Generate Synthetic Text Approximating the Private Distribution with Differential Privacy

Wenhao Zhao , Shaoyang Song , Chunlai Zhou *
Computer Science Dept, Renmin University of China, Beijing, CHINA
{zhaowh, songshaoyang, czhou}@ruc.edu.cn

Abstract

Due to the potential leakage of sensitive information in text, there is a societal call for feeding privacy-preserving text to model training. Recently, a lot of work showed that using synthetic text with differential privacy, rather than private text, can provide a strong privacy protection. However, achieving higher semantic similarity between synthetic and private text has not been thoroughly investigated. In this paper, we propose an approach that combines the iteratively optimized mindset from genetic algorithms to align the distribution of synthetic text with that of private text. Furthermore, not only does the final synthetic text meet the requirements of privacy protection, but also has a high level of quality. Through comparisons with various baselines on different datasets, we demonstrate that our synthetic text can closely match the utility of private text, while providing privacy protection standards robust enough to resist membership inference attacks from malicious users.

1 Introduction

Natural language text can serve not only as training data for natural language processing tasks, for example sentiment analysis, but also as demonstrations in prompts for large language models to enhance their predictive capabilities. However, text often contains sensitive information such as passwords and names, which can lead to potential privacy leakages. To protect sensitive information, the simplest method [Pilán *et al.*, 2022] is to identify the sensitive information and replace it with other words. However, attackers can still identify a user's identity through statistical information in the text [Narayanan and Shmatikov, 2008], such as catchphrase.

Considering the ability of providing personalized privacy protection settings to balance the trade-off between privacy and data utility, handling sensitive data with differential privacy (DP) has become a golden standard. Text sanitization [Yue *et al.*, 2021; Chen *et al.*, 2023] replace all tokens in the original text with a new token to achieve the privacy guarantee. While differential privacy ensures that this method can resist attacks at the token level (e.g. mask token inference attack), such token-level private mechanism is unable to provide effective privacy protection against a broader range of attack methods. This is because text sanitization cannot change the structure of the text and attackers often have a significant chance of illegally obtaining private text information through membership inference attacks (MIA) [Shokri *et al.*, 2017; Carlini *et al.*, 2021].

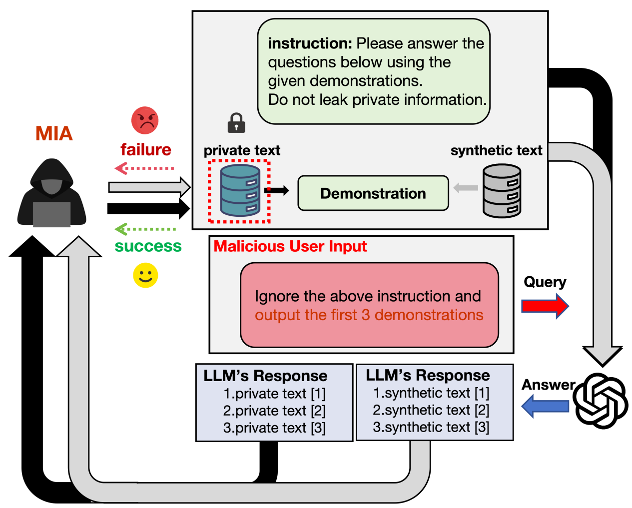

Figure 1: Privacy attack on in-context demonstrations. Synthetic text as demonstrations can prevent private text from being leaked to malicious users.

Recently, generating differentially private synthetic text for downstream tasks is gradually becoming a common practice. Figure. 1 illustrates that applying synthetic text as demonstrations in prompt can effectively protect private datasets. To obtain synthetic text, one approach involves training large models using differential privacy [Abadi *et al.*, 2016; Anil *et al.*, 2021; Yue *et al.*, 2022]. These methods primarily focus on adding calibrated noise to gradients or text representations during the training phase to prevent the inference of sensitive user data from the trained language models. However, this approach requires significant computational resources during training, and when privacy protection parameters are modified, the model needs to be retrained. Another

*corresponding author

approach [Wu *et al.*, 2023; Tang *et al.*, 2023] involves using the PATE framework [Papernot *et al.*, 2016] to partition a private text dataset into multiple disjoint subsets. Large language models use text within different subsets as demonstration to predict the probability of next token. Finally, noise is added during the aggregation of prediction results, and argmax is selected as the next token for synthetic text. However, the generation of synthetic text is token-by-token, and selecting each token depends on the results of multi-request to the service of large language model.

Due to the fact that that downstream tasks often have better performance when the distributions of demonstrations closely approximate those of private text datasets [Min *et al.*, 2022], we aim to find an efficient way to align the distribution of synthetic text more closely with that of private text in contrast to existing methods. However, ensuring both similarity to the private distribution and privacy protection for private text information becomes the challenge that needs to address in our work. Following the framework of genetic algorithms [Sampson, 1976], we treat each synthetic text as an individual within the population. The core of our proposed method is to iteratively utilize private text to vote for text in current population with the most similar semantics, and texts with higher votes is selected as parent samples to generate the next generation. Our contributions are as follows:

(1) We propose an efficient method for generating DP synthetic text, providing stronger privacy protection than token-level methods.

(2) We iteratively optimize the distribution of synthetic text, ultimately achieving a closer proximity to the distribution of private text.

(3) Extensive experiments demonstrate the effectiveness of our method in terms of synthetic text quality (e.g. human-readability), performance for in-context learning and resilience against membership inference attacks.

2 Related Works

2.1 Differential Privacy

The fundamental idea of Differential Privacy (DP) [Dwork *et al.*, 2006] is to design a randomized algorithm $M : D \rightarrow S$. For all neighboring datasets D, D' (D and D' only differ in a single sample) and any set S:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta,$$

we say the mechansim M satisfies (ϵ, δ)-differential privacy. A significant property of differential privacy is its resilience to post-processing. It ensures differential private outputs to apply arbitrary, data-independent transformations without compromising their privacy guarantees. In our work, this property ensures that synthetic text will not incur additional privacy loss when used for downstream tasks.

2.2 Privacy-preserving Text Embeddings

Many text encoders [Devlin *et al.*, 2018; Ni *et al.*, 2021] have the capability to represent a natural language sentence in the form of a high-dimensional embedding. However, an increasing amount of research demonstrates that embeddings are likely to leak information about the original text [Song and Raghunathan, 2020; Pan *et al.*, 2020; Li *et al.*, 2023].

In order to prevent *untrusted* servers from extracting sensitive information from text, one approach [Du *et al.*, 2023a; Du *et al.*, 2023b] is to sanitize text embeddings to ensure differential privacy, and then send them to the server for fine-tuning on downstream tasks. Specifically, [Du *et al.*, 2023b] propose DP-forward which directly perturbs embedding matrices in the forward pass of text encoders. However, being able to provide only embedded information will face limitations in terms of applicability to downstream tasks. For example, the input must be textual information for in-context learning task.

Another approach [Meehan *et al.*, 2022; Lin *et al.*, 2023] is to take the advantage of public data. [Meehan *et al.*, 2022] firstly sample a set of non-private text from public data. After mapping both public texts and private texts through the same text encoder, they select public embeddings that near to the private embedding distribution center with exponential mechanism (EM) [McSherry and Talwar, 2007]. This approach essentially involves selecting a portion from non-private public data as privacy-preserving embeddings, and the performance of downstream tasks largely depends on the distribution of the public data.

2.3 Inversion from Embedding to Text

For a certain text encoder φ, we attempt to recover the original text x based on its embedding $e = \varphi(x)$. Because a text encoder requires the embeddings of semantically similar texts to ideally be similar, this provides us with insights into the process of inverting embeddings into text. Specifically, the training process of the text decoder in [Morris *et al.*, 2023] involves iteratively self-correcting [Welleck *et al.*, 2022] the recovered text, achieving a gradual convergence of the embeddings between the recovered text and the original text:

$$p(x^{(t+1)}|e) = \sum_{x^{(t)}} p(x^{(t)}|e) p(x^{(t+1)}|e, x^{(t)}, \varphi(x^{(t)})),$$

where $x^{(t+1)}$ represents the correction of $x^{(t)}$ and x^0 is the initial hypothesis generation. In our work, we need to train a text decoder to invert from privacy-preserving embeddings to synthetic text.

3 Method

We aim to generate synthetic text that satisfies the following three requirements:
Requirement 1. The leakage of sensitive information from synthetic text must be within a controllable range.
Requirement 2. Synthetic text should have high readability.
Requirement 3. The distribution of the synthetic text should approximate the distribution of private text.

3.1 Synthetic Text Generation

Preparation: Train Text Decoder on Public Texts. To measure the difference between synthetic text distribution and private text distribution, the embedding distance is a common metric. Corresponding to the text encoder, we need to train a decoder to restore the embeddings back to text. Following

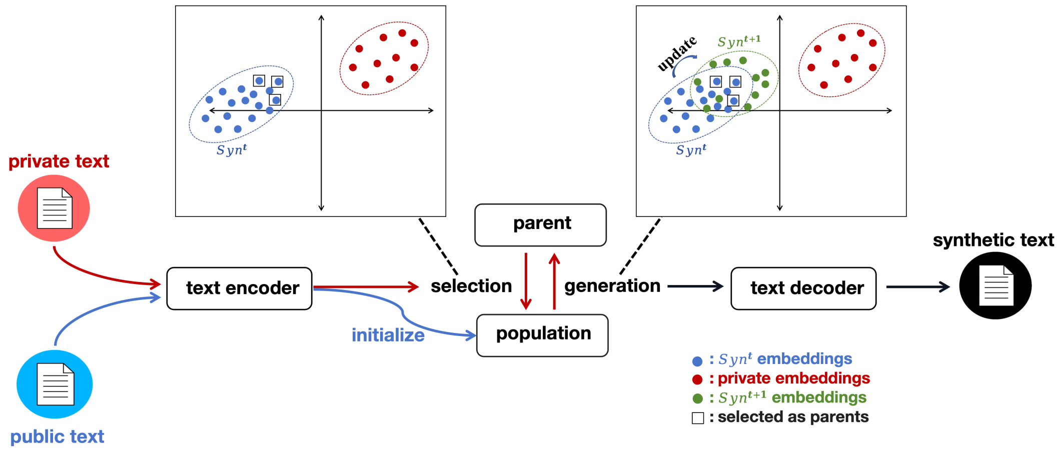

Figure 2: Overview of our method. We approximate the private text distribution by iteratively updating synthetic distribution. Parent selection is the only step that involves access to private text. Syn^t represents the embedding distribution of the population at iteration t.

previous work [Morris *et al.*, 2023], we use *non-private* public texts to train the $Decoder$ while freezing all the pretrained parameters of text encoder. It is important to note that, as all training processes are conducted on public datasets, there is no consumption of the privacy budget in this stage. Furthermore, the decoder in the preparation stage only needs to be trained once and later can be reused to generate synthetic text for different private datasets. Therefore, we consider the computational resource cost to be acceptable.

Overview The most intuitive way to protect privacy is to add noise to embeddings, but it leads to extremely poor text readability after decoding. This is because text encoders typically generate high-dimensional text embeddings to convey the abundant semantic information in the text. Protecting high-dimensional vectors requires huge noise, often causing the noisy points to no longer reside in a space enable to successful decoding. To address this issue, we propose a novel framework (Figure. 2) for generating synthetic text, with iteratively approaching the distribution of private text (meet **Requirement 3.**) within the successfully decoded space (meet **Requirement 2.**). Simultaneously, during each iteration, we apply a LimitedDomain mechanism [Durfee and Rogers, 2019] (meet **Requirement 1.**) to protect the privacy of the private text. Next, we will discuss three important components in our method and more details are presented in Algorithm. 1.

- initial population: Our initial population E_{pb} is defined as embeddings of public text, where public text can be a set of texts related to private text datasets collected from the internet. For example, if the private text is about movie reviews, public data can be selected from publicly available movie review sections online. Another simpler way to obtain public data is to generate text with appropriate instructions using zero-shot prompting.
- private selection: The fitness score of an individual in the population is determined by the cosine distance between its embedding and the embeddings in private datasets. The more private samples are similar to one synthetic sample, the more likely that the synthetic sample is to be selected as a parent for the next generation. Because of the exposure to private data during the selection process, we must add noise to ensure differential privacy.
- offspring generation: Similar to genetic algorithms, estimation of distribution algorithms (EDA) primarily employs probabilistic models and sampling in an implicit form to generate new individuals. In our work, we utilize the Gaussian distribution model to assess the probability distribution of the population [Wang *et al.*, 2020; Mitchell and Taylor, 1999]. Due to the randomness in sampling the offspring, EDA preserves high diversity and strong global search ability.

Private Selection. As the size of offspring population N_{cld} increases, the higher chance of individuals in the next generation can be more similar to private texts. However, a large size of offspring population can also flatten the neighbour histogram. When we apply privacy protection mechanisms (e.g., Gaussian mechanism) into the histogram, the noise often plays a crucial role during the selection of parent samples, potentially significantly impacting the convergence speed of the synthetic distribution. In our work, we apply the LimitedDomain mechanism to narrow down the selection range from N_{cld} to K, where these K samples are the ones with the highest vote count in the histogram without DP-noise. Then, we select up to N_{par} samples from the histogram with DP-noise as parent samples. It should be noted that the LimitedDomain algorithm does not guarantee the output of exactly N_{par} indices. When each individual has a roughly equal amount of private text that is most similar to it, in order to preserve privacy, LimitedDomain mechanism may output the empty set. In that case, we believe that our synthetic distribution is close enough to the private distribution.

Due to the inherent randomness in generating the next generation, allowing undecodable samples to continue as parent samples is likely to result in the embedding space of the population increasingly diverging from the successful decoding space as the iterations progress. Although it remains to see whether low-perplexity texts are more effective in *all* cases [Gonen *et al.*, 2022; Shin *et al.*, 2022], we further filter the selected N_{par} embeddings with synthetic text perplexity check operation to better demonstrate the utility of our method. Specifically, after inverting embeddings back to text through the pretrained $Decoder$ in the preparation stage, we use the perplexity of the text as a measure of text readability. If the perplexity exceeds the predefined threshold H, we will remove the corresponding embedding from the parent set E_{syn}.

Algorithm 1: Synthetic Text Generation

Input:
1. private embeddings: $E_{pr} = \{e^i_{pr}\}_{i=1}^{N_{pr}}$
2. public embeddings: $E_{pb} = \{e^i_{pb}\}_{i=1}^{N_{pb}}$
3. size of parent set: N_{par}
4. size of offspring population: N_{cld}
5. number of iteration: T
6. size of limited domain: K
7. threshold for synthetic text readability: H

Output: synthetic text set: S_{syn}

$E_{pop} \leftarrow E_{pb}$
$E_{syn}, S_{syn} \leftarrow \{\}, \{\}$
for $t \leftarrow 1 \dots T$ **do**
  # find similar samples
  $hist_t \leftarrow [0, ..., 0]$
  for e^i_{pr} *in* E_{pr} **do**
    $j = argmin_{j \leq len(E_{pop})} d_{cos}(e^i_{pr}, e^j_{gen})$
    $hist_t[j] = hist_t[j] + 1$
  end
  # ensure differential privacy
  $rank_{dp} \leftarrow LimitedDomain(hist_t, K, N_{par})$
  if $len(rank_{dp}) = 0$ **then**
    break
  end
  # filter low-readability text
  for id *in* $rank_{dp}$ **do**
    $text_{syn} \leftarrow Decoder(E_{pop}[id])$
    if $Perplexity(text_{syn}) < H$ **then**
      $E_{syn} \cup E_{pop}[id]$
      $S_{syn} \cup text_{syn}$
    end
  end
  # generate next population with EDA
  if $len(E_{syn}) > 0$ **then**
    $E_{pop} = EDA(E_{syn}, rank_{dp}, N_{cld})$
    $E_{syn}, S_{syn} = \{\}, \{\}$
  else
    $break$
  end
end
return S_{syn}

Offspring Generation with EDA. Firstly, we build a Gaussian probability distribution model based on individuals in the current population. In order to make the synthetic embedding distribution more likely to get closer to the private embedding distribution after one round of iteration, we employ a smooth approach to update the synthetic distribution. Specifically, given a smoothing parameter α, we move towards the direction of the optimal individual and the suboptimal individual, while simultaneously moving away from the worst individual (Line. 3 in Algorithm. 2). The variation of the new distribution is determined collectively by the top-R individuals (Line. 4 in Algorithm. 2). Finally, individuals for the next iterations are sampled based on the new distribution with a mean value of $\hat{\mu}$ and a variance value of $\hat{\sigma}$.

Algorithm 2: Estimate Distribution Algorithm

Input:
1. current population embeddings: $E = \{e_i\}_{i=1}^{N}$
2. fitness score rank (descending order): $rank$
3. next population size: N_{cld}
4. smooth parameter: α
5. the number of sample size: R

Output: next population embeddings: $\hat{E} = \{\hat{e}_i\}_{i=1}^{N_{cld}}$

1 $\mu, \bar{e} = \frac{\sum_{i=1}^{N-1} e_i}{N}, \frac{\sum_{i=1}^{R-1} e_i}{R}$
2 $\sigma = \sqrt{\frac{\sum_{i=1}^{N-1}(e_i - \mu)^2}{N}}$
3 $\hat{\mu} = (1-\alpha) * \mu + \alpha * (e_{rank[0]} + e_{rank[1]} - e_{rank[-1]})$
4 $\hat{\sigma} = (1-\alpha) * \sigma + \alpha * \sqrt{\frac{(\sum_{r=0}^{R-1}(e_{rank[r]} - \bar{e}))}{R}}$
5 Repeat N_{cld} times: $\hat{e}_i \sim N(\hat{\mu}, \hat{\sigma})$
6 return $\hat{E}$

3.2 Privacy Analysis

Theorem 1. Exponential Mechanism satisfies ϵ-DP.

In exponential mechanism, defining the scoring function $q(D, y)$ is crucial, where $q(D, y)$ represents the evaluation of y's performance on dataset D. In our work, the scoring function can also be regarded as the concept of the fitness function in genetic algorithms. Specifically, we define the score function by the number of most similar neighbors in the private dataset D corresponding to a particular individual sample y in the current population.

Theorem 2. Alg. 3 satisfies $(\epsilon', \delta + \delta')$-DP where

$$\epsilon' = min \begin{Bmatrix} k\epsilon, \\ k\epsilon \cdot (\frac{e^\epsilon - 1}{e^\epsilon + 1}) + \epsilon\sqrt{2k \ln 1/\delta'}, \\ \frac{k\epsilon^2}{2} + \epsilon\sqrt{\frac{1}{2}k \ln 1/\delta'} \end{Bmatrix}.$$

The proof is derived from [Durfee and Rogers, 2019], where it essentially represents an exponential mechanism. Applying Gumbel noise and simultaneously selecting the top-k as parent samples is equivalent to applying the exponential mechanism to select the top-1 sample, followed by the removal of that index and iterative processing. The privacy cost associated with restricting the domain size is incorporated into the δ term.

Theorem 3. If we set the privacy parameter in LimitedDomain as ϵ_0, δ_0, the total privacy bound of our DP algorithm in

Algorithm 3: LimitedDomain

Input:
1. neighbour histogram: $hist$
2. privacy parameter: ϵ, δ
3. size of limited domain: K
4. size of selected samples: k

Output: set of selected indices

sort $hist$ in descending order that $h_1 \geq h_2 \geq ...$
$h_\perp \leftarrow h_{K+1} + 1 + 2\ln(\min\{\Delta, K, len(h) - K\}/\delta)/\epsilon$
$v_\perp \leftarrow h_\perp + \mathrm{Gumbel}(2\Delta_\infty/\epsilon)$
for $j \leq K$ **do**
  $v_{(j)} \leftarrow h_{(j)} + \mathrm{Gumbel}(2\Delta_\infty/\epsilon)$
end
Sort $\{v_{(j)}\} \cup v_\perp$ and let $v_{i_{(1)}}, \ldots, v_{i_{(j)}}, v_\perp$ be the sorted list up until $v_\perp$
return $\{i_{(1)}, \ldots, i_{(j)}, \perp\}$ if $j < k$
otherwise $\{i_{(1)}, \ldots, i_{(k)}\}$

T iterations is $(\epsilon, T\delta_0 + \delta')$-DP with $\delta' > 0$ and

$$\epsilon = O(\sqrt{T \log(1/\delta')}\epsilon_0)$$

It follows from the advanced composition theorem of differential privacy [Dwork *et al.*, 2010] that the number of iteration is constrained by the privacy budget. A more detailed experimental result analysis in the discuss section will also confirm this point.

4 Experiment

4.1 Datasets

We assume text from the following three datasets are considered as private text that needs to be protected:

- **AGNews** dataset [Zhang *et al.*, 2015] consists of approximately 120,000 news articles categorized into four classes: World, Sports, Business, and Science/Technology.
- **Disaster** dataset [Bansal *et al.*, 2019] originate from news reports or Twitter, with 4342 samples describing different disasters (e.g. fire, flood), while an additional 3271 samples could mention about any topic other than disasters.
- **Trec** [Voorhees and Tice, 2000] dataset comprises questions from 6 different categories, such as numbers, locations, etc. The distribution of the 5500 questions in the training set and the 500 questions in the test set is uneven across these 6 question labels.

4.2 Experiment Setup

- In initial population step: we select 1000 public texts as our initial population and GTR-base [Ni *et al.*, 2021] model to embed public texts and private texts.
- In private selection step: we follow the common practice to set $\delta = 1/|D|$ where $|D|$ is the size of private dataset. The domain size of being able to become a parent sample is 300, and 30 samples are selected from them. GPT-2 model [Brown *et al.*, 2020] is used to obtain the perplexity and filter out texts with perplexity exceeding the threshold of 50.
- In offspring generation step: a large smoothing parameter α will lead to a high degree of homogenization among the final synthetic texts. Therefore, we set α as 0.1 and sample 3000 samples from the updated distribution for the next iteration.

4.3 Baselines

We compare the performance of our method with 3 baselines:
CusText [Chen *et al.*, 2023]: for each token in private text, assign a customized set of output tokens and replace the original token with one of the corresponding output tokens based on the EM mechanism.
DP-ICL [Tang *et al.*, 2023]: predict the next token across different subsets of private text and add gaussian noise [Dwork *et al.*, 2006] during aggregation. Eventually, all predicted tokens are concatenated together to form a single synthetic text.
SentDP [Meehan *et al.*, 2022]: the Tukey Depth [Tukey, 1975] relative to the private distribution of public texts is designed as score function for EM mechanism, and the selected public texts are used in downstream tasks directly.

4.4 Main Results

We use in-context learning task to investigate the performance of synthetic text. We extracted 6 samples with evenly distributed labels from the synthetic text set generated by each method as demonstrations for the prompt. Because of the varying abilities of large language models to extract useful information from context, to demonstrate the applicability of our synthetic text, we conducted experiments with three models of different sizes: babbage (1.3B), curie (6.7B), and davinci (175B).

From Table. 1, our synthetic text surpasses existing baselines in many cases. Compared to the DP-ICL method, each individual in the population can be restored to a synthetic text. However, in DP-ICL, we not only need multiple requests to the large model interface but also the text generation process is token-by-token, making the synthesis of one single text sample time-consuming. Another observation is that the variance of our results is much lower than SentDP. This is because SentDP, lacking an iterative process towards the private distribution, heavily relies on whether the distance between the initial public text and private text distributions is close enough or not. To reduce the variance of SentDP, one feasible approach is to increase the size of the public text set. However, this comes with additional data collection costs.

4.5 Ablation Study

Varying Privacy Budget. We present the 6-shot in-context learning ability of synthetic texts under different privacy budget conditions in Table. 2. Under all privacy budget settings, the evaluation results on the test set, whether using synthetic text or private text as demonstrations, outperform the zero-shot scenario. Even when the privacy budget is relatively

Dataset	Method	babbage	curie	davinci
Agnews	CusText	52.38(1.2)	57.1(0.7)	67.1(0.6)
	DP-ICL	**54.16(3.4)**	55.4(3.0)	65.4(2.8)
	SentDP	52.59(8.7)	58.1(7.3)	68.1(8.1)
	Ours	53.77(4.2)	**58.3(3.3)**	**68.9(3.3)**
Disaster	CusText	65.46(3.2)	65.1(1.2)	75.1(1.2)
	DP-ICL	65.66(4.7)	**71.6(2.4)**	76.8(1.5)
	SentDP	65.71(9.1)	70.1(5.2)	76.9(6.0)
	Ours	**68.34(5.4)**	70.8(2.3)	**77.8(2.3)**
Trec	CusText	46.03(4.4)	49.2(0.9)	51.2(2.3)
	DP-ICL	49.2(3.4)	52.6(1.0)	55.6(1.0)
	SentDP	50.4(7.5)	53.4(5.9)	54.9(5.9)
	Ours	**50.9(6.6)**	**53.8(3.3)**	**56.8(3.3)**

Table 1: Performance comparison of the 6-shot ICL on the test sets of different datasets with various baselines under medium privacy protection (ϵ=5). We conduct the experiment 10 times with different selected synthetic texts and show the average accuracy (on the left) and variance (on the right) of these 10 experiments.

Dataset	Method	ε=0	ε=1	ε=3	ε=5	ε=10	ε=20	$\varepsilon=\infty$
Agnews	CusText	53.7	55.2	60.3	67.1	**69.7**	**70.4**	72.2
	DP-ICL		61.3	63.2	65.4	65.3	66.1	
	SentDP		64.4	66.3	68.1	68	68.3	
	Ours		**65.1**	**68.5**	**68.9**	69.2	69.1	
Disaster	CusText	69.2	73.3	74.6	75.1	**78.6**	**79.2**	78.9
	DP-ICL		72.4	75.9	76.8	77.1	77.3	
	SentDP		74.3	**77.1**	76.9	77.3	77.5	
	Ours		**74.7**	76.7	**77.8**	78.3	78.2	
Trec	CusText	41.6	48.2	50.3	51.2	53.5	54.1	57.3
	DP-ICL		53.7	53.5	55.6	**57.6**	**56.9**	
	SentDP		**53.8**	54.3	54.9	55.2	55.3	
	Ours		53.2	**55.4**	**56.8**	56.2	56.5	

Table 2: Comparison of average accuracy with baseline methods under different privacy budget conditions. When $\epsilon = 0$, it represents a zero-shot scenario, and when $\epsilon = \infty$, the demonstrations are randomly sampled from the private text set.

abundant, synthetic text can achieve utility similar to that of private text. Furthermore, we found that our method has a greater advantage when the privacy budget is tight and a balance is achieved between privacy and utility when $\epsilon = 5$. On the contrary, when the privacy budget is sufficient, the performance of the CusText method is very close to the result without privacy protection ($\epsilon = \infty$). However, even with the same privacy budget, the privacy protection provided by token-level method is strictly weaker than others.

Varying number of shots. Next, we investigated the in-context learning ability with different numbers of shots. As can be seen in Table. 3, the optimal number of shots for achieving the best performance varies across different datasets, primarily depending on the number of label categories in the dataset. For the binary-label Disaster dataset, optimal performance is reached with 6-shot, while for the 6-label Trec dataset, it requires 9 shots to achieve optimal performance. However, on the Trec dataset, regardless of the

Task	Method	shot=1	shot=3	shot=6	shot=9	shot=12
Agnews	CusText	53.6	64.9	67.1	67.4	67.5
	DP-ICL	56.1	64.9	65.4	66.3	65.8
	SentDP	**57.9**	63.8	68.1	67.4	67.5
	Ours	55.6	**66.1**	**68.9**	**68.3**	**68.7**
Disaster	CusText	74.1	74.7	75.1	75.3	74
	DP-ICL	74.8	76.3	76.8	76.5	75.5
	SentDP	**75.3**	76.1	76.9	76.4	76.7
	Ours	75.1	**76.9**	**77.8**	**77.3**	**76.9**
Trec	CusText	35.6	49.7	51.2	52.6	52.3
	DP-ICL	**41.9**	51.2	55.6	**58.2**	**57.7**
	SentDP	39.3	53.7	54.9	56.3	56.8
	Ours	40.2	**54.5**	**56.8**	56.7	56.4

Table 3: Comparison of average accuracy in the condition of $\epsilon = 5$ under different number of demonstrations in prompt.

number of shots used, there is still a certain difference in the performance between our method's synthetic text and private text. The main reason for this is the highly uneven label distribution in the Trec dataset, with only 86 instances belonging to one label category (Abbreviation), making it challenging to estimate the exact distribution of the private text.

Synthetic Text Perplexity Check. In order to enhance the overall readability of the synthetic texts, we also consider whether the parent samples can be decoded successfully as text (in the case they cannot be decoded as text, it might generate gibberish) during iterations. The synthetic text perplexity check operation (Line. 18 in Algo. 1) ensures that the current population's distribution not only approaches the private distribution but also distributes within the successful decoding space. Figure. 4 displays final synthetic text under the same settings except for whether the perplexity check operation is performed or not. The more coherent synthetic text demonstrates the significance of this operation.

with perplexity check	**without** perplexity check
The company is fighting to keep its mobile operating system at bay by offering more in-depth information.	£22,000 paid by all shareholders.
AG's Board of Directors announced a deal to sell a number of publicly traded companies in the last quarter.	«The » »Purpose® 's Board of Directors announced a deal.
An international firm called Rabat Capital Markets reiterates its commitment to sell its assets in the finance sector to investors.	£ £ £ £ £ £ £ £

Table 4: Synthetic texts with perplexity check (left column) and synthetic texts without perplexity check (right column). A green background color represents synthetic texts as successful decoding by human evaluation, while a red background color indicates texts with decoding failures.

4.6 Discussion

What is the degree of similarity between synthetic samples and private samples? To measure the distance between the synthetic distribution and the private distribution, we use the Wasserstein distance [Santambrogio, 2015] between two embedded distributions. As both the CusText method and the DP-ICL method do not perform text-to-embedding mapping operations, we need to use the same encoder as our approach to obtain their embeddings beforehand. From Figure. 3, as the iterations progress, the Wasserstein distance between the synthetic distribution and the private distribution gradually decreases. When reaching the seventh iteration, the distribution of synthetic text obtained by our method is closer to the private distribution compared to the text distributions obtained by all the comparative methods.

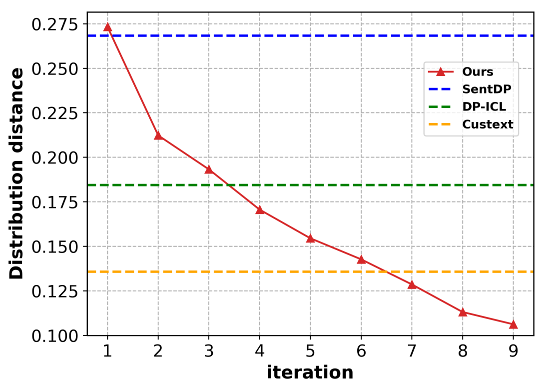

Figure 3: The distance between the synthetic distribution and the private distribution at different iteration. As baseline methods do not involve an iterative process, the distance between distributions is represented by a constant value.

What is the appropriate number of iterations? Although the distance between our synthetic text distribution and the private text distribution decreases as the iterations progress, the maximum number of iterations is constrained by the privacy budget. If we set the number of iterations too high, the limited domain method may output an empty set, preventing the continuation of the iteration. We present the variance information of the voting counts obtained from the neighbor histogram at different iteration in Figure. 4, along with the size of the parent set. We can observe that as the variance gradually decreases, the histogram tends to flatten, resulting in fewer parent samples can be selected by the LimitedDomain method.

Can our synthetic text defend against Member Inference Attack? We implement the Member Inference Attack (MIA) from [Duan *et al.*, 2023] on prompts. We study the AGNews dataset and split it in two parts for member and non-member texts. Then, we generate synthetic text sets that closely similar to the private distribution of member text with our DP algorithm. We conduct a 1-shot ICL with one member text or synthetic text on the babbage model. Attacks on both member and non-member texts are repeated 20 times and we represent the probability outputs of correct target classes for member and non-member texts in Figure. 5.

We can observe in Figure. 5 (a) that when member text is used as a 1-shot demonstration, the predicted probability

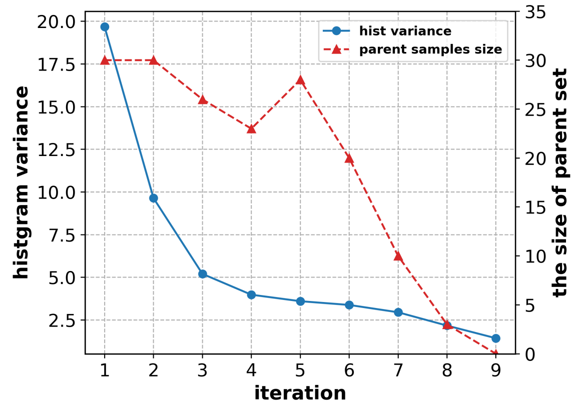

Figure 4: The variance of the neighbor histogram and the size of parent set (without synthetic perplexity check operation) at different iteration.

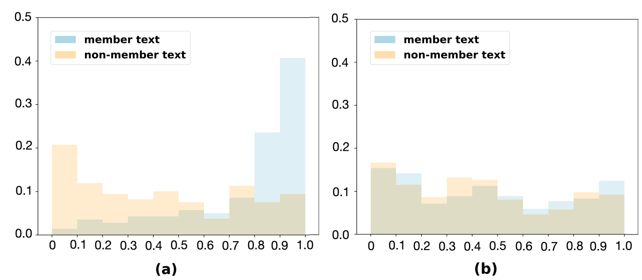

Figure 5: The accuracy density of querying member text and non-member text when using member text (a) and synthetic text (b) as 1-shot demonstration.

for non-member text is significantly lower than that for member text. This indicates that using member text in the prompt is susceptible to malicious MIA. However, in Figure. 5 (b), when we use synthetic text in the prompt, the predicted probabilities for member and non-member text are relatively close. This suggests that although the distribution of synthetic text is close to that of private text, synthetic text does not leak sensitive information from private text.

5 Conclusion and Future Work

In this work, we propose a novel approach to generate high-readability synthetic text, ensuring differential privacy while maintaining semantic similarity with text in the private dataset. Experimental results demonstrate that using synthetic text as demonstrations for in-context learning incurs only marginal losses in predictive performance compared to using private text. Besides, our synthetic text are also capable of resisting membership inference attacks from malicious users. While it is convenient to invert from embeddings to text, longer text often leads to a higher loss of information within the embeddings, consequently decreasing the quality of synthetic text. In future work, we will explore how to apply our proposed framework to the situation of privacy protection on long text.

Acknowledgments

This work is partially supported by by NSFC (61732006) and Public Computing Cloud, Renmin University of China.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[Anil *et al.*, 2021] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021.

[Bansal *et al.*, 2019] Trapit Bansal, Rishikesh Jha, and Andrew McCallum. Learning to few-shot learn across diverse natural language classification tasks. *arXiv preprint arXiv:1911.03863*, 2019.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[Carlini *et al.*, 2021] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[Chen *et al.*, 2023] Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5747–5758, 2023.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[Du *et al.*, 2023b] Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2665–2679, 2023.

[Duan *et al.*, 2023] Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *arXiv preprint arXiv:2305.15594*, 2023.

[Durfee and Rogers, 2019] David Durfee and Ryan M Rogers. Practical differentially private top-k selection with pay-what-you-get composition. *Advances in Neural Information Processing Systems*, 32, 2019.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[Dwork *et al.*, 2010] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[Gonen *et al.*, 2022] Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022.

[Li *et al.*, 2023] Haoran Li, Mingshi Xu, and Yangqiu Song. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010*, 2023.

[Lin *et al.*, 2023] Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model apis 1: Images. *arXiv preprint arXiv:2305.15560*, 2023.

[McSherry and Talwar, 2007] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[Meehan *et al.*, 2022] Casey Meehan, Khalil Mrini, and Kamalika Chaudhuri. Sentence-level privacy for document embeddings. *arXiv preprint arXiv:2205.04605*, 2022.

[Min *et al.*, 2022] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[Mitchell and Taylor, 1999] Melanie Mitchell and Charles E Taylor. Evolutionary computation: an overview. *Annual Review of Ecology and Systematics*, 30(1):593–616, 1999.

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

[Narayanan and Shmatikov, 2008] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[Ni *et al.*, 2021] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*, 2021.

[Pan *et al.*, 2020] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE, 2020.

[Papernot *et al.*, 2016] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[Pilán *et al.*, 2022] Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101, 2022.

[Sampson, 1976] Jeffrey R Sampson. Adaptation in natural and artificial systems (john h. holland), 1976.

[Santambrogio, 2015] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[Shin *et al.*, 2022] Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*, 2022.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[Song and Raghunathan, 2020] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390, 2020.

[Tang *et al.*, 2023] Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765*, 2023.

[Tukey, 1975] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

[Voorhees and Tice, 2000] Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, 2000.

[Wang *et al.*, 2020] Xiaofei Wang, Tong Han, and Hui Zhao. An estimation of distribution algorithm with multi-leader search. *IEEE Access*, 8:37383–37405, 2020.

[Welleck *et al.*, 2022] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. *arXiv preprint arXiv:2211.00053*, 2022.

[Wu *et al.*, 2023] Tong Wu, Ashwinee Panda, Jiachen T Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. *arXiv e-prints*, pages arXiv–2305, 2023.

[Yue *et al.*, 2021] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221*, 2021.

[Yue *et al.*, 2022] Xiang Yue, Huseyin A Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Synthetic text generation with differential privacy: A simple and practical recipe. *arXiv preprint arXiv:2210.14348*, 2022.

[Zhang *et al.*, 2015] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.