Data Synthesis for Motion Data

Kevin Zhao
zhaokty@g.ucla.edu

Chinmay Devadhar
cdevadhar@g.ucla.edu

Renuka Bhusari
renukabhusari@g.ucla.edu

Stewart Kwok
stewartkwok@g.ucla.edu

Teong Seng Tan
teongseng@g.ucla.edu

Abstract

Differential privacy is a privacy-preserving technique designed to counter differential attacks. The central problem of our project is that motion data used to train generative models can be recovered by adversaries through such differential attacks. Many differential privacy modules already exist to defend against these differential attacks, but none have been attached to motion diffusion models. Thus, our goal was to protect sensitive motion data used to train models by applying differential privacy techniques to a diffusion model that synthesizes motion data. We achieved this by attaching one specific privacy module, Opacus, to a well-known motion diffusion model, Guy Tevet's MDM. Furthermore, we attempt to deduce the effects of different levels of noise injected on the utility of the dataset. Our results show that it is feasible to attach a privacy module to motion diffusion models, but due to hardware limitations and poor benchmarking, we concluded that further research is needed to better quantify utility of data generated by motion diffusion models trained with noise injected by privacy modules.

Introduction

In the analysis of motion data, whether performing classification of motion or running a predictive model on motion data, differential privacy (DP) is increasingly important because an individual's motion data may be sensitive and be Personally-Identifying Information (PII). An attacker should thus not be able to make any inferences about whether an individual's motion data was involved in the model – a membership inference attack – since this would be a violation of privacy. Existing research in the area focuses on applying differential privacy to image diffusion models through techniques such as differentially-private stochastic gradient descent (DP-SGD), reducing the amount of noise injected while preserving DP, but has not involved applying any of those techniques to diffusion models which synthesize motion data. Our project thus demonstrates how DP techniques can be applied to a motion diffusion model to ensure DP of the model while investigating the effect of the techniques on the quality of output. We used the motion data set

HumanML3D to train the diffusion model, and the DP library Opacus to ensure DP. Subsequently, model utility was evaluated based on model output and loss.

Background

Differential Privacy

A privacy mechanism that has arisen to prominence in recent years in response to differential attacks is differential privacy. Differential attacks are executed when results of two datasets with minimal differences can be analyzed to figure out information about the differences in data. Differential privacy involves ensuring that individual data points cannot be deduced properly through this comparison of differences [2].

The central objective of differential privacy is to render each data point non-discriminatory through such differential attacks while still remaining a robust and useful model. This is generally achieved by means of injecting a certain quantity of noise into the data while training, thereby blurring the data and thus significantly reducing the likelihood of any polynomial-time adversary reversing it, which is a concern for private data such as medical records. Worth noting is that the above definition is akin to the notion of "computational" differential privacy, while other definitions of differential privacy may hold to satisfy hiding from infinitely-powerful adversaries as opposed to simply feasibly efficient adversaries [1].

Definition of an ϵ-indistinguishable Differentially Private Model

Let D be a dataset that contains one person's private data x, and D' be a dataset that does not contain x, and let $M(D)$ be a model trained on the dataset D that produces some output dataset. Then M is ϵ-indistinguishable if

$$P(M(D) = x) \leq e^{\epsilon} \cdot P(M(D') = x) + \delta \quad \forall x \; for\ all\ (D, D')\ pairs$$

[4]. We term ϵ the privacy budget of M [2]. Intuitively, the privacy budget should represent how "un-private" the model is, or in other words, how much of a risk the model poses with respect to revealing an individual's data. Note that a privacy budget of 0 is the perfect scenario since everyone's data is perfectly secure, however, these models are useless since models necessarily require themselves to be based on their input data, and thus no useful model would be able to achieve a privacy budget of 0. Modern definitions of ϵ-indistinguishability commonly introduce the δ term to more strictly define differential privacy, however, $\delta = 0$ is the same notion of differential privacy introduced by Dwork et al.

Diffusion Models

With the emergence in popularity and ease of use of motion sensors, motion data has become a large well of data to analyze. Furthermore, software has evolved such that common tasks can require simulation of human motion. Generation of such three-dimensional simulation via text input is possible through use of a recently-introduced class of generative model: the diffusion model. A diffusion model is trained to remove noise from assets (images, etc) to which multiple layers of noise have already been added. Adding this noise is known as the forward process that takes the form of a Markov chain, while the reverse process (removing the noise in multiple small incremental steps) is what the diffusion model is trained to do. Finally, sampling will take a random seed/noise as input to generate a unique output [3].

Related Work

Similar work has been done in the regime of image generation. We refer to dp-promise (Wang et al), which explores a similar concept of embedding differential privacy in a diffusion model. [4] However, Wang et al are more concerned with optimization of this framework to minimize noise while preserving differential privacy. In contrast, our project serves more as a proof of concept, for feasibility purposes, to show that it is possible to attach a differential privacy module to a diffusion model for motion data. Thus, we are less concerned with the same intensive optimizations performed in dp-promise.

Similarly, Mofusion (Dabral et al) is a project focused on reducing the noise in motion diffusion models. [5] This would have also been a good resource to improve a diffusion model's performance. However, due to time and compute constraints, we were only able to show that it is possible to attach a differential privacy module, but we were unable to perform optimizations such as the ones defined by Mofusion.

Threat Model

We define our threat model as follows:

First, we assume that our model is aggregating sensitive data from a large pool of people from whom motion data has been collected. This data is inherently sensitive as it reveals information about their routines and actions in their life. Naturally, the people sending in this data do not want it to be leaked, hence the need for DP.

We assume an attacker of *reasonable* computing power. This generally implies that they are operating under top-of-the-line extant technology, and thus can run polytime algorithms but not superpolynomial ones. We define this attacker thusly as it is most realistic — there is no need to consider more powerful attackers since they cannot possibly exist, and this also helps lower the necessary threshold of differential privacy that our model would need to achieve.

Next, this attacker should be able to send polynomially many queries to our model, and our model will output a result that the attacker can analyze with their given polytime algorithm for membership inference. Attacks that do not involve attempts to exfiltrate information about the training set are not interesting to us for this project. Ideally, the attacker's algorithm is not able to output useful data so that they cannot learn anything about the training data used in our model.

Methodology/Design

The primary objective of our project was to attach a differential privacy module to a motion diffusion model, so that we can train the model with differential privacy and compare how different injected noise levels affected the utility of the generated motion data. We elected to start with MDM (Tevet et al), which was the most well-known motion diffusion model we could find. [6] As for the differential privacy module, we considered several options, including Amazon's fastDP module, but we ultimately settled on Meta's Opacus module.

The first big issue encountered when we tried to attach Opacus to MDM was a layer compatibility issue. MDM used PyTorch's transformer encoder and decoder layers, both of which utilize another PyTorch class "MultiheadAttention" with which Opacus is incompatible. Opacus actually provides a separate class "DPMultiheadAttention" which has the same functionality as the PyTorch class but is compatible with their privacy engine. Ultimately, we got around this issue by making our own copies of the PyTorch transformer layers that instead use this Opacus-provided class, and modifying our copy of MDM to use these custom transformer layers rather than the default PyTorch layers.

After making these modifications, the next challenge we encountered was version incompatibility issues. Several modules in the most recent Opacus version (1.5.2) were dependent upon modules only in PyTorch 2.0 and above, but the most recent version of MDM (last updated in 2022) was only compatible with PyTorch 1.7. Thus, we decided to instead use Opacus 0.11.0, which was also compatible with PyTorch 1.7. It is worth noting that this old Opacus version does not have a Privacy Engine consistent with the most recent Opacus version, so we had to remodify some of our code that we wrote in our MDM training loop to account for this version rollback.

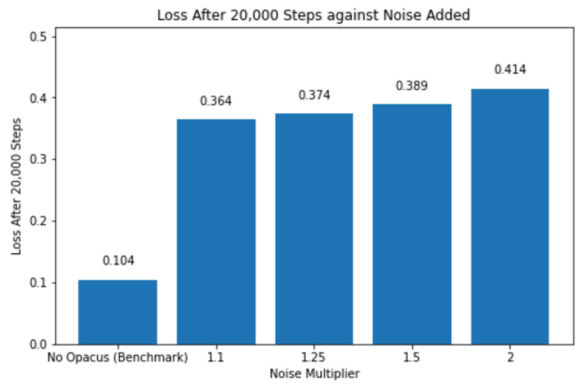
Besides the compatibility issues, we also encountered some dimensionality issues with the Opacus module, so we had to pad our data accordingly.

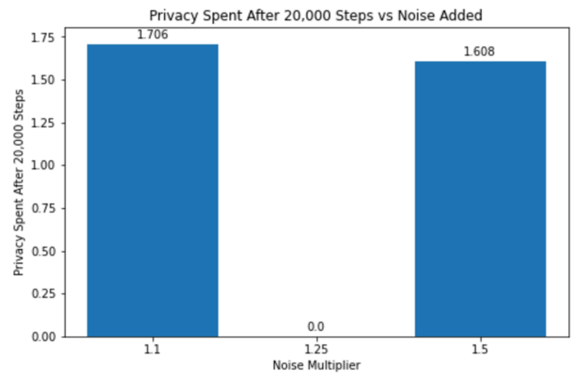
After these adjustments, we were able to successfully attach the Opacus 0.11.0 privacy engine with the most recent version of MDM. Our next objective was to train MDM at different noise levels injected by the privacy engine and compare the resulting utility of the data generated by the modules. For our training, we encountered several hardware issues that will be further discussed in more detail in the subsequent sections. After shrinking the model size and lowering the batch size for our training to account for these hardware issues, we were able to train a few different versions of MDM with varying levels of noise injected using the privacy engine. We then compared the outputs of each model, as discussed in the following results section.

Results

We trained a benchmark motion diffusion model with no differential privacy noise injected, and compared this with four experimental models with different noise multipliers specified from our Opacus

PrivacyEngine. Each model was trained to 20,000 steps with batch size 16 and feed forward size 256 (scaled down from the default motion-diffusion-model settings of batch size 64 and feed forward size 1024 due to GPU constraints). The loss after 20,000 steps for each model is graphed below. There is a clear positive correlation between noise multiplier and model loss, although because adding DP significantly increased training time, and given our GPU constraints we could not increase the number of steps, loss remained quite high (>0.35) for all experimental models. The positive correlation is expected: adding noise naturally deteriorates model performance, which is reflected in the loss. We also graph the privacy spent for 3 of the experimental models – one of the data points was lost due to Kaggle crashing before we could record the result.





Discussion

In general, a privacy budget of between 0 and 1 is considered good, a privacy budget of between 1 and 10 is considered "better than nothing", while a privacy budget of more than 10 is considered bad. [7] Our results thus prove that a "better than nothing" degree of DP is achievable with the Opacus library, but at significant time cost – not shown in the results above is that it took more than 8 hours to train MDM with DP (batch size 16) to 20,000 steps , while it took around 3 hours to train MDM without DP (batch size 16) to 50,000 steps. This is because for every step, Opacus clips all the gradients before adding Gaussian-distributed noise, which is computationally expensive.

It is notable that even with a privacy budget of around 1.7 that is "better than nothing", the probability of the result of two datasets differing by a single data point can differ maximally by a factor of about ~550% since $e^{1.7} \approx 5.5$. However, we should note that the probability of any one result is minuscule, and thus our DP guarantee is simply that as we move across the landscape of probabilities, perturbations are limited to constant scaling, which limits what can be inferred from the probability landscape surrounding two nearby points.

The models described in this paper were not optimally trained due to temporal and computational constraints. We did not have access to NVIDIA GPUs that supported CUDA, which were necessary for all motion data diffusion models. Thus, we had to turn to cloud computing options. However, free tiers were limited in computation power and computation time, so we only have reasonable results for 20,000 steps, which is much fewer than we would have liked. This still took a long time, and we thus weren't able to improve upon this model or collect very many data points. Additionally, due to these computational constraints, we were not able to train our models on a batch size of 64, and instead had to significantly shrink our model's batch size and feed-forward size, which severely degraded model performance. This implies that more privacy would have to be spent in training in order to achieve minimal data quality levels for MDM with DP.

Future work should explore this avenue with superior computation power, training more epochs with a larger batch size and feed forward size. Furthermore, it would be ideal to adopt some of the ideas outlined in Mofusion and dp-promise in order to make the model more performant and useful. [4,5] Mofusion and dp-promise have already been shown to work in their respective regimes, so applying a similar idea to differential privacy in diffusion models would be a viable optimization to make.

Conclusion

We have shown that it is possible to rudimentarily attach a differential privacy module to a motion diffusion model such as MDM. This project uses an Opacus differential privacy library to apply the noise to the model, and we have attempted to analyze the model after injecting certain levels of noise. While hardware limitations affected our training parameters and the amount of epochs we could train to, our results demonstrated that it is a viable option to attach a DP module to MDM, thus laying the groundwork for future explorations in this field.

Teammate Contributions

Kevin Zhao contributed to model training and drafting of the sections regarding background information, threat modeling, results, and discussion of results. Chinmay Devadhar contributed to setting up the model/dataset on Kaggle and setting up Opacus. Renuka Bhusari contributed to model training and data analysis. Stewart Kwok contributed to model training, version debugging, and writing the abstract and methodology/design sections. Teong Seng Tan contributed to modifying Opacus, model training, result collection and visualization, and analyzing the results in the Discussions section.

Availability

The code that was run for this project can be found at the following link: https://github.com/cdevadhar/dp-mdm.

References

1. Mironov, I., Pandey, O., Reingold, O., Vadhan, S. (2009). Computational Differential Privacy. In: Halevi, S. (eds) Advances in Cryptology - CRYPTO 2009. CRYPTO 2009. Lecture Notes in Computer Science, vol 5677. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-03

356-8_8https://doi.org/10.1007/11681878_14

2. Dwork, C., McSherry, F., Nissim, K., Smith, A. (2006). Calibrating Noise to Sensitivity in Private Data Analysis. In: Halevi, S., Rabin, T. (eds) Theory of Cryptography. TCC 2006. Lecture Notes in Computer Science, vol 3876. Springer, Berlin, Heidelberg.
3. Chang, Z., Koulieris, G., Shum, H. (2023). On the Design Fundamentals of Diffusion Models: A Survey. https://arxiv.org/pdf/2306.04542
4. Wang, H., Pang, S., Lu Z., Rao Y., Zhou Y., Xue M. (2024). dp-promise: Differentially Private Diffusion Probabilistic Models for Image Synthesis. https://www.usenix.org/system/files/usenixsecurity24-wang-haichen.pdf
5. Dabral, R., Mughal, M., Golyanik, V., Theobalt, C. (2022). MoFusion: A Framework for Denoising-Diffusion-based Motion Synthesis. https://arxiv.org/abs/2212.04495
6. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A. (2022). Human Motion Diffusion Model. https://arxiv.org/abs/2209.14916
7. Cyphers, B. (2017). Understanding Differential Privacy and Why It Matters for Digital Rights. https://www.accessnow.org/understanding-differential-privacy-matters-digital-rights/.