DIFFERENTIALLY PRIVATE CONDITIONAL TEXT GENERATION FOR SYNTHETIC DATA PRODUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Companies have faced increasing pressure in recent years to anonymize user collected data when sharing internally or to third parties. Text data in particular contains copious amounts of personally identifiable information that has proven to be difficult to de-identify and remain useful for the party of interest. Previous works have suggested that synthetic text generation could provide a promising avenue to curate high performant and private datasets. In this paper, we introduce an approach to synthesize high utility text classification datasets by performing conditional generation through a large language model, GPT-2, while providing measurable guarantees via differential privacy. We show that naive approaches suffer heavily from utility loss and may entangle crucial factors in the transformer embedding space, preventing meaningful generation. We analyze how incorporating a secondary learning objective can alleviate this issue, improving utility of the generated data. Ultimately, we show it is possible to create useful synthetic data with strong privacy guarantees.

1 INTRODUCTION

In recent years, language models have seen dramatic improvements in performance over NLP tasks. In large part, this has been due to the rapid accumulation of user generated text on the internet. Companies have been able to aggregate millions of documents available online as well as their user data to train these large language models. However, lawmakers and their constituents have grown wary of data collection and usage practices, urging more stringent regulations.

In 2018, the European Union set the General Data Protection Regulation (GDPR) into motion, one of the first large actions to address the issue of data misuse (Voigt & Bussche, 2017). California followed suit in the same year, releasing their own data regulations (Bukaty, 2019). The goal of these laws are to increase transparency about what kind of information is being collected and to give users more control over how their data is handled. As a consequence of such laws, companies are now increasingly seeking ways to anonymize their data such that they can be used to improve their algorithm performances without putting user data at risk. The GDPR begins with the statement: "The protection of natural persons in relation to the processing of personal data is a fundamental right"; it is imperative that we innovate on methods to use data effectively without risking user privacy. Our paper aims to address two problems companies face: private internal data sharing and model training.

In this paper, we study privatization of unstructured text data. This is the most common form of data available on the internet, and is also some of the most vulnerable to exploitation and user re-identification. Even with safety measures in mind, there has been massive exploitation of user text data. For example, in 2006, as part of their algorithm contest, Netflix released a de-identified dataset of user generated movie reviews. Researchers discovered that surprisingly little information was required to reconstruct the identities of users that contributed to the reviews (Narayanan & Shmatikov, 2006). Further studies have shown how other methods, such as authorship and membership inference attacks (Carlini et al., 2020), can be utilized to reconstruct user identities. All this to say, without proper privacy guarantees and careful data analysis, companies risk user data to exploitation.

In order to provide a formal framework for privacy, Dwork (2006) proposed differential privacy to provide provable and quantifiable guarantees about privacy. Generally, we say that a randomized algorithm satisfies DP if the output distribution is indistinguishable when run on neighboring datasets. Subsequently, Abadi et al. (2016) proposed the differentially private stochastic gradient descent (DP-SGD) learning algorithm for providing similar guarantees while training neural network based models. However, previous studies have shown a drastic trade-off in privacy and utility when training, especially when generating synthetic text data for downstream tasks (Sasada et al., 2021). This has made it impractical for many companies to train and share data while providing strong privacy guarantees.

Current approaches to anonymize textual data predominantly focus on word or sentence level perturbations in order to reduce the ability of authorship or membership inference attacks to recover user identities. However, these approaches do not solve our goals of creating coherent, private, and high-utility text data for the purpose of sharing information. Other approaches seek to generate data synthetically, such as what was proposed in Libbi et al. (2021). However, while their approach is able to synthesize useful textual data, they do so without training language models with differential privacy, urging future research to recreate their results while retaining formal guarantees of privacy. By synthesizing data with DP guarantees, we hope that data science and development teams at companies can use this data to work on shared tasks where real data cannot be shared because of privacy concerns. Moreover, even within the bounds of a research institution, real data may have certain access restrictions. Using synthetic data as a surrogate for the real data can help organizations to comply with privacy regulations. Besides addressing the privacy concerns, synthetic data is an effective way to augment datasets without incurring additional costs. Previous methods such as Anaby-Tavor et al. (2019) have shown promising results in this area. The promise of synthetic data is that it avoids the problem of anonymizing an individual's data by instead producing information about non-existent persons.

In this paper, we present an approach of generating synthetic text data by performing controllable generation through large language models, namely GPT-2. Previous methods in synthetic text generation have been unsuccessful in creating high-utility while retaining end-to-end DP guarantees; we analyze the embedding space of models trained with DP-SGD and find that conditionally prompted language models trained this way suffer from entanglement which reduces its ability to produce high-utility text. By disentangling these embeddings by simply adding a secondary learning objective during training, we show that private, high-utility synthetic text generation is possible. With this approach, we hope that companies and organizations will be able to share data and train high quality models without putting their users data at risk. Our contributions are as follows:

1. We present findings on the difficulty of performing conditional finetuning of large language models with DP-SGD. We show that this process produces embeddings that cause entanglement of important factors, particularly with the labels of a classification dataset. This has the consequence of increasing the difficulty of generating synthetic data that reproduces the distribution of the original. We investigate this issue, proposing a solution and analysis of the phenomena.
2. We generate synthetic datasets for the SST-2 and AG News datasets by performing conditional text generation over GPT-2. We analyze the privacy and utility metrics of the new data at varying settings of differential privacy (epsilon values), showing state of the art privacy-utility loss.

2 Background

2.1 Language Modeling

Given a sequence of tokens X = x0 , · · · , xn , language models (LMs) are trained to compute the unconditional probability of the sequence p(X). This probability can be rewritten in terms of product of conditional probabilities by recursively applying the chain-rule (Bengio et al., 2003) as:

$$p(X) = \prod_{i=1}^{N} p(x_i|x_0, ..., x_{i-1}) \tag{1}$$

This allows modeling the language via next-word prediction. We use the transformer architecture (Vaswani et al., 2017) to model the distribution of natural language. Generation of a new sequence y can be created by sequentially sampling its constituents: $p_\theta(y_0), p_\theta(y_1|y_0), ..., p_\theta(y_m|y_{<m})$.

2.2 Conditional Text Generation

Conditional generation of text attempts to steer the output of a large language model given a desired condition or control variable. A traditional method, originally introduced by Keskar et al. (2019), to accomplish this goal is by performing conditional fine-tuning of the language model over a dataset, such that the desired condition is prepended to the text body: "[condition] SEP text" (where SEP is a special token used to separate between the condition and text).

On the other hand, plug and play controllable language generation (Dathathri et al., 2019) is a method that attempts to keep the language model frozen and manipulate its output by using potential functions such as a discriminator. With a probabilistic function such as this, one can perform gradient updates over the hidden states of the model in order to optimize for the desired control (measured via the discriminator) as well as fluency (measured via the language modeling head distribution).

In this paper, we use a combination of the two approaches in order to generate high-quality data. We first fine-tune a large language model over the desired dataset with conditional prompting similar to Keskar et al. (2019) and then use a gradient-based approach as described by Dathathri et al. (2019) to guide generation.

2.3 Differential Privacy

Differential Privacy (DP) is a formal definition of privacy which offers strong assurances against various re-identification and re-construction attacks (Dwork, 2006; Dwork & Roth, 2013). In recent years, DP has attracted significant attention due to its mathematically sound and provable privacy guarantees and ability to quantify privacy loss, as well as unique properties such as robustness to auxiliary information and post-processing, composability enabling modular design, and group privacy (Dwork & Roth, 2013; Abadi et al., 2016).

Definition 1. *(Differential Privacy Dwork et al. (2006)) A randomized function $\mathcal{M}$ provides (ϵ, δ)-differential privacy if for all adjacent datasets $X, X' \in \mathcal{X}$ and all $Y \subset \mathcal{Y}$,*

$$Pr[\mathcal{M}(X) \in Y] \leq \exp(\epsilon) \cdot Pr[\mathcal{M}[X'] \in Y] + \delta \tag{2}$$

This is a standard definition of DP, implying that the outputs of differentially private algorithms for datasets that vary by a single individual are indistinguishable, bounded by the privacy parameter ϵ. Here, ϵ is a non-negative number otherwise known as the privacy budget. Smaller ϵ values more rigorously enforce privacy, but can often decrease data utility. Unlike other heuristic privacy notions (Huang et al., 2020), DP allows for tracking privacy loss throughout the execution of a program by computing its leakage parameters.

Composability and robustness to post-processing are important properties of DP that are necessary for the guarantees in our paper. Composability allows for reasoning about overall privacy loss from the composition of multiple DP algorithms releasing multiple statistics about a particular dataset. Robustness to post-processing implies that if some mechanism $\mathcal{M}$ satisfies ϵ-differential privacy, then for any deterministic or randomized function $\mathcal{F}$, so does $\mathcal{F}(\mathcal{M})$

Currently, the widespread accessibility of data has increased data protection and privacy regulations, leading to a surge of research into applied scenarios for differential privacy. We use the DP-SGD (Abadi et al., 2016) algorithm to provide guarantees over our fine-tuning process.

DP-SGD Differentially Private Stochastic Gradient Descent (DP-SGD), proposed by Abadi et al. (2016), is one of the first studies to make the Stochastic Gradient Descent computation differentially private. Intuitively, DPSGD minimizes its loss function while preserving differential privacy by clipping the gradient in the optimization's l_2 norm to reduce the model's sensitivity, and adding noise to protect privacy by mitigating the impact of out-of-distribution samples on the model.

3 RELATED WORKS

Current methods on text privitization fall into three general categories: word/sentence level perturbations, private text embeddings, and synthetically generated text. Here, we discuss the advantages and disadvantages of each method.

Word/Sentence Level Perturbations: Many works have discussed anonymizing text by perturbing word or sentence level embeddings to satisfy ϵ-differential privacy. This set of approaches change individual words in a document, often following a variant of metric based DP (Alvim et al., 2018) which has shown to be a more utilitarian perspective of privacy in the context of NLP. However, as discussed by Mattern et al. (2022), these perturbations struggle to provide overall privacy guarantees in the context of language peculiarities and leakage of other personally identifiable information (PII) that allow for re-identification. They also suffer from immense utility losses as grammatical errors and syntactic structure are degraded. Other methods suggested by Weggenmann & Kerschbaum (2018) and Bo et al. (2019) investigate differentially private mechanisms via latent space perturbations and adversarial training, respectively, to reduce authorship inference attacks. However, these methods, again, do not address the issue of PII leakage and suffer from significant uility losses.

Private Text Embeddings: Other methods have investigated the method of releasing private text embeddings instead of releasing the original text content. Recent work such as Lyu and Xu et al. (2021) propose randomization mechanisms that can transform text embedding vectors into one that satisfies metric space differential privacy guarantees. This method has shown promise in providing formal guarantees while also retaining high utility. However, in this paper, we're concerned with problems companies face in the use of internal data sharing that often require human readable text; we examine our approach independent of this body of work.

Synthetic Text: Other methods, especially in the domain of medical and clinical text analysis have attempted to address the issue of privacy through synthetic text generation. This method offers unique privacy issues involving reidentification of individuals via PII leakage. Synthetic data addresses this concern by simply not describing real people, and thus retaining a sense of plausible deniability over the data produced. However, recent methods such as Libbi et al. (2021) which employs a similar method to the one we describe provide no formal differentially private guarantees, suggesting that the utility loss when incorporating a DP training algorithm incurs too high of a utility loss. [**temp:** Another work, Sasada et. al, 2021, discusses how text preprocessing can improve differentially private text generation by creating "multiple duplicates by generalizing named entities" in the dataset. This feels like some DP guarantees are lost by creating duplicates? Even if you generalize named entities this now relies on the accuracy of your NER?] In this paper, we examine how differentially private language models can perform high quality text generation.

4 DATASETS AND PREPROCESSING

We generate artificial datasets for text classification. We choose this task because it allows us to best compare utility and privacy in one dataset. We experiment over two datasets. Each dataset is split 80:20 for training and evaluation.

4.1 SST-2

The SST-2 corpus was curated as a binary classification task, consisting of 11,855 single sentences extracted from movie reviews. Each were labeled as positive or negative sentiment by human annotators. This dataset was perfectly balanced with each sentiment having equal representation (Socher et al., 2013).

4.2 AG NEWS

The AG News corpus is a topic classification task. News articles have been gathered from more than 2000 news sources by ComeToMyHead in more than 1 year of activity. This dataset consists of over 120,000 samples, each labeled under a topic from: Sports, World, Business, Sci/Tech. This dataset was perfectly balanced with each topic having equal representation (Zhang et al., 2015).

5 EXPERIMENTS

This paper provides a means for generating high-utility synthetic text data with privacy guarantees. Current privacy approaches don't fit company needs for text data sharing because they incur too great of utility losses when trained with full, end-to-end DP algorithms. As a consequence, many companies aren't able to reasonably utilize DP text datasets. With this approach, we show that for large language models, we can train with DP, and disentangle embeddings based on particular non-sensitive contextual information we need present. This will significantly boost utility of the synthetic data while retaining sound privacy guarantees.

At a high level, our algorithm is straightforward:

1. Fine-tune a large language model over a desired dataset, performing gradient updates using DP-SGD.
2. Sample from the language model to generate the artificial dataset.

All our experiments were run on one NVIDIA V100 GPU instance.

5.1 FINE-TUNING

The baseline language model that we use for training is a pretrained distilgpt2. Distilgpt2 is a smaller version of GPT2 that was trained using knowledge distillation to be faster and lighter while retaining original performance Sanh et al. (2019). We use this model over the larger versions to provide faster iteration of training under different configurations. Moreover, preliminary experiments using the larger GPT-2 model showed that there were no significant differences in performance between the two. The distilgpt2 model is obtained via HuggingFace's transformers library.

We fine-tune the language model $\mathcal{G}$ to the task of synthesizing labeled sentences, to obtain the fine-tuned language model $\mathcal{G}_{tuned}$. Here, $\mathcal{G}$ is specifically fine-tuned to the linguistic domain of D_{train} (that is, the sentences, vocabulary, style, etc.), as well as the particular classes in D_{train}. Generally speaking, we would like to use $\mathcal{G}_{tuned}$ to generate a sentence set of any size, and each sentence labeled with a class.

In our case, G is the neural model of GPT-2. We fine-tune GPT-2 by training it with the data in $D_{train} = \{(x_i, y_i)\}_{i=1}^{n}$. We generate training samples for conditional finetuning by prepending the label with the text body so that we end up with: $U =$ BOS y_i SEP x_i where BOS and SEP are special tokens to indicate the beginning of the sentence and to separate label from text body, respectively. We fine-tune this model under different privacy settings by changing the epsilon parameter. When training with DP, we substituted the Adam optimizer with the DP-SGD based optimizer from the private-transformers library, provided by Li et al. (2021). We also use the ghost-clipping mechanism outlined by this paper which introduces a memory efficient method to perform per-example gradient clipping during training. We utilized Renyi differential privacy (Mironov, 2017) for accounting purposes.

5.1.1 NAIVE CONDITIONAL FINE-TUNING

Our first model is a naive baseline that uses the processed labeled text data. Each layer in the pre-trained GPT-2 model embed text at different abstraction levels. For transfer learning, the key is to take advantage of the previously learned information that is relevant for the current task, but adjust representations such that they are suitable for the new language and domain-specific terminology. We investigate how differential learning rates along with layer freezing impacts the quality of generation. The various training setups and results we obtained from these experiments are available in Appendix A (**todo**).

In this approach, we (1) perform full fine-tuning of $\mathcal{G}$ with the training procedure described above to produce $\mathcal{G}_{tuned}$. As discussed above, our generation procedure is similar to the PPLM approach, and requires training a discriminator. So, (2) we freeze the weights of of $\mathcal{G}_{tuned}$ and feed the transformer embedding of the last token in the sequence to a discriminator head. (3) We generate labeled text samples by performing nucleus sampling with parameters $p = 0.95$ and then updating the hidden states through gradient updates from the discriminator head by computing the loss of the generated

sample with respect to the desired label. (4) To evaluate the utility of the generated dataset, we train a BERT classifier to predict labels, and compare the performance to the held-out test set.

We noticed that while this method worked well and produced high-quality results for the non-differentially private setting, as other studies have shown in the past, this method degraded after training under DP-SGD even at extremely high epsilon settings ($\epsilon = 256$). By qualitatively analyzing the generated samples produced by the model, we observed that training with DP-SGD caused $\mathcal{G}_{tuned}$ to "forget" how to conditionally generate text. Without the model being able to separate classes implicitly, it becomes impossible to generate data that has any utility.

For example, when the non-private model was prompted to generate a positive sentiment sample for the SST-2 dataset, "BOS positive SEP", it would generally produce a positive sentiment sequence (e.g. "[**todo**: non-DP example]"). However, the same approach with the $\mathcal{G}_{private}$ model produced text that fit neutral or even negative sentiments (e.g. "[**todo**: DP example]). We observe that naively training the GPT-2 model this way produces embeddings that entangle the sentiment classification of the text. This makes it impossible to generate an artificial dataset that retains the distribution of labels since we cannot precisely control what label the generated text will fit into. We further confirm these findings by analyzing the embedding space through projections and perform classification over the embeddings via decision trees. These results are discussed in depth in Section 6.

5.2 Multitask Conditional Fine-tuning

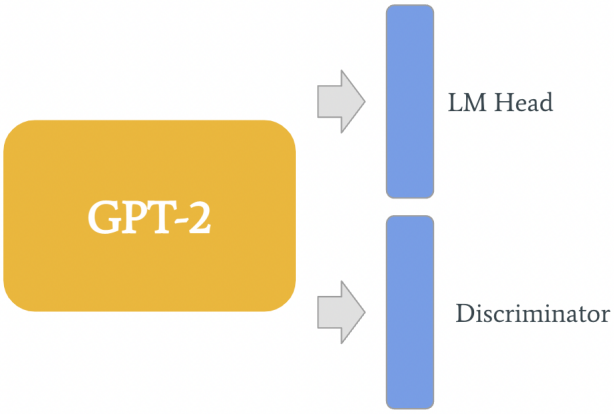

Figure 1: [**todo:** include better diagram]

In order to address this issue we perform multitask learning during fine-tuning. In the naive approach, the transformer architecture is attached to a linear language modeling head that models the probability distribution of the next word. In this new approach, we simultaneously train a discriminator head to the transformer architecture, as shown in the diagram above. Now, we perform two gradient updates at every step – one to update the language modeling head and the other to update the discriminator head.

Since we still want to retain conditional prompting for the model, we want the language model to be able to see the conditional prompt, i.e. "BOS positive SEP text", which includes the prepended label so that the model is able to understand prompting. Meanwhile, the discriminator head should be able to learn to classify labels without seeing the answer in the input. So, for the language head, we feed the label prompted text data and perform a gradient update. Then, for the discriminator head, we replace the label text data with a random token, the intuition being that the discriminator head will give less attention to the embeddings related to early in the sequence.

We also train this discriminator head to classify text at different prefix lengths. For example, if the prefix step was specified to be 2, we would compute the loss given the transformer output for the second token, fourth token, sixth token, and so on. The loss is linearly weighted such that the first

Algorithm 1 DP Multitask Conditional Training **todo: make this easier to read**

Data: $\mathcal{G}_{\text{pretrained}}, D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, number of iterations T, learning rates η_{lm}, η_{discrim}, noise σ, clipping bound C, initial parameter vectors $\theta^{(0)}_{\text{transf}}$, $\theta^{(0)}_{\text{lm}}$, $\theta^{(0)}_{\text{discrim}}$
, sampling probability q, batch size B

for $k \in [K]$ **do**

  Draw batch b_t from $\mathcal{D}$ with sampling probability q.

  for $(x_i, y_i) \in b_t$ **do**

    rand $\leftarrow$ random token from vocabulary

    $s_{\text{lm}} \leftarrow$ "BOS y_i SEP x_i"

    $s_{\text{discrim}} \leftarrow$ "BOS rand SEP x_i

    $g^{(i)}_{\text{lm}} \leftarrow \nabla\mathcal{L}(G_{\theta^{(k)}_{\text{transf, lm}}}(s_{\text{lm}}), s_{\text{lm}})$

    $g^{(i)}_{\text{discrim}} \leftarrow \nabla\mathcal{L}(G_{\theta^{(k)}_{\text{transf, discrim}}}(s_{\text{discrim}}), y_i)$

    $g^{(i)}_{\text{lm}} \leftarrow \min(1, C/||g_i||_2)$

    $g^{(i)}_{\text{discrim}} \leftarrow \min(1, C/||g_i||_2)$

  end

  $g_{\text{lm}} \leftarrow \frac{1}{B}\sum_{i \in b_t} g^{(i)}_{\text{lm}} + N(0, \sigma^2 C^2 I)$

  $g_{\text{discrim}} \leftarrow \frac{1}{B}\sum_{i \in b_t} g^{(i)}_{\text{discrim}} + N(0, \sigma^2 C^2 I)$

  $\theta^{(k+1)}_{\text{transf, lm}} \leftarrow \theta^{(k)}_{\text{transf, lm}} - \eta_{\text{lm}} \cdot g_{\text{lm}}$

  $\theta^{(k+1)}_{\text{transf, discrim}} \leftarrow \theta^{(k)}_{\text{transf, discrim}} - \eta_{\text{discrim}} \cdot g_{\text{discrim}}$

end

Output: *Trained Model* $\theta^{(K)}_{transf}, \theta^{(K)}_{lm}, \theta^{(K)}_{discrim}$

prefix is weighted the least and the last prefix is weighted the most. Lastly, this loss is averaged, and then the gradient update is computed. [**todo:** include diagram or example to showcase this].

This loss procedure is to ensure the discriminator head is robust enough to provide meaningful classifications at different parts of a sequence to improve its contribution during gradient based generation. We found that this improves robustness of generation. For example, the following two examples illustrate the discriminator's classification scores where the first model was trained using the prefix losses and the second model was trained without. [**todo:** include example for generation w and w/o prefix loss]

Ultimately, we find that by training both the discriminator and language modeling head simultaneously, we are able to disentangle the embeddings with respect to the classification labels. In Figure 2 and Table 1 [**todo:** include random forest results to show inseparability], we show how this approach impacts the embedding space of models trained at different differential privacy settings compared to the naive approach via a UMAP projection. We observe that this approach makes it possible to perform conditional text generation with differential privacy. We find that the noise injected via differential privacy doesn't prioritize the model to implicitly learn important distributional factors about the original dataset such as the prompted classification labels, and needs an explicit loss mechanism to do this.

5.3 Generation

Next, we perform conditional generation to produce a private version of the dataset. We wish to retain the same distribution of classes in the dataset as the original. In order to guide generation towards a particular class, we apply a PPLM based gradient approach. We utilize the discriminator trained in the previous step to perform gradient updates over the hidden states of the model to guide the generation towards this class. The steps for generation of a single sample are as follows:

1. Prompt the model with **BOS class SEP** and generate the distribution of the next word via the language modeling head.

2. Use the hidden states of the language model to generate the classifier distribution and compute its loss with the desired classifier. Update the hidden states of the language model by performing gradient udpates as described by PPLM.
3. Sample from the new language modeling head distribution for the next word with top-p and top-k filtering.
4. Repeat steps 1-3 until the end of sentence token is reached or the specified maximum length is reached.

We discuss further implications and limitations of this approach in Section 7..

6 EVALUATION

With the described approach, we generate synthetic versions of the SST-2 and AG news dataset. 5 variations are generated with different differential privacy settings: $\epsilon \in \{256, 16, 8, 3\}$ and a non-private version.

6.1 DISENTANGLED EMBEDDINGS

Here, we present our findings on the entangled embeddings when training GPT-2 with DP-SGD. In Figure 1, we plot the UMAP embeddings from the baseline training procedure where we perform standard conditional fine-tuning over the SST-2 dataset. We observe that when trained without differential privacy, the model implicitly learns separation between the labeled classes. However, even at an extremely high and meaningless privacy setting at $\epsilon = 256$, this implicit learning is degraded. We hypothesize that this contributes to the poor conditional generation ability of the naive model. By creating an explicit separation of classes by training the discriminator head (still using DP-SGD), the model embeddings are clearly separable as seen in Figure 2, even at meaningful privacy settings at $\epsilon = 3$. Moreover, we analyze how this embedding space changes with respect to each layer in the language model. These findings can be seen in Appendix B.

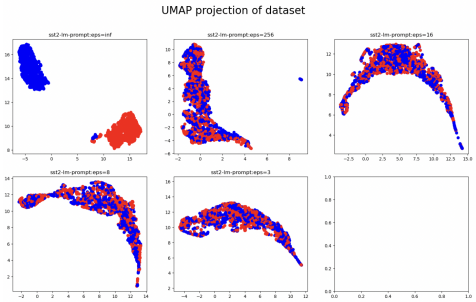

Figure 2: UMAP projection of SST-2 test set on language model trained with baseline training algorithm. Blue is positive sentiment and red is negative sentiment.

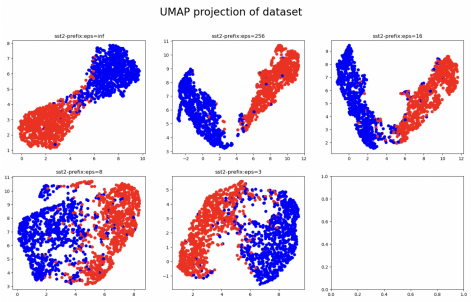

Figure 3: UMAP projection of SST-2 test set on language model trained with multitask training algorithm. Blue is positive sentiment and red is negative sentiment.

6.2 PRIVACY

Differentially private training provides formal guarantees about the generated text as a consequence of the post-processing theorem. However, recent works have cited that the impact of epsilon DP on large language model training is unclear, and we could observe empirically high privacy-preservation even at high epsilon levels (Ponomareva et al., 2022). To test this, we test the artificial dataset for memorization by comparing the proportion of n-grams (for $n \in [3...7]$) in the synthesized data to those present in the original dataset.

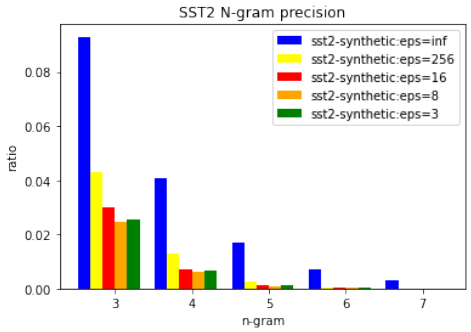

Figure 4: N-gram ratios of the different synthetic datasets trained on SST-2

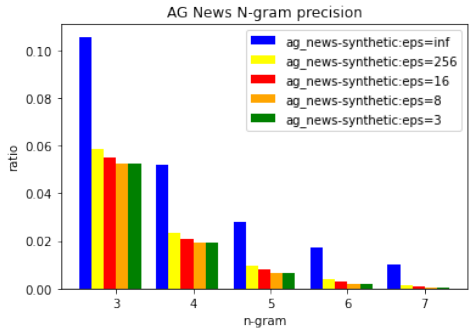

Figure 5: N-gram ratios of the different synthetic datasets trained on AG News.

Table 1: Classification Accuracies of Naive and Multitask Model

Dataset Variant	**SST-2 (Naive)**	**SST-2 (Multitask)**	**AG News (Naive)**	**AG News (Multitask)**
Original	0.94	0.94	0.93	0.93
Synthetic Non-Private	0.86	0.89	0.88	0.913
Synthetic $\epsilon = 256$	0.53	0.88	0.25	0.883
Synthetic $\epsilon = 16$	0.5	0.86	0.25	0.874
Synthetic $\epsilon = 8$	0.5	0.82	0.25	0.871
Synthetic $\epsilon = 3$	0.5	0.80	0.25	0.867

6.3 Utility

We measure the utility of the synthetic dataset by training a classifier over the synthesized data and evaluate the performance on the held-out test dataset. We compare the performance of this classifier to one trained over the original dataset. We are only concerned with utility over state of the art classifiers, so we choose using DistilBERTForSequenceClassification as provided by the HuggingFace Transformers library for this measurement.

7 Discussion

We believe this method has potential for creating private and relevant artificial datasets that will enable companies to share information without putting users' personal information at risk. However, we want to point out some limitations and future directions for this line of work. Firstly, previous studies have shown that training neural network models with DP-SGD can result in increased bias. In our work, we chose to use perfectly balanced datasets in order to mitigate the problems of unequal representation of classes. This could potentially lead to fairness issues when generating synthetic data, and biases from the original data may be amplified in the new dataset (Kuppam et al., 2019; Zhu et al., 2020). Future work may investigate how using this method affects fairness, and the implications of using of a discriminator to explicitly separate embeddings.

8 Ethics Statement

References

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. ACM, oct 2016. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145%2F2976749.2978318.

Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. Metric-based local differential privacy for statistical applications. *CoRR*, abs/1805.01456, 2018. URL http://arxiv.org/abs/1805.01456.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Not enough data? deep learning to the rescue!, 2019. URL https://arxiv.org/abs/1911.03118.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003. ISSN 1532-4435.

Haohan Bo, Steven H. H. Ding, Benjamin C. M. Fung, and Farkhund Iqbal. ER-AE: differentially-private text generation for authorship anonymization. *CoRR*, abs/1907.08736, 2019. URL http://arxiv.org/abs/1907.08736.

Preston Bukaty. *The California Consumer Privacy Act (CCPA)*. IT Governance Publishing, June 2019. doi: 10.2307/j.ctvjghvnn. URL https://doi.org/10.2307/j.ctvjghvnn.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. *CoRR*, abs/2012.07805, 2020. URL https://arxiv.org/abs/2012.07805.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Cynthia Dwork. Differential privacy: A survey of results. In *Lecture Notes in Computer Science*, pp. 1–19. Springer Berlin Heidelberg, 2006. doi: 10.1007/978-3-540-79228-4_1. URL https://doi.org/10.1007/978-3-540-79228-4_1.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. *CoRR*, abs/2010.02772, 2020. URL https://arxiv.org/abs/2010.02772.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858, 2019. URL http://arxiv.org/abs/1909.05858.

Satya Kuppam, Ryan McKenna, David Pujol, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Fair decision making using privacy-protected data. *CoRR*, abs/1905.12744, 2019. URL http://arxiv.org/abs/1905.12744.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *CoRR*, abs/2110.05679, 2021. URL https://arxiv.org/abs/2110.05679.

Claudia Alessandra Libbi, Jan Trienes, Dolf Trieschnigg, and Christin Seifert. Generating synthetic training data for supervised de-identification of electronic health records. *Future Internet*, 13(5):136, May 2021. doi: 10.3390/fi13050136. URL https://doi.org/10.3390/fi13050136.

Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. The limits of word level differential privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 867–881, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.65. URL https://aclanthology.org/2022.findings-naacl.65.

Ilya Mironov. Renyi differential privacy. *CoRR*, abs/1702.07476, 2017. URL http://arxiv.org/abs/1702.07476.

Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006. URL http://arxiv.org/abs/cs/0610105.

Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. Training text-to-text transformers with privacy guarantees. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2182–2193, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.171. URL https://aclanthology.org/2022.findings-acl.171.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC*$^2 Workshop$, 2019.

Taisho Sasada, Masataka Kawai, Yuzo Taenaka, Doudou Fall, and Youki Kadobayashi. Differentially-private text generation via text preprocessing to reduce utility loss. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, April 2021. doi: 10.1109/icaiic51459.2021.9415242. URL https://doi.org/10.1109/icaiic51459.2021.9415242.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 1st edition, 2017. ISBN 3319579584.

Benjamin Weggenmann and Florian Kerschbaum. Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. *CoRR*, abs/1805.00904, 2018. URL http://arxiv.org/abs/1805.00904.

Nan Xu, Oluwaseyi Feyisetan, Abhinav Aggarwal, Zekun Xu, and Nathanael Teissier. Density-aware differentially private textual perturbations using truncated gumbel noise. *The International FLAIRS Conference Proceedings*, 34(1), April 2021. doi: 10.32473/flairs.v34i1.128463. URL https://doi.org/10.32473/flairs.v34i1.128463.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

Keyu Zhu, Pascal Van Hentenryck, and Ferdinando Fioretto. Bias and variance of post-processing in differential privacy. *CoRR*, abs/2010.04327, 2020. URL https://arxiv.org/abs/2010.04327.

A Appendix

You may include other additional sections here.

A.1 Naive Embeddings

A.2 Multitask Embeddings

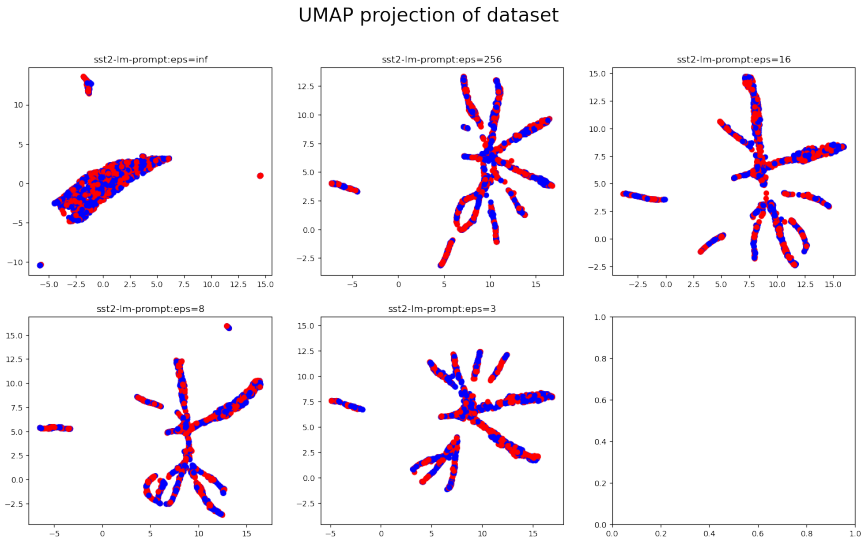

Figure 6: Naive Layer 1

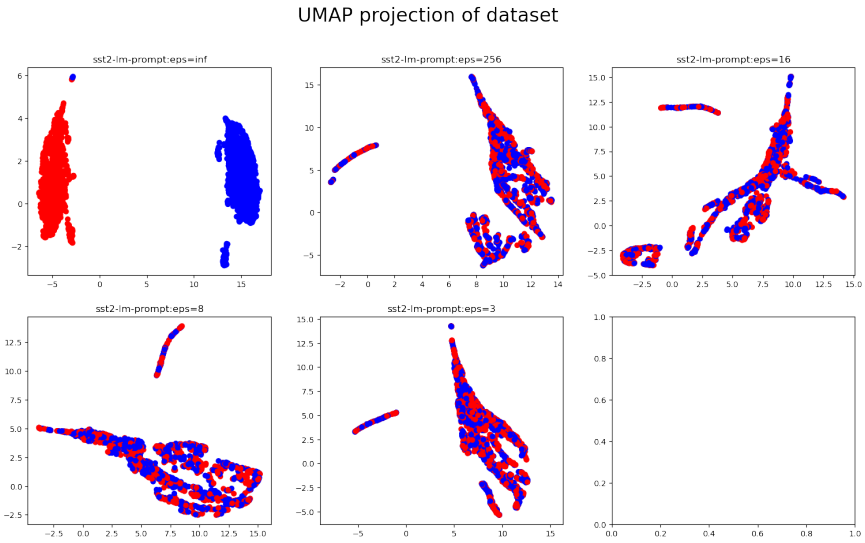

Figure 7: Naive Layer 2

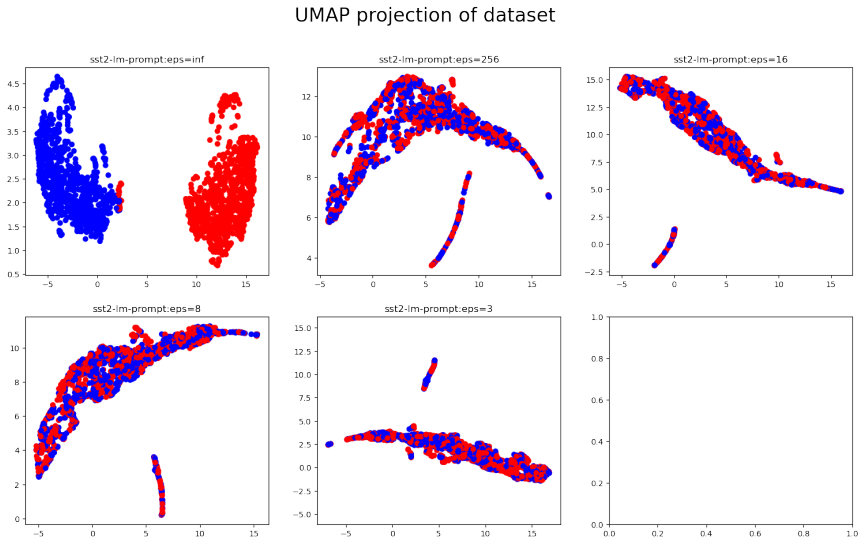

Figure 8: Naive Layer 3

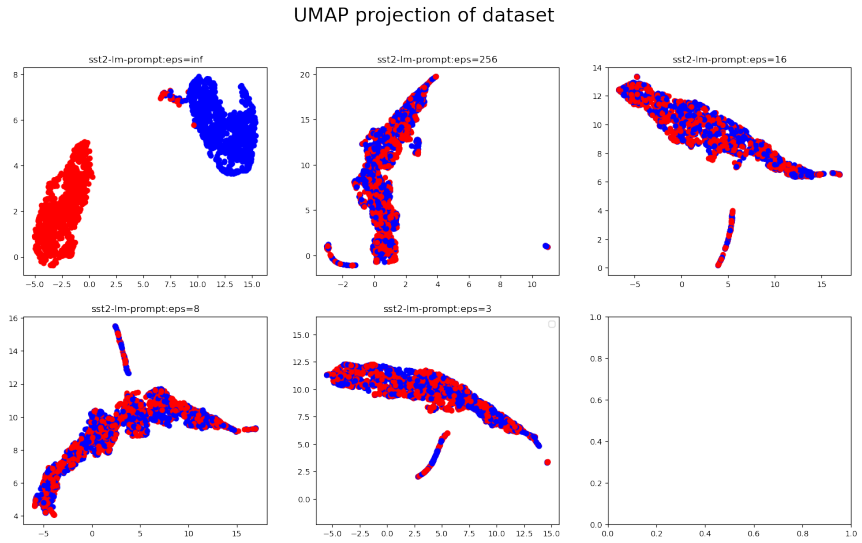

Figure 9: Naive Layer 4

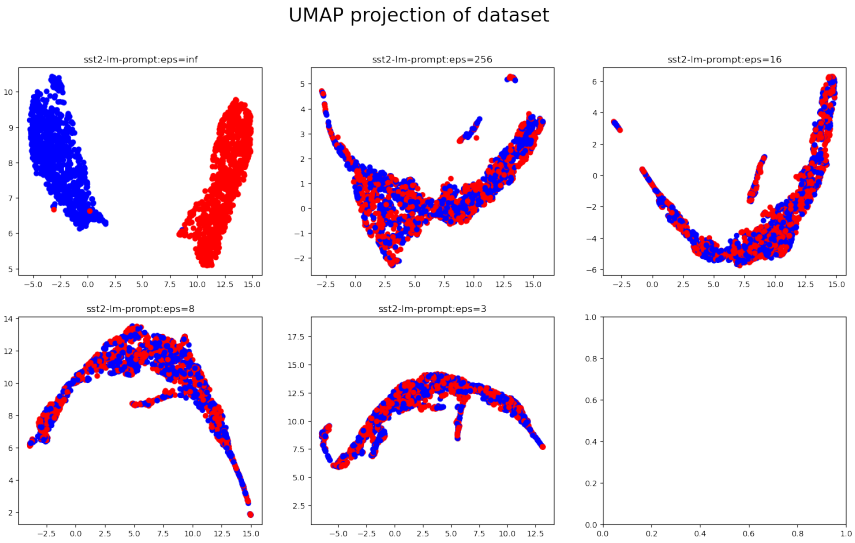

Figure 10: Naive Layer 5

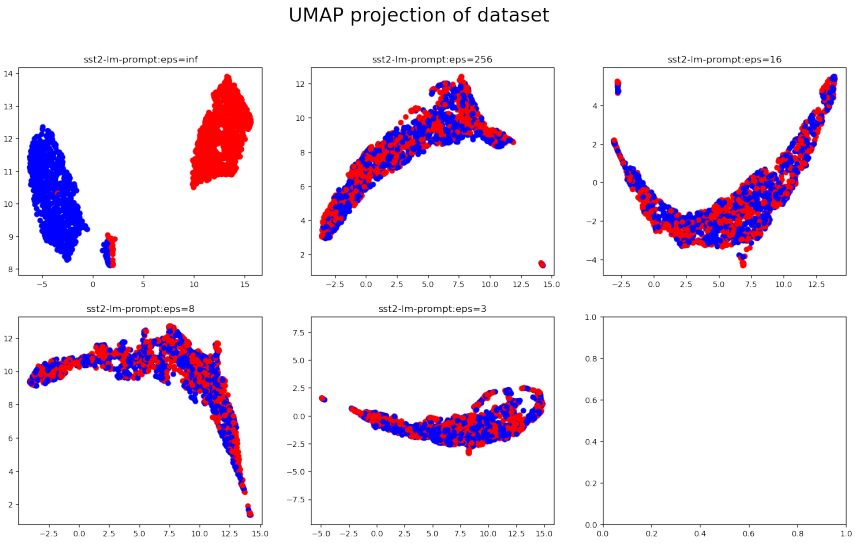

Figure 11: Naive Layer 6

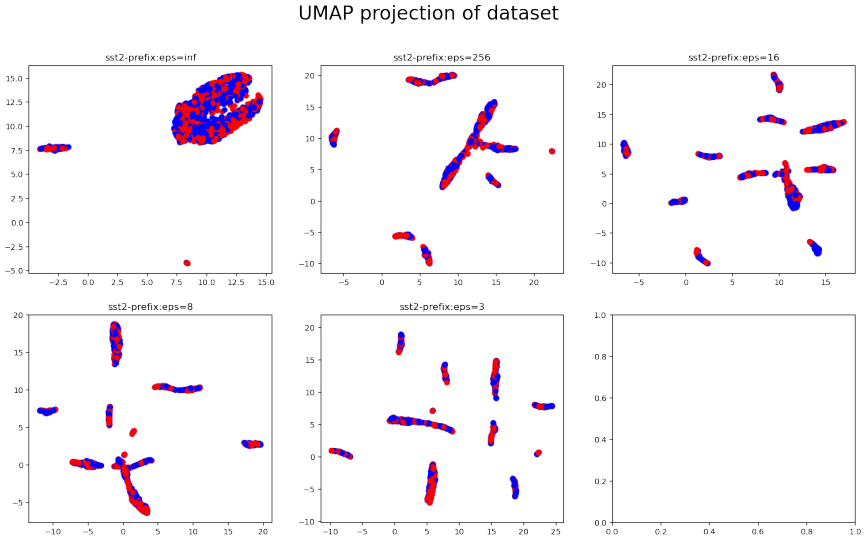

Figure 12: Multitask Layer 1

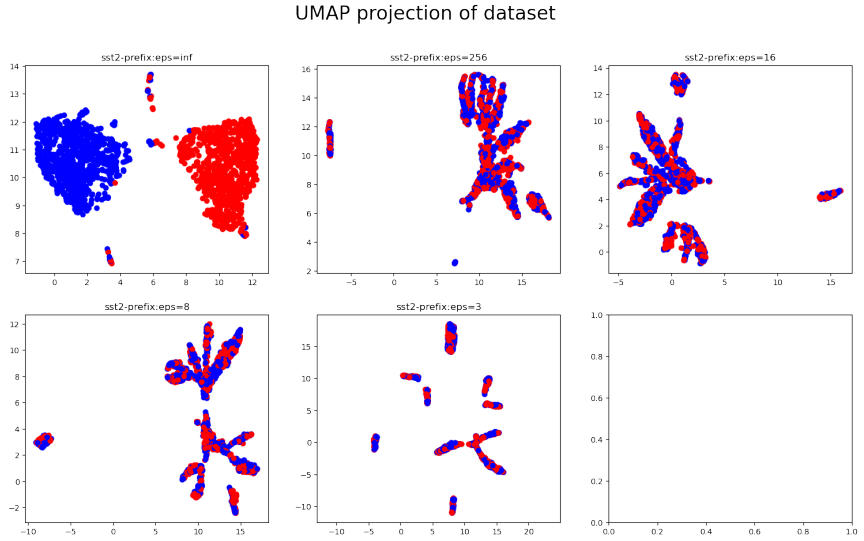

Figure 13: Multitask Layer 2

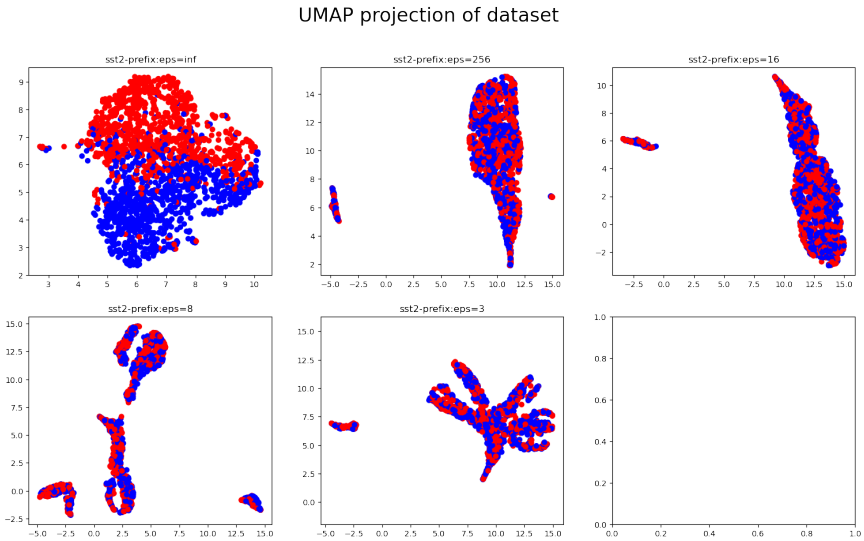

Figure 14: Multitask Layer 3

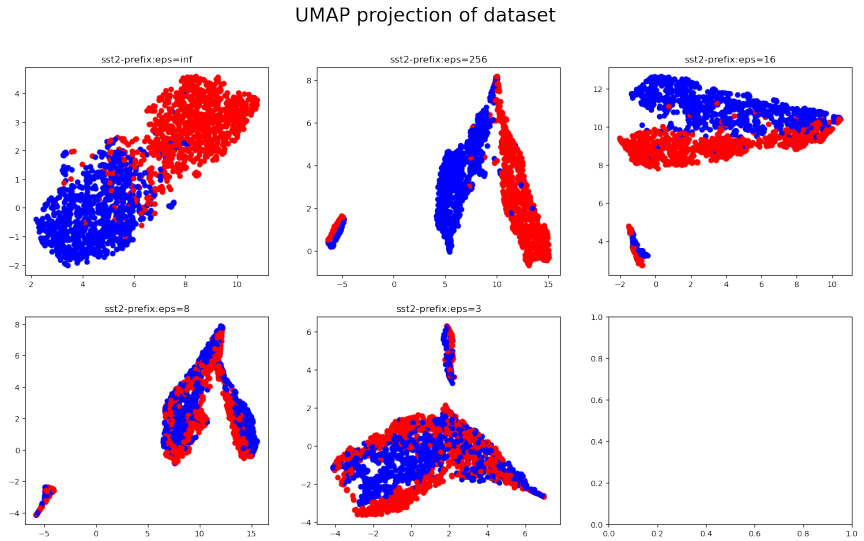

Figure 15: Multitask Layer 4

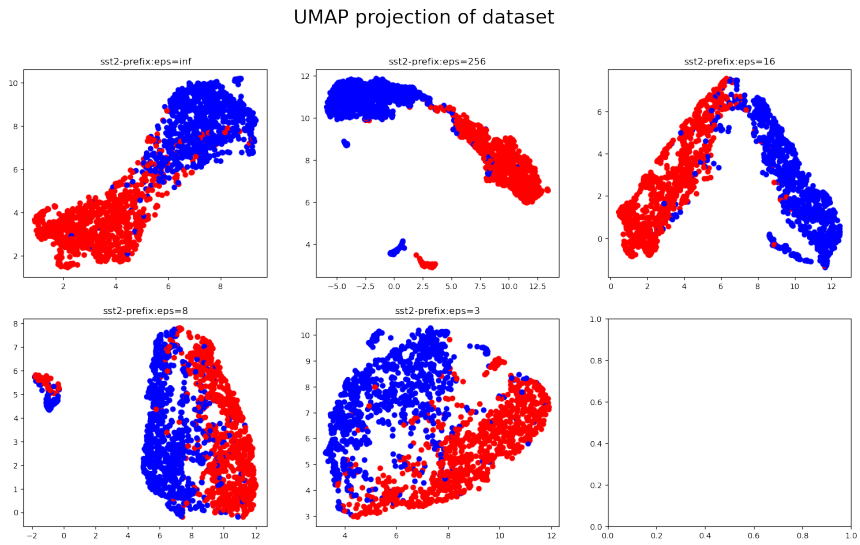

Figure 16: Multitask Layer 5

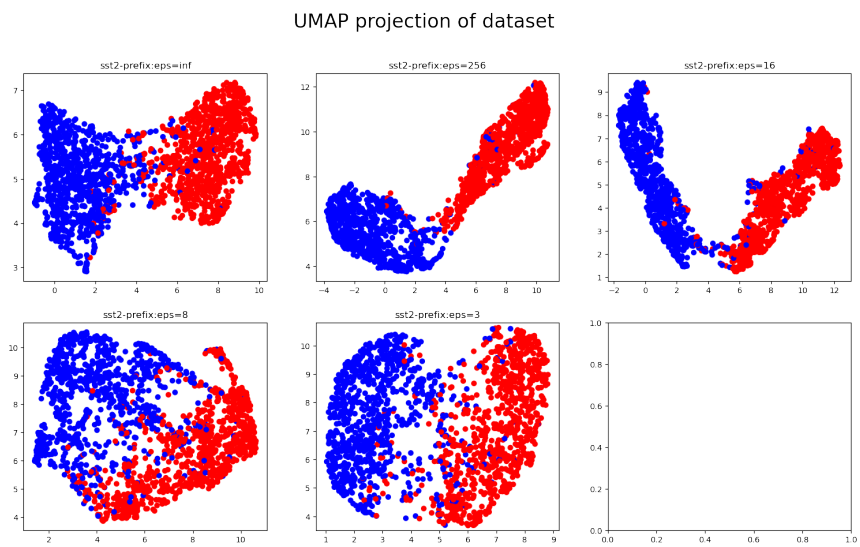

Figure 17: Multitask Layer 6