AFRL-AFOSR-JP-TR-2024-0045

Study of Privacy Preservation Techniques for Deep Learning

Pei-Yuan Wu
NATIONAL TAIWAN UNIVERSITY
1, ROOSEVELT RD., SEC. 4
TAIPEI, ,
TWN

01/27/2024
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
20240127	Final	START DATE 20200814	END DATE 20230813

4. TITLE AND SUBTITLE
Study of Privacy Preservation Techniques for Deep Learning

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
	FA2386-20-1-4039	
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Pei-Yuan Wu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NATIONAL TAIWAN UNIVERSITY
1, ROOSEVELT RD., SEC. 4
TAIPEI
TWN

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD UNIT 45002 APO AP 96338-5002	AFRL/AFOSR IOA	AFRL-AFOSR-JP-TR-2024-0045

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Machine Learning (ML) as-a-service (MLaaS) has brought much convenience to our daily lives. However, these MLaaS are often offered through cloud computing services which raises the potential risk of privacy leakage when personal data were used in the model development. We propose to implement Privacy-Preserving Machine Learning (PPML) through data transformation, where the data is first transformed through nonlinear lossy compression mapping mechanism before sending to the cloud to have the ML service. The transformed data is not reversible and thus, the data privacy could be preserved. Moreover, the most important information for ML could be retained for the ML service in the cloud. The nonlinear lossy compression mapping mechanism can be evaluated by how difficult the adversary can perform reconstruction attack based on the nonlinearly compressed data, while as how well the MLaaS can provide its service accordingly. This in turn can be formulated as an adversarial learning problem resembling Generative Adversarial Network. In this proposal we focus on (1) The implementation of PPML to various applications such as panorama image and video data; (2) The application of secure multi-party computation (SMPC) to the training of neural network as well as its speedup for practical large-scale datasets, so as to further enhance privacy protection on the training data.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR	19

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
AKIRA NAMATAME	3152277010

Final Report
Proposal 20IOA039

Pei-Yuan Wu

January 15, 2024

- **Federal Agency:** Asian Office of Aerospace R&D
- **Grant No:** FA2386-20-1-4039
- **Project Title:** Study of Privacy Preservation Techniques for Deep Learning
- **PI:** Dr. Wu, Pei-Yuan
  peiyuanwu@ntu.edu.tw (O) 886-2-33664687 (F) 886-2-23671909
- **Submission Date:** 15 Jan 24
- **Recipient Organization:** National Taiwan University
  No. 1, Section 4, Roosevelt Rd, Da'an District,
  Taipei City, 10617, Taiwan (R.O.C.)
- **Grant Period:** 14 Aug 20 to 13 Aug 23
- **Report Term:** Final

1 Summary

In the first year of this project, we seek to improve GAN-based privatizer to achieve compressive privacy over various application scenarios such as video and/or panorama data. Towards video application, we propose the privacy preserving class overlap network (PPCON) architecture which improves the GAN-based privatizer by introducing the idea of class overlap to privacy preserving from the field of domain adaptation. We demonstrated that PPCON achieves superior utility-privacy tradeoff on identity preserving action recognition task over video sequential data; towards panorama application, we propose a spherical encoding to improve the performance of convolution neural network on omnidirectional data, which is based on great circle distance to calibrate the convolution weights on distorted regions at high-latitudes.

In the second year of the project, we focus on implementing multiparty computation on neural networks to preserve both data privacy as well as intellectual property of neural network models. We co-developed a DNN-oriented MPC library, where we ensure correctness by solving renown bugs in open-source MPC libraries. By integrating SCI library implementation ReLU operations as well as CrypTFlow2 and SIRNN truncation protocols into ABY library, we achieve both computation cost improvement and bitwise correct functionalities. Our DNN-oriented MPC library implementation is evaluated on MobileNetV2, which demonstrates the implementation improvement we have made.

2 Introduction

2.1 Privacy preserving machine learning

Cloud computing allows us to enjoy a convenient life, but it also threatens our privacy and security. Google allows us to compose or receive emails on any device just by logging in to our account, but the content of our emails may be viewed and used to recommend personalized ads [1]. Facebook allows us to express our preferences in the community and share with friends, but these content may be used and analyzed by third parties, and may even be used to manipulate elections, such as the Cambridge Analytica scandal [2].

With the emergence of privacy issues, the trend in recent years has begun to emphasize the protection of cloud data. The EU's General Data Protection Regulation [40] has had a significant impact on suppliers who need to obtain data from EU citizens, and has formulated strict regulations and penalties to protect users. In the current era, privacy is always an issue that needs to be considered at the same time, we enjoy the convenience of cloud computing.

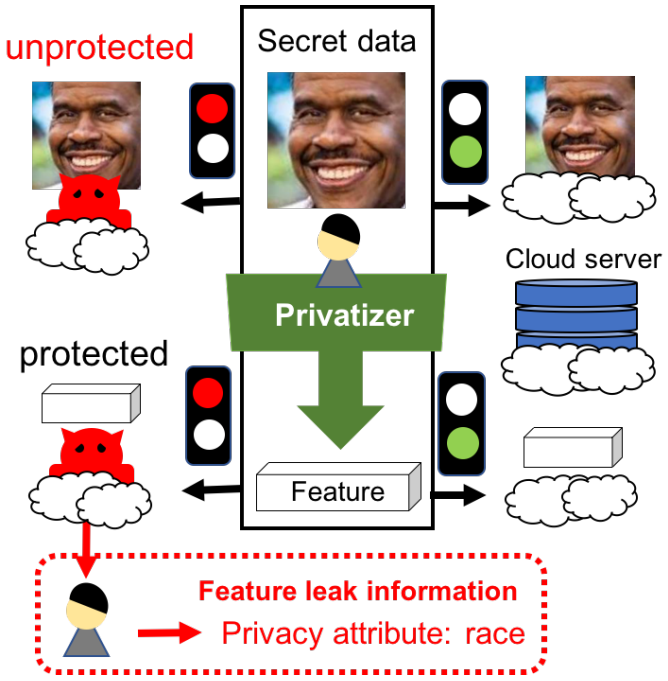

Figure 1: Uploading data to the cloud while enjoying machine learning service may reveal private information. The upper part shows that without privacy protection the attacker can directly obtain the raw data to identify the sensitive attribute information. As a result, a local privatizer is needed to obfuscate the data before sending to the cloud. The lower part illustrates the threat model where an attacker aims to steal privacy information from the obfuscated data.

Machine learning as a service(MLaaS) [31] is a kind of cloud computing. Many vendors provide cloud computing resources so that users can train models, store models, process and share data in the cloud, such as Google, Amazon, and Microsoft. AI-Rubaie [3] had conducted a complete survey of the privacy issues in MLaaS. An MLaas framework is shown in Figure.1. If the user uploads the raw data to the cloud, the data may be obtained by malicious attackers, resulting in privacy leakage. The privatizer is designed to obfuscate data before uploading the data to the cloud, by which avoiding the raw data from leaking privacy information. In the past, researchers designed many kinds of privatizers such as applying homomorphic encryption [12], removing the private area [23], or uploading only the information required by the model [32]. These methods not only obfuscated data as features to prevent privacy leakage, but also retain the information required by the model.

In recent years, with the development of deep learning, obfuscated data may be analyzed to reveal the private information of the raw data. Common privacy leaks of obfuscated data include reconstruction attacks [25] and attribute inference attacks [42] [34] [35]. In the attribute inference attack, the obfuscated data will leak the user's privacy attributes, such as race and identity. To deal with this problem, Tripathy et al. [37] proposed a generative adversarial network (GAN) model to simulate attacker through the inherent training procedure of GAN. The privatizer as a result can be trained to obfuscate the data containing as less privacy information as possible, yielding good results in their experiments.

2.2 Deep Learning with Secure Multiparty Computation Implementation

Recent research on the application of multi-party computation (MPC) to neural networks can be categorized as either towards the training-end or the inference-end. Related literature on MPC in the inference-end includes XONN [29], MiniONN [19], Chameleon [30], EzPC [5], etc, while the training-end includes SecureML [22], SecureNN [41], ABY3 [21], to name a few.

The key building blocks in inference-end MPC work mainly consist of oblivious transfer (OT), garbled circuits (GC), and secret sharing (SS). The merit is to transform a neural network, which is trained in plaintext, into a model which makes inferences based on MPC, so as to protect either the inference data privacy and/or the intellectual property of the trained neural network. There are various inference-end MPC works: XONN transforms the trained neural network into a binarized neural network (BNN), by which the costly matrix-multiplication operations are replaced with the essentially free XNOR operations in GC; Chameleon is a hybrid secure computation framework based on ABY which integrates sequential garbled circuits and optimizes vector dot product for fast matrix multiplications, while employs a semi-honest third party for offline precomputation of OTs and multiplication triples.

The core technique in training-end MPC literature is similar to that of the inference-end. However, since there are much computation pertaining to feed-forward and backward-propergations in the training end, it further highlights the issue of reducing both computation and communication costs while at the same time protects the privacy of the data/model. SecureML is the first work that incorporates MPC with deep learning, which supports secure arithmetic operations on

fixed-point trunction, along with MPC-friendly alternatives to non-linear activation functions such as sigmoid or softmax. SecureNN further improves based on SecureML, with a more modulized structural design. The SecureNN open source code has been adopted by famous software companies such as Facebook and Micriosoft, and further extended into other open source code such as EzPC-Porthos and Facebook-Crypten, to name a few.

Despite the rapid development of MPC-based machine learning packages, it is often the case that the public MPC library contains well-known unsolved bugs or is yet complete with crucial functionalities missing. For instance, the experimental MPC library CrypTen developted by Facebook is neither satisfactory in computation performance nor complete in functionality. As a Python package, CrypTen suffers from the general interpreter lock (GIL) issue, which makes it difficult to parallelize and hinders its computation performance. Besides, CrypTen also lacks the save/load functionalities. Another instance is the famous MPC package ABY, where the multiplication triples (MTs) generated during OT is likely to be erroneous, rendering erroneous MPC inferenced result.

3 Accomplishments

In this project we seek to improve GAN-based privatizer, and extend its application to various scenarios such as video and/or panorama data. We further elaborate the privacy enhancement through incorporating MPC with neural networks. A list of tasks and their objectives is provided below:

Task 1: Nonlinear compressive privacy scheme for panorama pictures.

- **Objective:** Extend the nonlinear compressive privacy scheme to panoramic photography, so a user can enjoy panorama-based MLaaS on cloud with less risk of revealing private content in the panorama picture.
- **Progress and Achievements:**

1. We proposed a spherical encoding to improve the performance of convolution neural network on omnidirectional data, which is based on great circle distance to calibrate the convolution weights on distorted regions at high-latitudes.
   Description: Equirectangular projection maps the longitude and latitude coordinates of the spherical image to the x- and y-coordinates of a plane image, respectively. This, however, is often accompanied with undesired distortions around the north and south poles, as well as discontinuity on the two sides (see Figure.2). As a result, special care must be taken before applying conventional euclidean-based learning algorithms [7] to equirectangular images projected from spherical signals. (See Section.5.1 for more details)

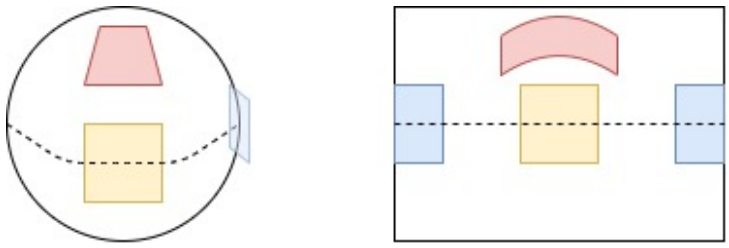

Figure 2: Distortion near poles (red part), discontinuity on two sides (blue part), and normal grid near equator (yellow part)

2. The proposed spherical encoding is compatible to prevailing deep learning modules, with its effectiveness supported by experimental results on omnidirectional image classification tasks.
   Description: We provide empirical evidences in Section.6.1 and Section.6.2.

Task 2: Nonlinear compressive privacy scheme for elder care remote monitoring.

- **Objective:** Extend the nonlinear compressive privacy scheme to motion detection in video sequence, so that remote monitoring can be achieved by compressed video sequence with less risk of revealing private content within.
- **Progress and Achievements:**

1. We improve the GAN-based privatizer by introducing the idea of *class overlap* to privacy preserving from the field of domain adaptation.
   Description: In the field of domain adaptation, researchers devoted to make the class overlap so as to bring different domain data into a common feature space. We adopt such idea for the privacy preserving purpose, where we make different privacy classes overlap with each other to defend against attribute inference attack. (See Section.5.2.1 for more details)

2. We propose the *privacy preserving class overlap network* (PPCON) architecture which applies distribution matching technique to train the privatizer.
   Description: We choose *Wasserstein distance* [4] to measure the similarity between data distributions. The model is trained to minimize the distance between different sensitive attributes, thus hindering the attacker from finding out the correct sensitive attribute. (See Section.5.2.2 for more details)
3. We demonstrated that PPCON achieves superior utility-privacy trade-off on video sequential data.
   Description: We consider the application scenario where we wish to maximize the utility for MLaaS to correctly recognize the action in a video, while minimizing the risk of revealing privacy information such as the actors' identity. We demonstrated that by nonlinearly transform the original video into obfuscated data by PPCON, the MLaaS can provide good action recognition service while the adversary find itself difficult to infer the actor identity information from the obfuscated data. (See Section.6.3 for more details)

Task 3: Privacy enhancement for training CPGAN.

- **Objective:** Implement secure multi-party computation (SMPC) into the training of CPGAN, so as to protect the privacy of training data.
- **Progress and Achievements:** In collaboration with Industrial Technology Research Institute (ITRI) and National Cheng Kung University (NCKU), we developed a deep neural network (DNN)-oriented MPC library - Privacy Preserving DNN (PPDNN) - with the following improvements:
  1. We ensure the correctness of MPC-inference result through fixing a serious bug in ABY library.
  2. We integrate ReLU gate with SCI library implementation within ABY framework, leading to computation performance improvement.
  3. We achieve bitwise correct between ciphertext and plaintext inference results through integrating MPC truncation protocols within ABY framework.

  Please find empirical evidences in Section.6.4.

4 Impact

To enhance data privacy in machine learning, in this project we extend our proposed data-driven privatization mechanism CPGAN from image to video data, where user data is transformed into some low-dimension representation before sending it to the cloud for enjoying the machine learning service. We elaborate both a Wasserstein generative adversarial network and the idea of class overlapping to obfuscate data for better resilience against the leakage of attribute-inference attacks (i.e., malicious inference on users' sensitive attributes). Experiments show that the proposed method can be employed to enhance current state-of-the-art works and achieve superior privacy–utility trade-off. Furthermore, the proposed method is shown to be less susceptible to the influence of imbalanced classes in training data. Finally, we provide a theoretical analysis of the performance of our proposed method to give a flavour of the gap between theoretical and empirical performances. We published our findings and shared with the research community. [18]

In joint collaboration with Industrial Technology Research Institute (ITRI) and National Cheng Kung University (NCKU), our team participated in the implementation of bit-wise correct arithmetic sharing as well as the development of a privacy preserving neural inference compiler framework that incorporates secure multiparty computation into the neural network computation, so as to protect the intellectual property of the model weights as well as the privacy of user data from being leaked to others during the inference of a deep learning model. We have transfered the technology to ITRI who plays an important role in further spreading the technology to the industry. We also published our findings and shared with the research community. [15, 38]

To better deal with the explosive amount of information contained in the omni-directional view of panorama videos, we propose a visual saliency prediction model that directly takes panorama video in the equirectangular format. We study the statistical properties of viewing biases present in panorama datasets across various video types, which leads to the design of a fusing mechanism that incorporates the predicted saliency map with the viewing bias in an adaptive manner. The proposed model yields state-ofthe-art performance, as evidenced by empirical results over renowned panorama visual saliency datasets such as Salient360!, PVS, and Sport360. We published our findings and shared with the research community [6].

5 Proposed Approach

5.1 Spherical encoding for neural networks on panorama

We consider an equirectangular image by two parts: i) the position of a pixel is regarded as positional information, and ii) the context and the value of channels are regarded as features. In equirectangular projection:

$$f(\lambda,\ \varphi) = (R(\lambda - \lambda_0), R(\varphi - \varphi_0)). \tag{1}$$

Here, R is the radius of the globe, (λ, φ) is the longitude and latitude of the location to project, (λ_0, φ_0) is the central parallel and the meridian of the map, and (x, y) is the coordinate on the equirectangular projected image. If we acquire the coordinate of input pixel we can have the positional information of pixel from eq. 1.

Coming up with the positional encoding in self-attention mechanism, we introduce the relationship between convolution and self-attention in sec. 5.1.1 and positional encoding in sec. 5.1.2. We propose our structure in sec. 5.1.5.

5.1.1 Convolution and Self-Attention on Feature Extraction

In 2D convolution the pixel value on position (i, j) is evaluated as:

$$f(i,j)=\sum_{|\Delta_W|,|\Delta_H|\leq\lfloor\frac{K}{2}\rfloor} F_{(\Delta_W,\Delta_H)}\cdot\mathfrak{I}_{(i+\Delta_W,j+\Delta_H)}. \tag{2}$$

Here $F\in\mathbb{R}^{K\times K\times D_{in}}$ is the weights of the convolution filter, K is the size of the filter, and $\mathfrak{I}\in\mathbb{R}^{W\times H\times D_{in}}$ is the input image of size $W\times H$ and D_{in} channels. $\mathfrak{I}_{(i,j)}\in\mathbb{R}^{D_in}$ is the channel values of the pixel on position (i,j). Note that convolution is a shift-invariant operation, which is not the case for equirectangular projected images. For instance, shifting an object from the equator to the North or South pole on sphere leads to not only a shift in the equirectangular projected image, but also a distortion that depends on latitude. Therefore, a position-dependent filter is needed in panoramic image processing.

Another useful method is self-attention mechanism for computer vision. Let $I\in\mathbb{R}^{WH\times D_{in}}$ be the flattened image. In self-attention, the output of a query pixel q is computed as the weighted sum of every key pixel with the weight of attention probabilities, where the attention probabilities measures the similarity between the query pixel and a key pixel as follows:

$$\text{Self-Attention}(I)_q=\sum_k^{WH}\text{softmax}(A_{q,:})_k I_{k,:}W_{val},\quad \text{softmax}(A_{q,:})_k=\frac{\exp(A_{q,k})}{\sum_p\exp(A_{q,p})}. \tag{3}$$

Here, $W_{val}\in\mathbb{R}^{D_{in}\times D_{out}}$ linearly transforms the input image I from D_{in} channels to D_{out} channels, $A\in\mathbb{R}^{WH\times WH}$ denotes the pairwise attention score between each pixel in the image I, and $\text{softmax}(A_{q,:})_k$ is the attention probabilities on pixel q contributed by pixel k. The attention score between pixel q and pixel k is commonly calculated in an inner product form:

$$A=(IW_{qry})(IW_{key})^T=IW_{qry}W_{key}^TI^T. \tag{4}$$

Here, W_{qry}, $W_{key}\in\mathbb{R}^{D_{in}\times D_{out}}$ linearly transform each pixel from $\mathbb{R}^{D_{in}}$ to $\mathbb{R}^{D_{out}}$, on which the inner product is computed as attention score between pixels. In self-attention mechanism, we can use positional encoding to preserve the positional information of each pixel.

$$A=(I+E)W_{qry}W_{key}^T(I+E)^T, \tag{5}$$

where $E\in\mathbb{R}^{WH\times D_{in}}$ is the positional encoding of the pixels. (see Sec. 5.1.2 for more details)

It has been pointed out by Cordonnier *et al.* [9] that the convolution layer can in fact be realized through multi-head self-attention mechanism, where the position information of each key pixel is taken into account through positional encoding. In this work, we extend the idea of realizing the convolution operation through self-attention mechanism to omnidirectional images. Moreover, we propose **Spherical Encoding** for equirectangular projected image to preserve positional information of each pixel based on the great circle distance. We will further elaborate spherical encoding in sec. 5.1.2.

5.1.2 Encoding with Positional Information

There are several ways of positional encoding to preserve the positional information of pixels, including *absolute* and *relative* encoding [10]. In *absolute* encoding, each pixel p is represented by a fixed or learned vector $E_p^{abs}\in\mathbb{R}^{D_{in}}$, by which the positional information between a pair of pixels is represented by the dot product between their encoding vectors. More precisely, the attention score between query pixel and key pixel is computed as:

$$\begin{aligned}A_{(q,k)}^{abs}&=(I_q+E_q^{abs})W_{qry}W_{key}^T(I_k+E_k^{abs})^T\\&=I_qW_{qry}W_{key}^TI_k^T+E_q^{abs}W_{qry}W_{key}^TI_k^T+I_qW_{qry}W_{key}^TE_k^{absT}+E_q^{abs}W_{qry}W_{key}^TE_k^{absT},\end{aligned} \tag{6}$$

where $I_q\in\mathbb{R}^{D_{in}}$ is the q^{th} pixel in teh flattened image I.

On the contrary, in *relative* encoding, it is the relative position between the key and query pixel $\mathbf{q},\mathbf{k}\in[1,W-1]\times[0,H-1]$ that is considered, by which the attention score is computed as:

$$\begin{aligned}A_{\mathbf{k}-\mathbf{q}}^{rel}&=(I_q+e)W_{qry}W_{key}^T(I_k+E_{\mathbf{k}-\mathbf{q}}^{rel})^T\\&=I_qW_{qry}W_{key}^TI_k^T+I_qW_{qry}(W_{key}^TE_{\mathbf{k}-\mathbf{q}}^{relT})+(eW_{qry})W_{key}^TI_k^T+(eW_{qry})(W_{key}^TE_{\mathbf{k}-\mathbf{q}}^{relT})\\&=I_qW_{qry}W_{key}^TI_k^T+I_qW_{qry}W_{key}^TE_{\mathbf{k}-\mathbf{q}}^{relT}+uW_{key}^TI_k^T+vW_{key}^TE_{\mathbf{k}-\mathbf{q}}^{relT},\end{aligned} \tag{7}$$

where $E_{\mathbf{k}-\mathbf{q}}^{rel}\in\mathbb{R}^{D_{in}}$ is the relative encoding vector that depends on the relative position between the key and query pixels, and $e\in\mathbb{R}^{D_{in}}$ is a learned or fixed vector.

5.1.3 Spherical absolute encoding

In order to contain the spherical topological information, we propose **Spherical Encoding** based on the great circle distance. For two point (λ_1,φ_1), (λ_2,φ_2) on the unit sphere in spherical coordinate, the distance in between can be written as:

$$d((\lambda_1,\varphi_1),(\lambda_2,\varphi_2))=\arccos(\sin\varphi_1\sin\varphi_2+\cos\varphi_1\cos\varphi_2\cos\lambda_1\cos\lambda_2+\cos\varphi_1\cos\varphi_2\sin\lambda_1\sin\lambda_2). \tag{8}$$

Here, the arc-cosine function can be approximated by the Taylor series expansion:

$$\begin{aligned} d((\lambda_1,\varphi_1),(\lambda_2,\varphi_2))&=\frac{\pi}{2}-\sum_{k=0}^{\infty}\frac{1}{2^{2k}}\frac{(2k)!}{(k!)^2}\frac{1}{2k+1}(\sin\varphi_1\sin\varphi_2+\cos\varphi_1\cos\varphi_2\cos\lambda_1\cos\lambda_2+\cos\varphi_1\cos\varphi_2\sin\lambda_1\sin\lambda_2)^{2k+1}\\ &\sim\frac{\pi}{2}-\Psi_n(\lambda_1,\varphi_1)\cdot\Psi_n(\lambda_2,\varphi_2), \end{aligned} \tag{9}$$

where $\Psi_n(\lambda,\varphi)$ is the spherical encoding that corresponds to the n^{th} order Taylor series approximation, which is a $\sum_{k=0}^{n}\binom{2k+3}{2}=\frac{2}{3}n^3+\frac{7}{2}n^2+\frac{35}{6}n+3$ dimensional vector where each element (indexed by k,p,q,r where $0\le k\le n$, $0\le p,q,r$, and $p+q+r=2k+1$) takes the form:

$$\left[\Psi_n(\lambda,\varphi)\right]_{k,p,q,r}=\frac{1}{2^k}\frac{(2k)!}{k!}\sqrt{\frac{1}{p!q!r!}}\sin^p\varphi\cos^{q+r}\varphi\cos^q\lambda\sin^r\lambda, \tag{10}$$

Though Taylor expansion has infinite terms, as convolution usually operates on local patterns, the low order terms are fairly enough to elaborate positional information of adjacent pixels appropriately. The first order spherical encoding is given by $\Psi_0(\lambda,\varphi)=(\sin\varphi,\cos\varphi\cos\lambda,\cos\varphi\sin\lambda)$.

5.1.4 Spherical relative encoding

By rewriting eq.9 with $(\varphi_1,\ \lambda_1)$ and $(\varphi_2,\ \lambda_2)$ replaced by (φ,λ) and $(\varphi+\Delta\varphi,\lambda+\Delta\lambda)$, respectively, we can express the great circle distance in terms of spherical relative encoding as follows:

$$\begin{aligned} &d((\lambda_1,\varphi_1),(\lambda_2,\varphi_2))\\ &=\frac{\pi}{2}-\sum_{k=0}^{\infty}\frac{1}{2^{2k}}\frac{(2k)!}{(k!)^2}\frac{1}{2k+1}(\sin\varphi_1\sin\varphi_2+\cos\varphi_1\cos\varphi_2\cos\lambda_1\cos\lambda_2+\cos\varphi_1\cos\varphi_2\sin\lambda_1\sin\lambda_2)^{2k+1}\\ &=\frac{\pi}{2}-\sum_{k=0}^{\infty}\frac{1}{2^{2k}}\frac{(2k)!}{(k!)^2}\frac{1}{2k+1}\Big(\sin^2\varphi\cos\Delta\varphi+\sin\varphi\cos\varphi\sin\Delta\varphi+\cos^2\varphi\cos\Delta\varphi\cos\Delta\lambda-\cos\varphi\sin\varphi\sin\Delta\varphi\cos\Delta\lambda\Big)^{2k+1}\\ &\approx\frac{\pi}{2}-\Phi_n^{(qry)}(\varphi,\lambda)^T\Phi_n^{(key)}(\Delta\varphi,\Delta\lambda) \end{aligned} \tag{11}$$

Here $\Phi_n^{(qry)}(\varphi,\lambda)$ and $\Phi_n^{(key)}(\Delta\varphi,\Delta\lambda)$ are the query/key encodings that correspond to the n^{th} order Taylor series approximation, respectively. The query encoding $\Phi_n^{(qry)}(\varphi,\lambda)$ is a $\frac{1}{3}k^4+\frac{8}{3}k^3+\frac{23}{3}k^2+\frac{28}{3}k+4$ dimensional vector, with each element (indexed by k,p,q,r,s where $0\le k\le n$, $p,q,r,s\ge 0$, and $p+q+r+s=2k+1$) taking the form

$$\left[\Phi_n^{(qry)}(\varphi,\lambda)\right]_{k,p,q,r,s}=\frac{1}{2^k}\frac{(2k)!}{k!}\sqrt{\frac{1}{p!q!r!s!}}(-1)^s\sin^{2p+q+s}\varphi\ \cos^{q+2r+s}\varphi.$$

The key encoding has the same dimension with elements taking the form

$$\left[\Phi_n^{(key)}(\Delta\varphi,\Delta\lambda)\right]_{k,p,q,r,s}=\frac{1}{2^k}\frac{(2k)!}{k!}\sqrt{\frac{1}{p!q!r!s!}}\cos^{p+r}\Delta\varphi\sin^{q+s}\Delta\varphi\cos^{r+s}\Delta\lambda$$

In particular, for first order approximation the query and key encodings are given by

$$\begin{aligned} \Phi_0^{(query)}(\varphi,\lambda)&=(\sin^2\varphi,\sin\varphi\cos\varphi,\cos^2\varphi,-\sin\varphi\cos\varphi)\\ \Phi_0^{(key)}(\Delta\varphi,\Delta\lambda)&=(\cos\Delta\varphi,\sin\Delta\varphi,\cos\Delta\varphi\cos\Delta\lambda,\sin\Delta\varphi\cos\Delta\lambda) \end{aligned}$$

Spherical relative encoding allows the proximity information between query and key pixels on the sphere to be represented as two integral parts in equirectangular format: the query encoding that only depends on the location of the query pixel (φ,λ), and the key encoding that solely depends on the relative position $(\Delta\varphi,\Delta\lambda)$. This allows us to effectively adjust the attention scores for pixels in the receptive field of equirectangular format based on its proximity on the sphere.

5.1.5 Encoding and Convolution

Though self-attention model hits a great success on various computer vision [26] [45] and natural language processing [11] applications, its dire memory usage poses a serious issue to be reckoned with. More precisely, for an input image of size $W \times H$ and $D_{in} \in O(WH)$ channels, the self-attention operation requires $O(W^2H^2)$ memory usage, a $\frac{WH}{D_{in}+D_{out}}$-fold increase compared to the convolution-based approach which requires $O(WH(D_{in}+D_{out}))$ memory usage.

To relieve the huge memory usage of self-attention, a common approach is convolution-based attention [43] [24] which uses additional parameters and incorporates techniques such as spatial attention or channel attention to learn the importance of features. This leads to memory usage in the order of $O(WH \times (D_{in}+D_{out}))$, for which D_{in} and D_{out} are usually much smaller than WH.

Instead of using additional parameters, here we use the inner product of spherical encoding as the attention map to refine the convolution features:

$$\begin{aligned} f^*(i,j) \sim & \sum_{|\Delta_W|,|\Delta_H| \le \lfloor \frac{K}{2} \rfloor} \frac{e^{d((\lambda_i,\varphi_j),(\lambda_{i+\Delta_W},\varphi_{j+\Delta_H}))}}{\sum_{|\Delta_W|,|\Delta_H| \le \lfloor \frac{K}{2} \rfloor} e^{d((\lambda_i,\varphi_j),(\lambda_{i+\Delta'_W},\varphi_{j+\Delta'_H}))}} F_{(\Delta_W,\Delta_H)} \cdot \mathfrak{I}_{(i+\Delta_W,j+\Delta_H)} \\ = & \sum_{|\Delta_W|,|\Delta_H| \le \lfloor \frac{K}{2} \rfloor} \frac{e^{-\Psi_n(\lambda_i,\varphi_j)\cdot\Psi_n(\lambda_{i+\Delta_W},\varphi_{j+\Delta_H})}}{\sum_{|\Delta_W|,|\Delta_H| \le \lfloor \frac{K}{2} \rfloor} e^{-\Psi_n(\lambda_i,\varphi_j)\cdot\Psi_n(\lambda_{i+\Delta'_W},\varphi_{j+\Delta'_H})}} F_{(\Delta_W,\Delta_H)} \cdot \mathfrak{I}_{(i+\Delta_W,j+\Delta_H)}, \end{aligned} \tag{12}$$

where (λ_i,φ_j) is the longitude and latitude of the pixel (i,j) on the equirectangular projected image.

In equirectangular projected images, pixels near the north and south pole which represent small area on the sphere will be overly weighted in conventional convolution. In our proposed spherical encoding, however, observe that in the most prevailing 3×3 kernel size scenario, the great circle distance between center pixel and high-latitude pixels are generally smaller than that of low-latitude pixels(fig. 3). This generally leads to smaller weights assigned to high-latitude pixels and mitigates the issue of overly weighted high-latitude pixels suffered in conventional convolution on equirectangular projected images. Similarly, the discontinuity of two sides of the image can also be fixed by the $\sin\varphi$ term in the spherical encoding.

Figure 3: In the 3×3 kernel size setting, the pixels that are farther away from center (blue) represent greater surface area, and are assigned more attention score through **Spherical encoding**. (eq. 12)

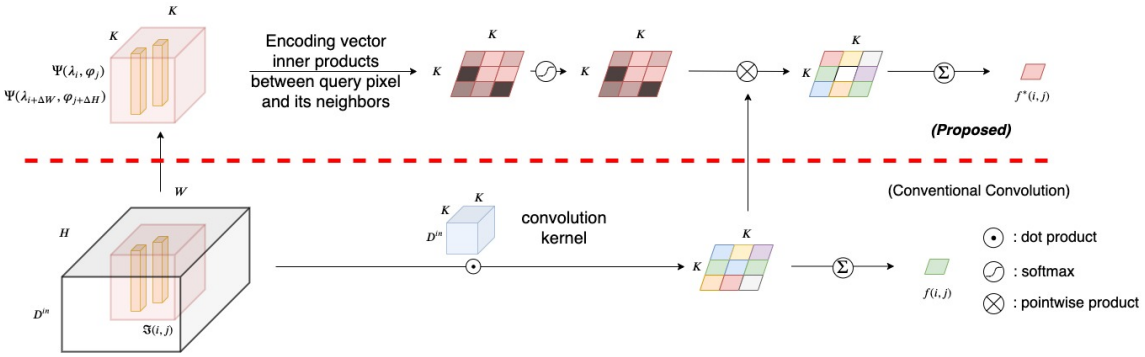

Figure 4: Process of **Spherical encoding** (eq. 12)

5.2 Privacy preserving class overlap network

5.2.1 Avoid privacy leakage through class overlapping

Our goal is to find a privatizer that obfuscates data before uploading to the cloud, so as to enjoy the machine learning service while avoiding leaking private information. Intuitively, the distributions of the obfuscated data with various sensitive labels should be "closely overlapping" in $\mathcal{Z}$, and therefore difficult to distinguish. Towards this end, we adopt the concept of WGAN [4] to remove the private information in obfuscated data. Furthermore, as the sensitive attributes usually contain more than two classes, we adopt the mathematical approach proposed by Li et al. [17] and proposed a privacy preserving class overlap network (PPCON) to obtain the privatizer through a data-driven approach.

The architecture of PPCON is shown in Fig 5. The Privatizer first obfuscates the original data, and the obfuscated data is then subsequently fed into the service network as well as an estimator network, which represent how well / harmful the service / adversary brings upon based on the obfuscated data, respectively.

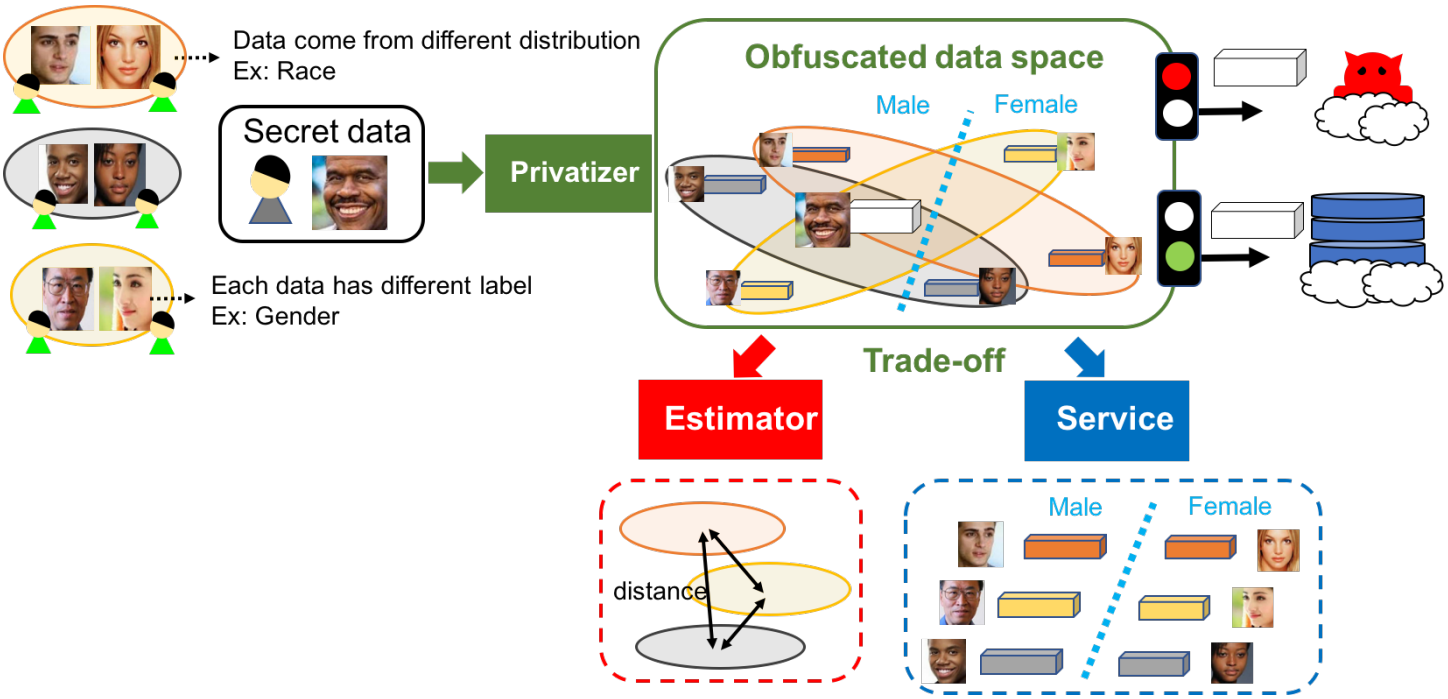

Figure 5: The proposed PPCON architecture allows users to enjoy cloud services as well as protecting their privacy. PPCON adopts a data-driven approach to train the privatizer, which obfuscates the data to be uploaded to the cloud.

PPCON is trained through an iterative process, where each iteration contains two phases: In phase one, the service network adopts its parameters so as to minimize the cross-entropy loss that measures the quality of the service based on the obfuscated data; whereas the estimator network computes the Wasserstein distance between data distributions of various sensitive labels in $\mathcal{Z}$, as an indication of how "closely overlapping" the data distribution of various sensitive labels is. In phase two, the parameters in both service / estimator networks are fixed, and the privatizer adopts its parameters so as minimize the cross-entropy loss pertaining to the service network, as well as the Wasserstein distances evaluated by the estimator network. The objective functions of the estimator network, service network, and the privatizer are elaborated as follows:

5.2.2 Wasseerstein distance estimator

The estimator measures how "closely overlapping" it is between obfuscated data points of different sensitive labels. Suppose there are S classes of sensitive attributes, where the data points are divided into S distributions $P_1,P_2,...,P_S$ according to their sensitive labels. Then an intuitive way to quantify how obfuscated data points of different sensitive labels overlap is

$$\frac{2}{S(S-1)}\sum_{1\le i<j\le S} W(f_p(P_i),f_p(P_j)), \tag{13}$$

where $f_p(P)$ denotes the distribution of $f_p(x)$ for which x is randomly drawn from distribution P, and $W(P_A,P_B)$ denotes the Wasserstein distance between distributions P_A and P_B.

However, when many types of privacy attributes exist, the computation cost for (13) grows quadratically with S. To speed up the computation, we instead only compute the Wesserstein distances between distribution pairs $(f_p(P_i),f_p(P_{/i}))$ for each $1\le i\le S$, where $P_{/i}$ denotes the normalized sum of the remaining distributions:

$$P_{/i}=\frac{1}{S-1}\sum_{1\le j\le S,\text{ and } j\neq i} P_j. \tag{14}$$

This leads to our design of the estimator which computes

$$\begin{aligned}&\frac{1}{S}\sum_{i=1}^{S} W(f_p(P_i),f_p(P_{/i}))\\ =&\frac{1}{S}\sum_{i=1}^{S}\max_{\|f_i\|_L\le 1}(E_{x\sim P_i}[f_i(f_p(x))]-E_{x\sim P_{/i}}[f_i(f_p(x))]).\end{aligned} \tag{15}$$

Here the second equality in (15) follows by the Kantorovich-Rubensted dual form [39]

$$W(P_A,P_B)=\max_{\|f\|_L\leq 1} E_{x\sim P_A}[f(x)]-E_{x\sim P_B}[f(x)], \tag{16}$$

where $\|f\|_L\leq 1$ denotes that the Lipschitz constant of the function $f(\cdot)$ is at most 1, i.e. $|f(x')-f(x)|\leq||x'-x||_2$, and the Lipschitz-smooth nonlinear function f can be approximated by a neural network [4].

For the sake of computational efficiency, in our implementation the Lipschitz-smooth nonlinear functions $f_1,...,f_S$ in (15) are approximated with a single neural network with S outputs $[f_1,...,f_s]$. To enable the minimization of (15), we adopt the mathematical approach proposed by Li et al. [17] as follows: Let N be the number of training data, and n_s be the number of data whose sensitive attribute is of class s. Denote $\pi_s=\frac{n_s}{N}$, the objective function that the estimator aims to maximize can be written as (cf.(15)):

$$\begin{aligned}
&\frac{1}{S}\sum_{i=1}^{S}\Big(E_{x\sim P_i}[f_i(f_p(x))]-E_{x\sim P_{/i}}[f_i(f_p(x))]\Big)\\
=&\frac{1}{S}\sum_{j=1}^{S}E_{x\sim P_j}[f_j(f_p(x))]-\frac{1}{S(S-1)}\sum_{i\neq j}E_{x\sim P_j}[f_i(f_p(x))]\\
=&\frac{1}{S}\sum_{j=1}^{S}E_{x\sim P_j}\left[f_j(f_p(x))-\frac{1}{S-1}\sum_{i\neq j}f_i(f_p(x))\right]\\
=&E_{s\sim\pi}\left[E_{x\sim P_s}\left[\frac{1}{S\pi_s}\gamma_s^T f_{EST}(f_p(x))\right]\right],
\end{aligned} \tag{17}$$

where $s\sim\pi$ indicates s is a random variable that takes value j with probability π_j, and $f_{EST}(y)=[f_1(y),...,f_S(y)]^T$, and $\gamma_s=[\gamma_s^{(1)},...,\gamma_s^{(S)}]^T\in\mathbb{R}^S$ is defined as

$$\gamma_s^{(i)}=\begin{cases} -\frac{1}{S-1} & \text{, if } i\neq s\\ 1 & \text{, if } i=s\end{cases} \tag{18}$$

We may empirically approximate (17) with a mini-batch of data,

$$\begin{aligned}
&E_{s\sim\pi}\left[E_{x\sim P_s}\left[\frac{1}{S\pi_s}\gamma_s^T f_{EST}(f_p(x))\right]\right]\\
\simeq&\frac{1}{SN}\sum_{i=1}^{N}\pi_{s_i}^{-1}\gamma_{s_i}^T f_{EST}(f_p(x_i)),
\end{aligned} \tag{19}$$

where s_i represents the privacy label of x_i.

5.2.3 Service and Privatizer

Here we consider the classification service with C categories. The service adopts its parameters so as to minimize the cross-entropy loss that measures the quality of the service based on the obfuscated data. Denote $f_{SER}:\mathcal{Z}\to\mathcal{U}$, and $\mathcal{U}=\{u=[u^{(1)},...,u^{(C)}]^T\in\mathbb{R}^C|\ \|u\|_1=1,u^{(i)}\geq 0,\forall i=1,...,C\}$. The objective function of the service is:

$$\min_{f_{SER}}-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}u_i^{(c)}\log(f_{SER}^{(c)}(f_p(x_i))), \tag{20}$$

where $f_{SER}^{(c)}$ is the c^{th} entry of the output, which is the predicted score of the c^{th} category, and $u_i^{(c)}$ means the c^{th} entry of the label of x_i in one-hot representation.

The privatizer adopts its parameters so as minimize the cross-entropy loss in (20) pertaining to the service, as well as the Wasserstein distances in (19) evaluated by the estimator. The complete objective function of PPCON is hence given by

$$\begin{aligned}
\min_{f_p,f_{SER}}\max_{\|f_{EST}\|_L\leq 1}&-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}u_i^{(c)}\log(f_{SER}^{(c)}(f_p(x_i)))\\
&+\lambda\frac{1}{SN}\sum_{i=1}^{N}\pi_{s_i}^{-1}\gamma_{s_i}^T f_{EST}(f_p(x_i)),
\end{aligned} \tag{21}$$

where λ controls the trade-off between the estimator and the service.

5.2.4 Algorithm

Denote θ_p as the parameters in the privatizer, θ_{EST} as the parameters in the estimator, and θ_{SER} as the parameters in the service. The training process of PPCON is illustrated in Algorithm 1. In lines 3-7, the objective function of the estimator is evaluated, and the parameters in the estimator are updated through gradient ascent. Here we choose k_E=7 as the number of inner loop iterations (line 6), and we apply the spectrum normalization [20] to implement the Lipschitz constraint. In lines 8-10, the objective function of service is evaluated, and the parameters in the service are updated through gradient descent. In lines 11-12, the objective function of the privatizer is evaluated. The objective function of the privatizer is the weighted sum of the objective functions of the service and the estimator, and the parameters in the privatizer are updated through gradient descent.

Algorithm 1: PPCON

1 **for** *iter*=1 *to maxiter* **do**
2 Sample a mini-batch of m data points $\{(x_i,u_i,s_i)\}_{i=1}^{m}$ from D_{train}.
3 Compute estimator loss $L_E=\frac{1}{Sm}\sum_{i=1}^{m}\pi_{s_i}^{-1}\gamma_{s_i}^{T}f_{EST}(f_p(x_i))$.
4 **for** $iter_E$=1 *to* k_E **do**
5 Compute $\nabla\theta_{EST}=\frac{\partial L_E}{\partial\theta_{EST}}$.
6 Update $\theta_{EST}\leftarrow\theta_{EST}+\eta_{EST}\nabla\theta_{EST}$.
7 **end**
8 Compute service loss $L_S=-\frac{1}{m}\sum_{i=1}^{m}\sum_{c=1}^{C}u_i^{(c)}\log(f_{SER}^{(c)}(f_p(x_i)))$.
9 Compute $\nabla\theta_{SER}=\frac{\partial L_S}{\partial\theta_{SER}}$.
10 Update $\theta_{SER}\leftarrow\theta_{SER}-\eta_{SER}\nabla\theta_{SER}$.
11 Compute $\nabla\theta_P=\frac{\partial(L_S+\lambda L_E)}{\partial\theta_p}$.
12 Update $\theta_p\leftarrow\theta_p-\eta_p\nabla\theta_P$.
13 **end**

6 Results and discussions

6.1 Adopt spherical encoding for residual module

Residual module has become a common architecture design for deep learning which often makes deep network structure more easily to optimize, and achieves better performance [14]. To elaborate the adaptability of the proposed spherical encoding in deeper models with residual modules, we conduct experiments on classification task over omnidirectional datasets, and demonstrate the ability of feature extraction in terms of classification accuracy.

Experiment Setup The Resnet18 is taken as the backbone where the convolution unit is replaced by (a) conventional convolution, (b) SphereNet convolution, and (c) spherical encoding with convolution *(our method)*. Note that in the experiement of SphereNet convolution, the pooling functions are substituted by the spherical pooling in their own work [8], for the purpose of addressing the issue of oversampling on high latitude regions.

Result As demonstrated in Table. 1, the proposed spherical encoding yields the best results. It is also observed that SphereNet convolution performs slightly worse than conventional convolution in the omni-CIFAR100 classificaiton task. This indicates that SphereNet convolution does not benefit from the residual structures. Here we give an intuitive explanation as follows:

As illustrated in fig. 6, in SphereNet convolution the distortion issue is dealt with by maintaining the size of receptive field on the tangent plane, with the output representing the convolution of spherical data on the tangent plane. This leads to a receptive field whose size depends on the latitude in the equirectangular format. As in residual module the convolution

Convolution	omni-CIFAR10	omni-CIFAR100
Conventional Convolution	0.8339	0.6020
SphereNet Convolution	0.8346	0.5979
Spherical Encoding(ours)	**0.8461**	**0.6413**

Table 1: Classification accuracy of various convolution schemes on ResNet18.

Feature extractor	Encoding	Omni-CIFAR10	Omni-CIFAR100
Conventional Convolution	-	0.8339	0.6020
Self-Attention	absolute encoding	0.8005	0.5747
Self-Attention	relative encoding	0.8178	0.5856
Self-Attention	spherical encoding (abs)	0.8230	0.5912
Self-Attention	spherical encoding (rel)	0.8000	0.5815
Conventional Convolution	spherical encoding (abs)	0.8461	0.6404
Conventional Convolution	spherical encoding (rel)	0.8294	0.6186
Conventional Convolution	spherical encoding (2^{nd} order)	**0.8511**	**0.6413**

Table 2: Classification accuracy of various encoding schemes on omni-cifar10 and omni-cifar100

result is added to the original input in equirectangular format, we hypothesize that the latitude-varying receptive field in the equirectangular format causes inconsistency in the addition operation in residual modules.

In contrast to SphereNet, in our method the size of receptive field remains identical in the equirectangular format, regardless of the latitude. That is, the convolution is computed as a weighted sum of samples drawn from a surface region whose latitude and longitude spans $K\Delta\varphi$ and $K\Delta\lambda$, respectively. Here K denotes the kernel size, while $\Delta\varphi$ and $\Delta\lambda$ denote the latitude and longitude spans of the surface region that a pixel in equirectangular format represents, respectively. The distortion issue is instead dealt with through spherical encoding, where pixels near the North and South poles will be assigned smaller weights computed as the dot product of spherical encoding, which is consistent to the intuition that high latitude pixels in equirectangular format represents smaller surface regions on the sphere.

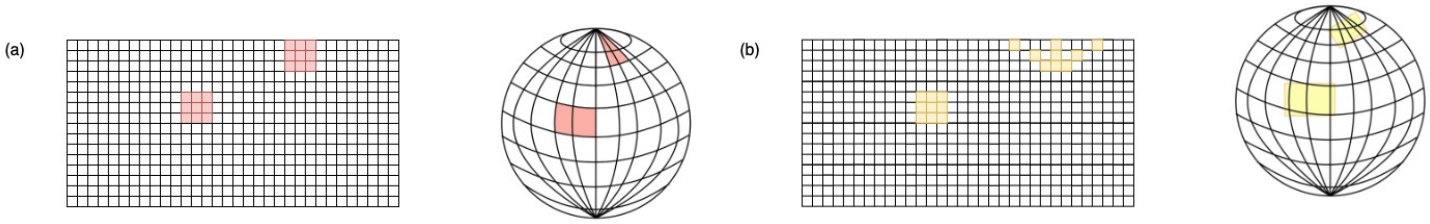

Figure 6: The physical meaning of pixel on equirectangular projection and on SphereNet kernel. (a) The pixel on the actual sphere surface by equirectangular projection (b) The receptive field of SphereNet kernel.

6.2 Adopt spherical encoding for self-embedding

We follow the experiment setup in [9] and [26] to demonstrate the effect of various encoding schemes, where conventional convolution and fully self-attention based models are compared over omnidirectional image datasets. We use the aforementioned Resnet18 as the backbone, and replaced the convolution layers with self-attention layers incorporated with various encoding schemes, as illustrated in Table. 2.

Result As illustrated in Table. 2, the fully self-attention model with either absolute or relative encoding does not perform as good as conventional convolutions for spherical data. For fully self-attention models, the proposed absolute spherical encoding yields better result over traditional absolute and relative encoding. The conventional convolution also benefits from absolute spherical encoding, and further improvements can be achieved by adopting higher order spherical encoding.

6.3 Identity-preserving action recognition

6.3.1 Dataset

SBU Kinect Interaction dataset [44] is a two-person interaction dataset for video-based action recognition. Samples from the dataset are given in Fig. 7. The dataset is composed of 21 sets, with each set containing two actors randomly selected from a total of 7 participants. Following Wu's setting [42], the different sets containing the same actor pairs are combined into a single class. For example, in set 1, actor A is acting while actor B is reacting; in set 4, actor B is acting while actor A is reacting. As set 1 and set 4 have the same actors, they are combined into a single class. This leads to 13 classes of actor pairs. The goal of this task is to predict actions from 8 action categories, and the sensitive attribute to protect is the actor identity from 13 actor pair categories.

Following Wu's setting [42], we divide the total 382 videos into 300/46/36 training/validation/testing sets, relatively. The 300 training videos are used both as D_{train} and D_{adv} to train the privatizer and attacker; the 46 validation videos are used

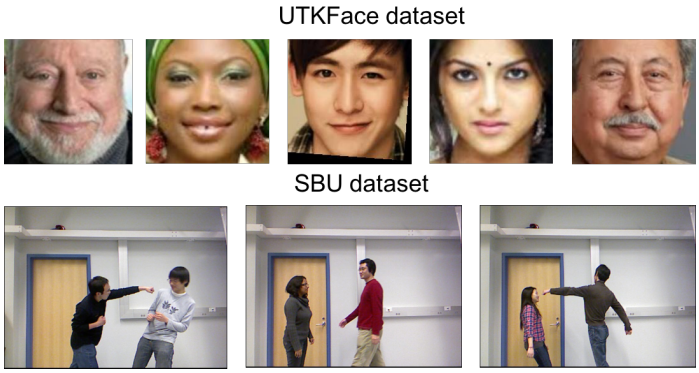

Figure 7: Examples of SBU Kinect Interaction dataset.

to tune hyper-parameters; while the 36 testing videos are used to evaluate the action recognition and actor pair recognition performances.

6.3.2 Experiment

We compare PPCON with four baseline methods on identity-preserving action recognition following Wu's setting [42]:

- **Crop Face** defines faces as the private content, and detects and crops out faces in each frames.
- **Crop Body** defines actors as the private content, and detects and crops out the whole actor bodies in each frame.
- **Downsample** performs down-sampling in each frame to decrease its resolution for privacy protection, while utilizes the temporal information between frames for action recognition.
- **Wu et al. ADV - with/without restarting** is the adversarial training method proposed by Wu et al. [42]. The ensemble and restarting mechanisms are added to adversarial training to enhance the generalization of the privatizer. We compare PPCON with their method under two settings: The ADV equipped with ensemble and restarting, and the ADV equipped with ensemble but without restarting.

Training detail: Following Wu's setting [42], We use the C3D network [36] as the action recognition model. The privatizer is adopted with the image transformation network in Li et al. [16] work. For the f_{EST}, we choose the MobileNet architecture, which is utilized as the adversary in Wu's work. The implementation detail is described in the appendix.

First, we initialize the parameters of the privatizer, action recognition model, and f_{EST}. For fair comparison, we use the same parameter initialization procedure as in Wu [42] et. al.'s work:

- For the privatizer, following the settings in Li et. al.'s work [16], we set the dimension of the input data and obfuscated data to be identical. This allows us to initialize the parameters with an aim for the privatizer to reconstruct the original video.
- For the service, a pre-trained C3D network is concatenated with the privatizer, which are jointly trained on the SBU dataset with an aim for the service C3D network to perform action recognition.
- For the estimator f_{EST}, we first freeze the currently trained privatizer and extract the obfuscated data accordingly, by which the estimator f_{EST} is trained with an aim to maximize the Kantorovich-Rubensted dual (cf.(17)) as an estimate of the Wesserstein distance.

Following Wu's setting [42], we apply ten CNNs including MobileNet, ResNet, and Inception for attribute inference attacks (see appendix for more details). In these CNNs, 8 models start from ImageNet pre-trained versions, whereas 2 models are trained from scratch to exclude the relevance between initialization and privacy prediction.

Result and Analysis: The experimental results of our proposed methods and other baseline methods are summarized in Fig. 8, showing the trade-off between the accuracy of action classification and identity classification. Note that a desirable trade-off should achieve maximal action recognition accuracy and minimal identity recognition accuracy, so a point closer to the top-left corner corresponds to a privatizer with better performance.

For the crop face method, the service can recognize actions with high accuracy, but the adversary can also recognize the identity with high accuracy as well. On the other hand, for the crop body method, it protects privacy well but sacrifices the action classification accuracy. Fig. 9 indicates that when faces are cropped out, these identities can still be recognized by other features such as clothing. However, cropping out the whole body increases the difficulty in action recognition. Thus, the drawback of these methods is the dilemma between what should (or should not) be removed.

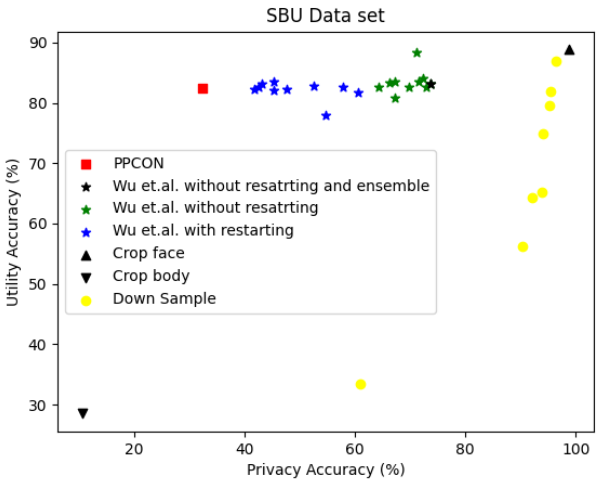

Figure 8: Utility and privacy protection trade-off of identity-preserving action recognition. Both PPCON and Wu's method choose λ=2, and multiple points in Wu's method indicate the various ensemble methods applied by the adversary. Except for PPCON, all results were reported in Wu et. al.'s work [42].

Compared to crop face and crop body methods, the downsample method has an adjustable parameter, namely the down sampling rates. However, as shown in Fig. 9, the identities of the actors are still recognizable even with the images being blurred. In fact, down-sampling yields the worst tradeoff between privacy protection and utility, as shown in Fig. 8. The ineffectiveness of down-sampling towards privacy preserving may be attributed to the fact that down-sampling obfuscates images indifferently rather than considering the specific features pertaining to action recognition and/or identity preservation.

In contrast, PPCON and Wu's methods are both data-driven, and yield better utility task performance and privacy protection tradeoff than the crop face, crop body, and downsample methods, as illustrated in Fig. 8. Wu's [42] results demonstrate that ensemble and restarting mechanisms do improve privacy protection (by comparing the green and blue stars with the black star in Fig. 8). However, the computation of multiple adversaries in the ensemble method requires much computation resource. Furthermore, it would be burdensome and rather ad-hoc to decide how many and which architectures to choose for the adversary.

In contrast, PPCON achieves the best privacy protection as well as maintaining its utility in Fig. 8, with far less parameters as required by Wu's method. To see this, in TABLE 3, we list the number of parameters needed for PPCON and Wu's method. In Wu's method, the total number of parameters increases linearly with the number of adversaries, and the best privacy protection is achieved at the price of the largest amount of parameters (i.e., M=18, the far left blue point in Fig. 8). On the contrary, PPCON is capable of achieving better privacy protection with only one-fifth of the parameters as required by Wu's method.

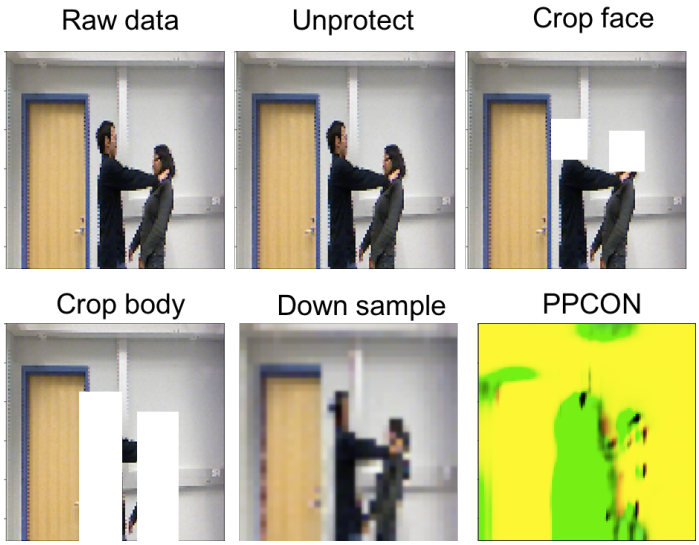

Figure 9: Visualization of obfuscated data. In PPCON, we normalize the output value of privatizer to [0,255] to visualize the obfuscated image.

PPCON						
Privatizer	Service	Estimator				
1,679,241	79,991,015	4,231,976				
Wu et al. method [42]						
Privatizer	Service	Adversary				
1,679,241	79,991,015	M=1	M=2	M=4	M=6	M=8
		1,783,497	3,467,251	6,561,251	9,308,037	11,716,933
		M=10	M=12	M=14	M=16	M=18
		13,815,875	15,619,215	17,152,172	18,429,280	19,475,696

Table 3: Comparison of Parameter usage. Here M denotes the number of adversaries.

6.4 MPC Implementation for Deep Learning

Since PPDNN is developted on top of ABY library, we focus on resolving existing bugs as well as improving the computation cost pertaining to ABY library.

6.4.1 Fixing the Multiplication Triple (MT) Error

It is reported in ABY GitHub Repo. Open Issue#114 [13] that, during the generation of multiplication triple (MT), there is a (roughly 13%) chance that the generated MT is erroneous. After carefully investigating the code and numerous trials and errors, we identify the root cause of the bug is that the original code may terminate the communiation prematurally, rendering the receiver being kicked out of the communication before receiving the complete message. This also supports our observation that the reported MT error always occur during the very last transmission. The root cause of the bug as well as our hotfix is illustrated in Figure.10.

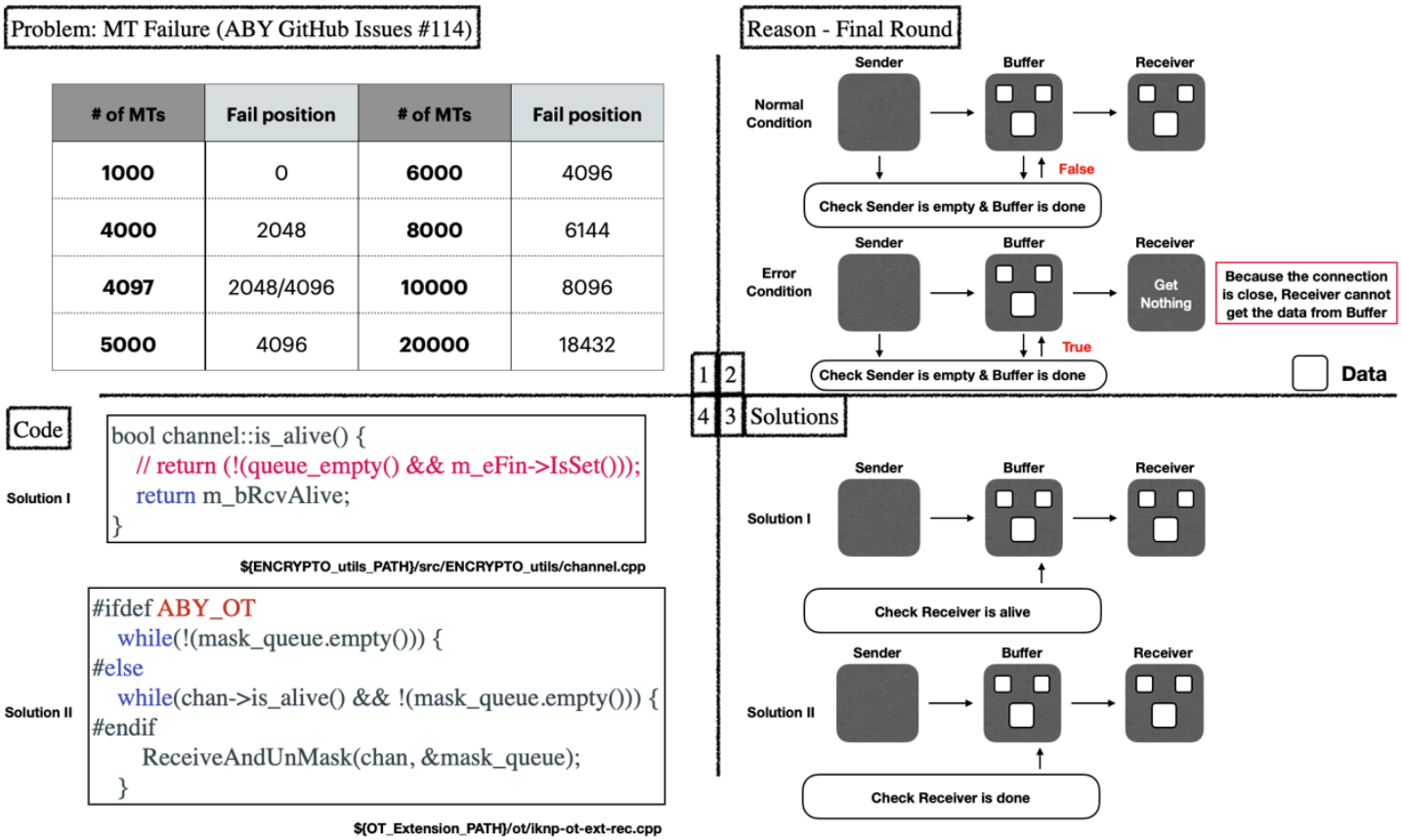

Figure 10: Multiplication triple failure root cause and hotfix.

6.4.2 Integrate ReLU gate with CrypTFlow2 implementation within ABY framework

The nonlinear activation is omnipresent in neural networks. However, the computation performance of its MPC counterpart still has large room for improvement. In ABY framework the ReLU is often implemented with multiple binary logic gates. We implement an ABY-compatiable ReLU gate based on the Millionaires and DReLU implementation in CrypTFlow2 [28] SCI library.

Figure.11 compares the ReLU implementations based on ABY versus SCI libraries, where the experiments are conducted on machines with spec: 3.2GHz 4 cores CPU / 32GB RAM / 492.5MBps LAN / Linux Ubuntu 20.04. We observe that ReLU implemented with SCI library costs a constant $0.2s$ for circuit establishment regardless of the number of ReLU gates. In comparison, the circuit establishment cost for binary-gate ReLU implementation increases significantly when more ReLU gates are involved, as it requires more logic gates as the number of ReLU gates increases. It is evident that integrating ReLU implemented with SCI library into ABY framework yields significant computation cost improvement especially in the case where the network contains a large amount of ReLU gates.

In view of the fact that SCI library adopts multithread functionality towards speedup, Figure.12 compares the computation cost of ReLU operation in SCI library under multiple threads. We observe that in large scale computation that involves more than 500 thousand ReLU gates, the more threads (say 4 threads) indeed yields the better computation cost; while in small scale computation that involves less ReLU gates, the multithread shows no benefit as it requires additional cost to split data.

6.4.3 Bitwise correct MPC multiplication truncation

As MPC often adopts fixed-point arithmetic, truncations are needed after multiplications to retain numerical precision. However, as illustrated in Figure.13, duplicated bit carrying may randomly occur during MPC secret sharing unless special care is taken for fixed-point trunction. We integrate the truncation protocol in CrypTFlow2 and SIRNN [27] into ABY framework, where the bit-carry information is acquired through MPC. We thus achieve bitwise correct computation identical with plaintext fixed-point multiplications. This allows us to directly verify the MPC inference results by comparing with plaintext inference results, which allows for more straightforward debugging in MPC implementation.

Our implemented PPDNN is tested on MobileNetv2 [33]. As illustrated in Figure.14, through integrating ReLU with SCI library implementation into the ABY framework, the cost for GT establishment and communication is significantly reduced, leading to 4x speedup. By paying a moderate cost in MPC trunction, we gain bitwise correct MPC inference results. Summing everything together, we gain both improvement in computation time as well as bitwise correctness.

7 Conclusion

In this project, we proposed PPCON as a privacy preserving machine learning model that incorporates both WGAN and the idea of class overlapping. PPCON is shown to achieve better utility-privacy trade-off in race-preserving gender classification and identity-preserving action recognition problems. In identity-preserving action recognition problem, we demonstrated that PPCON not only achieves better utility-privacy trade-off, but also with far less parameters as compared to the state-of-the-art method proposed by Wu et al. [42]. This may greatly alleviate the computation burden in video-based privacy preserving machine learning applications.

We also propose Spherical Encoding for omnidirectional images, and compare it with self-attention models and convolution models on omnidirectional image dataset. Spherical encoding preserves spatial information on the sphere, and can be easily adapted to both convolution and self-attention schemes in deep learning models. Experiments show that both conventional convolution and self-attention models benefit from spherical encoding on classification tasks. For deeper models, spherical encoding can be integrated with residual module, leading to state-of-the-art performance.

To further explore the privacy enhancement techniques for deep learning in the cryptographic front, in collaboration with ITRI and NCKU, we developed PPDNN as a DNN-oriented MPC library. We ensure the correctness of MPC inference result through fixing a serious and well-known multiplication triple failure issue. Through integrating ReLU with SCI library implementation into the ABY framework, our developed ABY-based PPDNN enjoys speedup in ReLU operations that comes from SCI implementation. Finally, we achieve bitwise correct MPC inference through integration of MPC truncation protocol in CrypTFlow2 and SIRNN. Experiment on MobileNetV2 shows the effectiveness of our implementation improvement.

References

[1] `https://cybernews.com/privacy/no-gmail-isnt-private-at-all-but-you-can-fix-that/`,

[2] `https://www.theguardian.com/news/2018/may/06/cambridge-analytica-how-turn-clicks-into-votes-christopher-wy`

[3] Mohammad Al-Rubaie and J Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

	ReLU based on CrypTFlow2 SCI Library				ReLU realized with multiple binary logic gates	
	Thread = 1	Thread = 2	Thread = 3	Thread = 4	party - 0	party - 1
Num_ReLUs	Comp. cost w/t circuit establishment and communiation / Comp. cost of purely ReLU operations (sec)					
1,000	0.28 / 0.08	0.34 / 0.14	0.35 / 0.15	0.36 / 0.16	0.23 / 0.005	0.21 / 0.007
10,000	0.30 / 0.10	0.34 / 0.15	0.37 / 0.17	0.39 / 0.19	0.46 / 0.04	0.41 / 0.07
20,000	0.35 / 0.15	0.38 / 0.18	0.40 / 0.20	0.42 / 0.22	0.67 / 0.08	0.53 / 0.17
30,000	0.39 / 0.19	0.41 / 0.21	0.45 / 0.25	0.46 / 0.26	0.93 / 0.12	0.67 / 0.24
40,000	0.45 / 0.25	0.45 / 0.25	0.49 / 0.29	0.50 / 0.30	1.16 / 0.15	0.86 / 0.32
50,000	0.51 / 0.31	0.51 / 0.31	0.55 / 0.35	0.55 / 0.34	1.42 / 0.23	1.03 / 0.45
60,000	0.56 / 0.36	0.56 / 0.36	0.57 / 0.37	0.56 / 0.36	1.66 / 0.28	1.14 / 0.47
70,000	0.61 / 0.41	0.58 / 0.38	0.62 / 0.42	0.62 / 0.42	2.02 / 0.32	1.29 / 0.53
80,000	0.65 / 0.45	0.61 / 0.41	0.66 / 0.46	0.64 / 0.44	2.14 / 0.38	1.46 / 0.67
90,000	0.72 / 0.52	0.67 / 0.47	0.70 / 0.50	0.68 / 0.48	2.45 / 0.44	1.68 / 0.76
100,000	0.77 / 0.56	0.71 / 0.50	0.75 / 0.55	0.73 / 0.53	2.71 / 0.51	1.82 / 0.85
200,000	1.28 / 1.08	1.08 / 0.88	1.17 / 0.97	1.10 / 0.90	5.18 / 0.91	3.7 / 1.82
500,000	2.86 / 2.66	2.30 / 2.10	2.52 / 2.32	2.30 / 2.10	12.61 / 2.38	8.65 / 4.51
600,000	3.32 / 3.12	2.77 / 2.57	2.92 / 2.72	2.67 / 2.47	15.70 / 3.92	11.77 / 6.37
700,000	3.84 / 3.64	3.22 / 3.02	3.39 / 3.19	3.06 / 2.86	17.56 / 3.31	11.32 / 5.83
800,000	4.36 / 4.15	3.53 / 3.33	3.83 / 3.63	3.51 / 3.31	21.25 / 5.72	14.27 / 7.84
900,000	4.87 / 4.67	4.01 / 3.82	4.36 / 4.16	3.97 / 3.77	24.87 / 6.32	15.88 / 8.53
1,000,000	5.53 / 5.32	4.38 / 4.18	4.79 / 4.59	4.29 / 4.08	26.75 / 7.02	18.52 / 10.21

Figure 11: Comparison between ReLU implementations based on ABY versus SCI libraries.

Num_relus	Thread = 1	Thread = 2	Thread = 3	Thread = 4
1,000	0.09	0.13	0.15	0.16
10,000	0.10	0.15	0.17	0.19
50,000	0.30	0.31	0.34	0.33
100,000	0.55	0.50	0.56	0.52
200,000	1.09	0.88	0.98	0.90
500,000	2.60	2.04	2.27	2.04
600,000	3.09	2.51	2.70	2.40
700,000	3.60	2.92	3.15	2.79
800,000	4.12	3.27	3.60	3.26
900,000	4.60	3.80	4.10	3.71
1,000,000	5.15	4.14	4.56	4.02

Figure 12: Comparison of ReLU with SCI implementation over multiple threads.

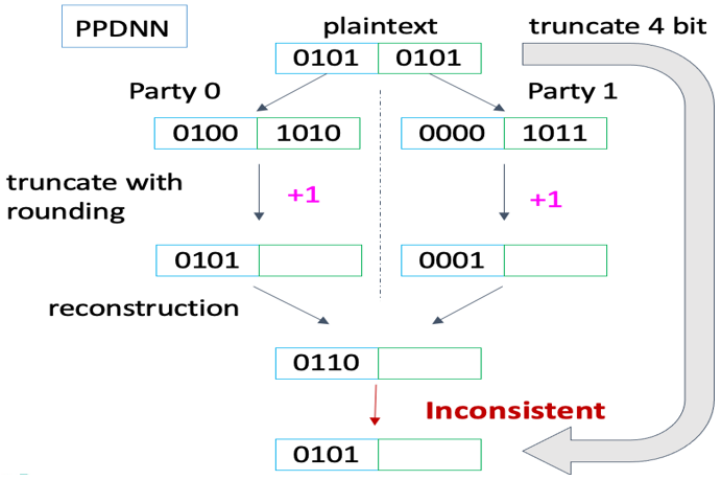

Figure 13: Duplicated bit-carrying issue in secret sharing truncation.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[5] Nishanth Chandran, Divya Gupta, Aseem Rastogi, Rahul Sharma, and Shardul Tripathi. Ezpc: Programmable, efficient, and scalable secure two-party computation for machine learning. In *IEEE European Symposium on Security and Privacy*. (IEEE EuroS&P 2019), February 2019.

[6] Peng-Wen Chen, Tsung-Shan Yang, Gi-Luen Huang, Chia-Wen Huang, Yu-Chieh Chao, Chien-Hung Lu, and Pei-Yuan Wu. Viewing bias matters in 360° videos visual saliency prediction. *IEEE Access*, 11:46084–46094, 2023.

[7] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[8] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018.

[9] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019.

[10] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Mobilenetv2	PPDNN (before)	PPDNN (after)
Total truncate time(local) Total truncate time(lan)	2.193 s 4.144 s	15.836 s 29.088 s
Truncation result	Right shift (not bitwise correct)	SCI truncate (bitwise correct)
Total clip time(local) Total clip time(lan)	116.71 s 138.78 s	26.044 s 33.519 s
Time Per inference (local) Time Per inference (lan)	429 s 510 s	360 s 425 s

Figure 14: PPDNN testing results on MobileNetV2.

[11] 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.

[12] Craig Gentry and Dan Boneh. *A fully homomorphic encryption scheme*, volume 20. Stanford university Stanford, 2009.

[13] GitHub. Erroneous multiplication results #114.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Po-Hsuan Huang, Chia-Heng Tu, Shen-Ming Chung, Pei-Yuan Wu, Tung-Lin Tsai, Yi-An Lin, Chun-Yi Dai, and Tzu-Yi Liao. Securetvm: A tvm-based compiler framework for selective privacy-preserving neural inference. *ACM Trans. Des. Autom. Electron. Syst.*, 28(4), may 2023.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[17] Yitong Li, David E Carlson, et al. Extracting relationships by multi-domain matching. In *Advances in Neural Information Processing Systems*, pages 6798–6809, 2018.

[18] Tsung-Hsien Lin, Ying-Shuo Lee, Fu-Chieh Chang, J. Morris Chang, and Pei-Yuan Wu. Protecting sensitive attributes by adversarial training through class-overlapping techniques. *IEEE Transactions on Information Forensics and Security*, 18:1283–1294, 2023.

[19] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 619–631, New York, NY, USA, 2017. Association for Computing Machinery.

[20] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. 2019.

[21] Payman Mohassel and Peter Rindal. Aby3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 35–52, 2018.

[22] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38, 2017.

[23] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

[25] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019.

[26] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[27] Deevashwer Rathee, Mayank Rathee, Rahul Kranti Kiran Goli, Divya Gupta, Rahul Sharma, Nishanth Chandran, and Aseem Rastogi. Sirnn: A math library for secure rnn inference. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1003–1020, 2021.

[28] Deevashwer Rathee, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow2: Practical 2-party secure inference. In *27th Annual Conference on Computer and Communications Security (ACM CCS 2020)*. ACM, August 2020.

[29] M. Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin Lauter, and Farinaz Koushanfar. XONN: XNOR-based oblivious deep neural network inference. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1501–1518, Santa Clara, CA, aug 2019. USENIX Association.

[30] M. Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M. Songhori, Thomas Schneider, and Farinaz Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. In *ASIACCS*. ACM, ACM, 03/2018 2018.

[31] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902. IEEE, 2015.

[32] Michael S Ryoo, Brandon Rothrock, Charles Fleming, and Hyun Jong Yang. Privacy-preserving human activity recognition from extreme low resolution. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[34] 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.

[35] Congzheng Song and Vitaly Shmatikov. Overlearning reveals sensitive attributes. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[36] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[37] Ardhendu Tripathy, Ye Wang, and Prakash Ishwar. Privacy-preserving adversarial networks. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 495–505. IEEE, 2019.

[38] Tsai Tung-Lin and Wu Pei-Yuan. Sepmm: A general matrix multiplication optimization approach for privacy-preserving machine learning. In *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–10, 2023.

[39] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[40] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[41] Sameer Wagh, Divya Gupta, and Nishanth Chandran. Securenn: Efficient and private neural network training. Cryptology ePrint Archive, Paper 2018/442, 2018. `https://eprint.iacr.org/2018/442`.

[42] Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Towards privacy-preserving visual recognition via adversarial training: A pilot study. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–624, 2018.

[43] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[44] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012.

[45] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020.