17.22	0.8804	85	84.99
	0.005	17.01	0.8748	85	84.99
	0.01	15.09	0.8489	85	84.99
	0.05	9.91	0.7136	83.5	83.5
	0.1	9.18	0.6505	76	75.91
	0.5	7.53	0.2297	55	54.93
	1	6.62	0.0293	49	48.87
MT5	-	21.83	0.9320	79.5	79.47
	WET	21.04	0.9307	80	79.93
	Shuffling	18.19	0.9321	78.5	78.47
	0.001	22.13	0.9327	77.5	77.43
	0.005	21.71	0.9277	80	79.99
	0.01	18.97	0.9119	79.5	79.41
	0.05	9.72	0.7410	77.5	77.34
	0.1	8.83	0.6924	64	62.79
	0.5	7.55	0.2407	51	50.98
	1	6.55	0.1930	49	48.95
MBERT	-	20.44	0.9211	76.5	76.06
	WET	20.19	0.9261	77	76.66
	Shuffling	17.07	0.9209	76	75.59
	0.001	20.32	0.9209	76.5	76.06
	0.005	20.01	0.9156	77	76.6
	0.01	18.22	0.9018	76	75.59
	0.05	10.97	0.7320	69.5	68.54
	0.1	8.6	0.6851	64.5	64.02
	0.5	7.06	0.3407	59.5	59.5
	1	6.12	0.0838	46	45.86
OPENAI (ADA-2)	-	20.15	0.9309	92	92
	WET	15.83	0.9258	91	91
	Shuffling	17.74	0.9311	91.5	91.5
	0.001	20.11	0.9308	91	91
	0.005	20.01	0.9300	91	91
	0.01	17.57	0.9179	90	90
	0.05	9.39	0.7787	82.5	82.5
	0.1	8.07	0.7502	71	70.9
	0.5	7.81	0.4743	51.5	51.4
	1	5.92	0.1324	53.5	52.2

Table 12: The Inversion and Utility Performance on Classification Tasks on SST2 dataset with WET, Shuffling, Guassian Noise Insertion. From a defender's perspective, ↑ means higher are better, ↓ means lower are better.

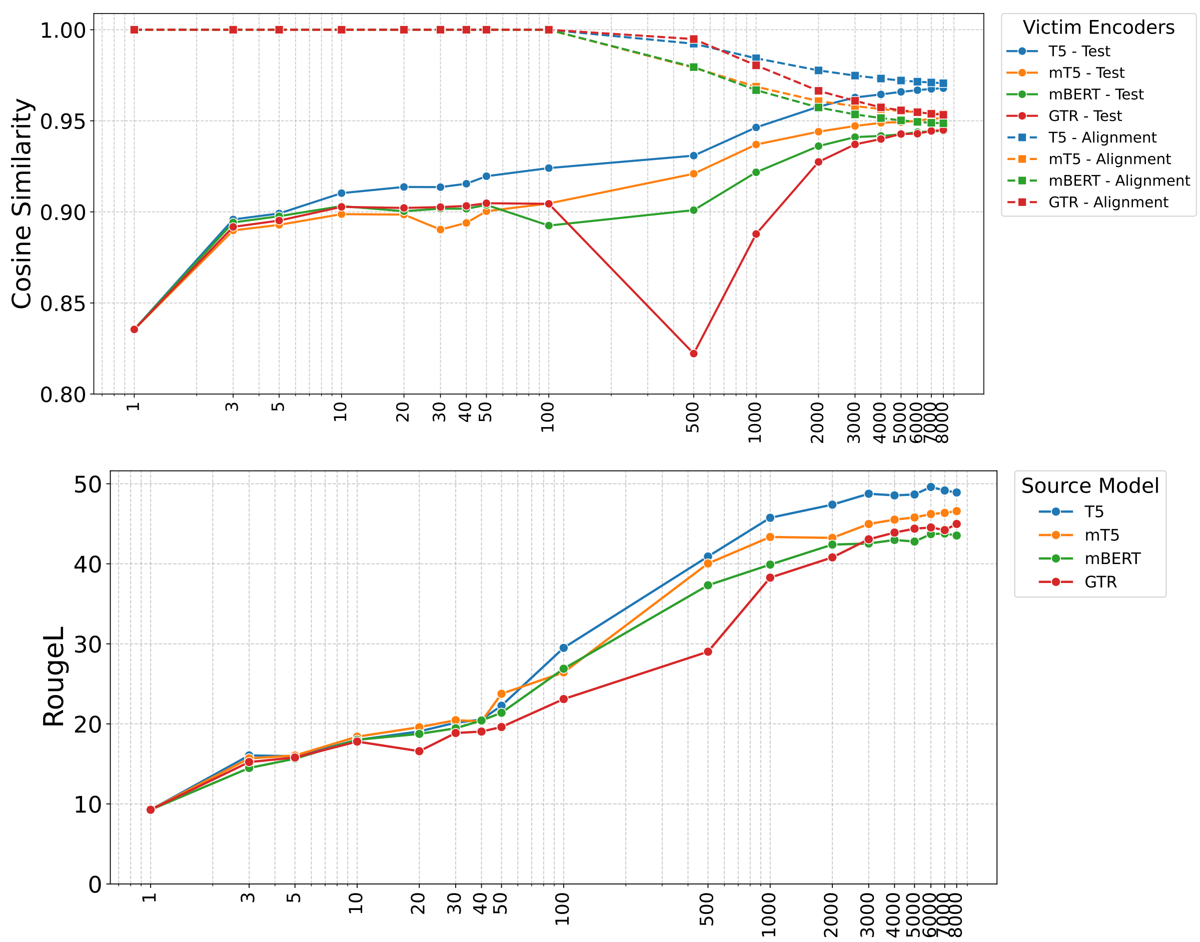


Figure 6: Inversion Performance vs. Leakage Data (Alignment) Sizes. (Top) shows the cosine similarities for embeddings of alignment data (dashed) and for embeddings of test data by the size of alignment data. (Bottom) shows the Rouge-L scores for the reconstructed texts compared to the original texts.

Victim	Defense	Rouge-L↓	COS↓	ACC↑	F1↑
RANDOM	-	5.01	0.0275	48.5	48.13
T5	-	23.04	0.9219	71.5	70.37
	WET	19.86	0.9186	69	67.42
	Shuffling	17.56	0.9245	71	70.03
	0.001	22.05	0.9230	71.5	70.37
	0.005	21.11	0.9188	70.5	69.45
	0.01	18.29	0.9078	69	67.84
	0.05	8.85	0.7561	68	66.8
	0.1	7.6	0.6901	59.5	58.86
	0.5	4.97	0.2678	46	45.95
	1	4.85	0.1101	50.5	50.47
GTR	-	13.22	0.8585	70.5	68.45
	WET	12.15	0.8861	71.5	69.35
	Shuffling	11.69	0.8599	70	68
	0.001	13.59	0.8594	69.5	67.38
	0.005	12.86	0.8515	70	68.17
	0.01	12.11	0.8260	71	69.23
	0.05	7.92	0.7055	70.5	69.21
	0.1	5.8	0.6137	64	63.28
	0.5	4.6	0.3075	56.5	56.47
	1	4.23	0.0582	53	52.83
MT5	-	19.82	0.9212	68	66.8
	WET	18.4	0.9219	63	61.61
	Shuffling	16.51	0.9215	66.5	65.44
	0.001	20.4	0.9217	67.5	66.47
	0.005	18.82	0.9173	67	65.89
	0.01	16.48	0.9037	65.5	64.28
	0.05	8.09	0.7625	58.5	57.03
	0.1	7.02	0.6952	60	59.6
	0.5	5.13	0.3770	55.5	55.31
	1	4.21	0.1505	42	41.99
mBERT	-	17.88	0.9070	64.5	64.18
	WET	16.83	0.9151	63	62.63
	Shuffling	13.8	0.9062	63.5	63.29
	0.001	17.83	0.9077	64	63.77
	0.005	17.01	0.9020	64.5	64.24
	0.01	15.44	0.8901	63.5	63.17
	0.05	8.47	0.7498	60.5	60.21
	0.1	7.26	0.6876	52	50.39
	0.5	5.25	0.2904	50	49.68
	1	4.34	0.1619	55.5	55.31
OPENAI (ADA-2)	-	17.13	0.9224	71.5	70.12
	WET	14.19	0.9237	69.5	67.55
	Shuffling	15.97	0.9229	71	69.53
	0.001	17.12	0.9225	71	69.53
	0.005	17.19	0.9208	72.5	71.17
	0.01	14.8	0.9100	71.5	70.12
	0.05	7.4	0.7974	74	73.13
	0.1	6.42	0.7691	67.5	67.4
	0.5	5.36	0.4271	53.5	53.16
	1	5.27	0.1753	55.5	54.09

Table 13: The Inversion and Utility Performance on Classification Tasks on S140 dataset with WET, Shuffling, Guassian Noise Insertion. From a defender's perspective, ↑ means higher are better, ↓ means lower are better.

Victim Model	ϵ	Rouge-L	COS	ACC	F1	Rouge-L	COS	ACC	F1
		LapMech				PurMech			
T5	-	25.06	0.9359	89.5	89.5	23.04	0.9219	71.5	70.37
	1	6.84	0.0103	54.5	54.5	4.19	-0.0235	55	54.93
	4	5.94	-0.0660	76	75.96	4.48	0.0322	63.5	63.39
	8	6.53	0.0526	84.5	84.5	4.29	-0.1135	71	70.03
	12	6.78	0.0551	89.5	89.49	4.29	-0.0210	71	70.03
GTR	-	18.14	0.8823	85.5	85.5	13.22	0.8585	70.5	68.45
	1	5.86	-0.0213	55.5	55.5	4.29	-0.0079	50	49.82
	4	6.64	0.0232	73	72.98	4.55	-0.0066	61	58.4
	8	6.75	0.0172	83	82.99	4.82	-0.0179	67	65.16
	12	7.05	0.0157	84	84	4.85	0.0439	70	68.75
MT5	-	21.83	0.9320	79.5	79.47	17.13	0.9224	71.5	70.12
	1	7.18	-0.0850	54	53.98	3.89	-0.0560	55	54.93
	4	6.78	0.0471	69.5	69.49	4.09	-0.0255	66.5	65.03
	8	6.75	-0.0008	78	78	4.9	-0.0166	68.5	66.66
	12	6.71	0.0638	81	80.95	5.3	0.1235	69.5	67.72
MBERT	-	20.44	0.9211	76.5	76.06	17.88	0.9070	64.5	64.18
	1	6.59	0.0182	50.5	50.5	4.23	0.0215	49.5	49.5
	4	6.11	-0.1486	70	69.97	4.6	-0.0190	63.5	63.46
	8	6.76	-0.0680	76	75.8	4.26	0.0942	63	62.91
	12	6.55	0.0431	78	77.68	4.66	-0.0340	62	61.69
OPENAI (ADA-2)	-	20.15	0.9309	92	92	19.82	0.9212	68	66.8
	1	6.06	-0.0289	50.5	50.47	4.52	0.0507	53	52.98
	4	6.56	0.0336	79	79	4.14	-0.0377	63.5	62.6
	8	6.65	0.1389	90.5	90.5	4.31	0.0997	67	65.76
	12	6.01	-0.0267	94.5	94.5	4.24	0.0022	66	64.99

Table 14: The Inversion and Utility Performance on Classification Tasks on SST2 dataset with local DP. From a defender's perspective, ↑ means higher are better, ↓ means lower are better.

Victim Model	ϵ	Rouge-L	COS	ACC	F1	Rouge-L	COS	ACC	F1
		LapMech				PurMech			
T5	-	23.04	0.9219	71.5	70.37	23.04	0.9219	71.5	70.37
	1	4.23	0.0109	52.5	52.49	4.19	-0.0235	55	54.93
	4	4.23	-0.0526	61.5	60.67	4.48	0.0322	63.5	63.39
	8	4.19	0.0457	68.5	67.5	4.29	-0.1135	71	70.03
	12	4.33	0.0082	72	71.06	4.29	-0.0210	71	70.03
GTR	-	13.22	0.8585	70.5	68.45	13.22	0.8585	70.5	68.45
	1	4.54	-0.0107	51	50.82	4.29	-0.0079	50	49.82
	4	4.89	0.0635	61.5	59.81	4.55	-0.0066	61	58.4
	8	4.55	0.0415	67	65.48	4.82	-0.0179	67	65.16
	12	4.64	0.0673	70.5	68.93	4.85	0.0439	70	68.75
MT5	-	19.82	0.9212	68	66.8	19.82	0.9212	68	66.8
	1	4.54	-0.0650	47	46.99	4.52	0.0507	53	52.98
	4	4.31	0.0432	61.5	60.67	4.14	-0.0377	63.5	62.6
	8	4.85	0.0386	65.5	64.86	4.31	0.0997	67	65.76
	12	4.62	0.1038	66	64.99	4.24	0.0022	66	64.99
MBERT	-	17.88	0.9070	64.5	64.18	4.66	-0.0340	62	61.69
	1	4.6	0.0153	54	54	17.88	0.9070	64.5	64.18
	4	5	0.1196	59	58.9	4.23	0.0215	49.5	49.5
	8	4.4	0.0156	60.5	60.28	4.6	-0.0190	63.5	63.46
	12	4.76	0.0117	61.5	61.45	4.26	0.0942	63	62.91
OPENAI (ADA-2)	0	17.13	0.9224	71.5	70.12	17.13	0.9224	71.5	70.12
	1	4.45	0.0041	53	52.83	3.89	-0.0560	55	54.93
	4	4.84	0.0572	63	62.46	4.09	-0.0255	66.5	65.03
	8	4.66	-0.0191	68.5	66.97	4.9	-0.0166	68.5	66.66
	12	4.84	0.0728	71.5	69.83	5.3	0.1235	69.5	67.72

Table 15: The Inversion and Utility Performance on Classification Tasks on S140 dataset with local DP. From a defender's perspective, ↑ means higher are better, ↓ means lower are better.