Distributed and Private Computation for Inference

by

Abhishek Singh

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....
Program in Media Arts and Sciences, School of Architecture and Planning
May 21, 2021

Certified by .....
Ramesh Raskar
Associate Professor
Thesis Supervisor

Accepted by .....
Tod Machover
Academic Head, Program in Media Arts and Sciences

Distributed and Private Computation for Inference

by

Abhishek Singh

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on May 21, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Recent progress in mobile and cloud computing coupled with the increase in data has resulted in a data-driven ecosystem that is making an impact in several domains of science and engineering. However, this data-driven ecosystem lacks protective measures for privacy resulting in regulations and behaviors that restrict data sharing. Augmenting the existing data-driven ecosystem with privacy preserving solutions could unlock the access to data silos, increasing the impact manifold. In this thesis, I discuss and identify gaps in some of the existing works and develop privacy preserving mechanisms for data analysis and distributed computation. At an abstract level, existing work in this domain includes federated learning, differential privacy, and encrypted computations. I describe the practical scenarios where all these approaches do not suffice due to their intrinsic computation infeasibility or suboptimal privacy-utility trade-off. This work augments such existing approaches by improving certain trade-offs and utilizing priors specific to the problem.

Thesis Supervisor: Ramesh Raskar
Title: Associate Professor

Distributed and Private Computation for Inference

by

Abhishek Singh

This thesis has been reviewed and approved by the following committee members

Ramesh Raskar

……

Associate Professor of Media Arts and Sciences
MIT Media Lab

Alex “Sandy” P. Pentland

……

Toshiba Professor of Media Arts and Sciences
MIT Media Lab

Rosalind W. Picard

……

Professor of Media Arts and Sciences
MIT Media Lab

Acknowledgments

First and foremost I am extremely grateful to my advisor Prof. Ramesh Raskar for his guidance and constant support throughout these two years. His mentorship has helped me grow as a person both professionally and personally. I would also like to thank my thesis readers, Prof. Rosalind Picard and Prof. Alex P. Pentland. Their feedback helped me in defining the scope of this thesis and finish it timely. Prof. Picard's meticulous review helped me tremendously improve this thesis.

I was fortunate to have collaborated and worked alongside many smart and talented people. Subhash Sadhu is the person with whom I spent most of my time at the lab and had conversations about almost all aspects of our lives. Ayush Chopra has been a collaborator and a friend with whom I do research, travel, entertainment, and even work out sometimes. With him, there are so many things to be thankful for. I look forward to our continued camaraderie. I thank Praneeth Vepakomma for our collaborations and mentorship. After my advisor, he was the person who guided me and helped me with feedback when I joined MIT. I am thankful to Vivek Sharma for our collaboration, fun memories, and long discussions of our social lives. All of the research presented in this thesis would not be possible without the help of my collaborators Prof. Anna Lysyanskaya, Dr. Antigoni Polychroniadou, Emily Zhang, Leo de Castro, and Ethan Garza. I am also thankful to my group members Connor Henley, Tristan Swedish, and Tomohiro Maeda. While we never got a chance to collaborate actively, I have learned a lot by seeing them excelling in their respective research areas. I am deeply grateful to Utkarsh Sarawgi for many of our fond memories made in the last two years in the greater Boston area. I would also like to thank Debojyoti Dutta for his guidance on navigating academic life and helping me in seeing the bigger picture.

Roughly 75% of this thesis's work was done during the COVID-19 pandemic. While isolation tested my composure, I was blessed to have many special people in my life who kept me going. My parents, Neelima Singh and Vijay Singh have always bestowed their endless love and care for me. I am immensely thankful to my brother Prateek

Singh and extended family for always being my support. Monthly calls with my friends Vishal Singh, Chandrakant Ojha, Chris Andrew, Tushar Maheshwari, Akash Das, Shyam Nandan Rai, Ashutosh Mishra, and many others was something I always enjoyed the most. For my daily sanity, I am deeply grateful to the online lectures by Dr. Jordan Peterson and meditation app Headspace.

While the pandemic disrupted the lifestyle of all of us, I was lucky to find a strong purpose in joining a fight against the pandemic through a project started by my advisor. Within two months, this project grew from few individuals to more than 2000 volunteer-driven non-profit foundation Pathcheck. The Pathcheck family is very close to my heart and it's awe-inspiring to see passionate people there taking time out of their daily lives to reduce the suffering in this world. Finally, I would like to express my deep gratitude to Media Lab and MIT for giving me a platform and support to explore my intellectual curiosity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recent progress in data driven computational techniques have shown tremendous promise in almost every aspect of life ranging from different domains of science [176, 153, 60, 17] to humanities [127, 26, 83, 131]. However, a majority of these data driven techniques rest on two interlinked assumptions - 1) data can be centralized for the analysis, and 2) data is not private. There could be instances where assumption 1) is true while 2) is not and vice-versa. A majority of the recent works in federated learning and split learning have focused on relaxing these assumptions to enable distributed and private training of neural networks. However, a lot of practical applications such as prediction, inference, and data release have received relatively less attention. The focus of this thesis is on using distributed computation to enable these three problems while preserving privacy.

As a motivating example, let us consider building a machine learning based diagnosis system for COVID-19 using chest X-rays. For building such a system, we need to perform data collection to train any machine learning model. Due to privacy concerns associated with medical data, it would be desirable to remove any sensitive information from the x-ray image before it is shared with an untrusted party building the diagnosis system. We address a similar problem in data release. Once the model is trained, it can be deployed on a server for production. However, hospitals can

not share the data during prediction either and hence we require privacy preserving prediction mechanisms. Finally, given a chest x-ray, a radiologist might be interested to know how many other patients globally have a similar infection. The radiologist can not share their own data with others and can not request other's data either due to privacy regulations. This problem can be posed as private set intersection which I attempt to address in this thesis.
There are three main questions in this thesis which I attempt to delineate and propose solutions based on different trade-offs.

1. How can users protect privacy when using machine learning models during the prediction phase?

2. How can individuals crowd-source data without revealing their sensitive information present in the data?

3. How can users compare their private data with other groups of users without leaking any data to either of the parties?

Here, I would like to strongly emphasize the fact that all these three questions have been well studied and the focus of my thesis is to improve different trade-offs currently achieved by the existing algorithms by proposing new algorithms and protocols. Some of the content of this thesis has been presented at:

- Singh, Abhishek, Ayush Chopra, Vivek Sharma, Ethan Garza, Emily Zhang, Praneeth Vepakomma, and Ramesh Raskar. "DISCO: Dynamic and Invariant Sensitive Channel Obfuscation for deep neural networks." CVPR 2021.

- Singh, Abhishek, Vivek Sharma, Ayush Chopra, Ethan Z. Garza, Praneeth Vepakomma, and Ramesh Raskar. "Dynamic Channel Pruning for Privacy." NeurIPS PPML Workshop 2020

1.2 Prelude to data privacy

While the seeds of privacy can be traced (long) back to Greek philosophers as discussed in Holvast [76], the first published article seems to be from the 1890s where Warren and Brendeis published "The right to privacy" [175], they defined privacy as "*to be let alone*". Privacy, in general, has a long history but data privacy appears to be introduced in 1965 by Warner [174] in which he proposes the idea of a randomized response scheme for collecting survey data with random noise to reduce the bias (that could arise due to privacy reasons) from the survey responders. This work spurred further analysis of the randomized response and its statistical analysis for different kinds of databases and distributions. Adam and Worthmann [7] review the data privacy methods of that era. While these lines of work focused on performing a query on a private database, Rivest et al. [149] introduced the idea of performing arithmetic operations on encrypted data in 1978, which later came to be known as homomorphic encryption. With the inclusion of personal identifiers in the databases, the notion of anonymity emerged as a way to provide confidentiality, where most of the methods were based on the idea of cell suppression [37, 35]. In 1998, Samarati and Sweeney [152] introduced k-anonymity which formalized the notion of suppression for guaranteeing anonymity. The idea of k-anonymity was further improved by ℓ-diversity [115] and t-closeness [105]. Ganta et al. [64] show that anonymity based methods are not sufficient to provide privacy for individual records in a database. Since then a relatively stronger definition of privacy like differential privacy [48] has taken precedence. In this thesis, I define data privacy in different ways based on context of the problem in the upcoming chapters.

1.3 Distributed computing and Privacy entanglement

A majority of the privacy preserving algorithms arise from the fact that we want to perform computation across data from many different people and sources. Many distributed computing algorithms have been developed to address the concerns around

performing computation on private data. Therefore, distributed computing and privacy preserving algorithms seem to be dovetailed in today's era where private data is ever increasing. Security has always been a special topic in distributed computing and privacy can be seen as a sub-topic in the domain of information security. For the scope of this thesis, I will be mostly talking about distributed machine learning (ML) instead of covering the broad spectrum of distributed computing. The first and third chapters of this thesis deal with the instances where distributed computing is introduced to enforce privacy while the second chapter introduces a privacy preserving mechanism for enabling distributed computing algorithms to be run on private data.

1.4 Why Inference

Statistical inference focuses on the estimation of unobserved random variables (or their statistics) based on random samples from a population. For the scope of this thesis, we consider inference for predictive problems as commonly used in the machine learning community. This broadly covers the three problems described in the Section 1.1. Some examples include predicting attributes of an image given a trained model, finding the intersection between two sets, etc. A majority of the work in the ML community that considers distributed computing and privacy [3, 121, 70, 134] focuses on the training aspects of learning, and therefore in this thesis I shift the focus towards the problems which are encountered before and after the training. In the first chapter, we focus on *after* as we discuss how to perform machine learning based prediction for a trained model. In the second chapter, we tackle the *before* aspects of the problem about how to aggregate the data for training a model. Finally, in chapter three we shift the discussion towards general privacy and multi-party computing problem of set intersection that does not require ML.

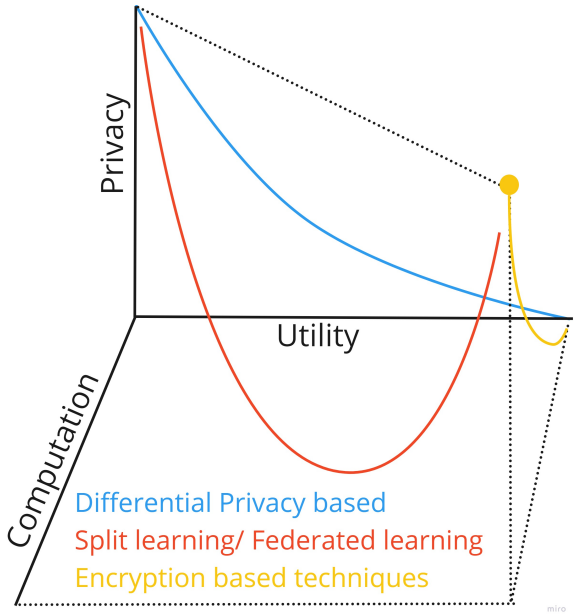

Figure 1-1: Illustration of different trade-offs in the distributed and privacy preserving techniques. Note that the slopes and area under the curve would depend upon a particular algorithm and this diagram only serves the purpose to illustrate what trade-offs are maintained. The focus of this thesis is more around the red curve although I utilize techniques from the blue as well as the yellow curve.

1.5 Trade-Offs

A majority of the work in this thesis is aimed at improving trade-offs therefore it is important to discuss what are the different parameters that can be traded off with each other. Since we focus on distributed computing and privacy, we have four different bases for comparison - 1) privacy, 2) utility, 3) communication, and 4) computation. Out of all four, privacy is a metric which is not trivial to quantify and measure because privacy could mean different things in different contexts and using different threat models. In the privacy community, differential privacy [48] is one of the widely used definition. There are other definitions like α-leakage [107] and Renyi differential privacy [126] that have received interest lately. Utility trade-off is typically measured by reduction in the performance on a task after applying a privacy preserving mechanism. Highest utility for a task can be attained by removing the privacy component, therefore, it is important to quantify the drop in utility. In some instances, it is possible to attain highest utility while preserving privacy by increasing the computation

and communication budget; this is typically the regime where cryptographic methods are used. Figure 1-1 illustrates three different techniques across computation, privacy, and utility axis.

1.6 Key components

A typical privacy preserving mechanism makes use of these three components - transformation, randomness, and invariance. Transformation here refers to a general linear or non-linear transformation; for example, we discuss a mechanism that leverages affine transformation for removing sensitive information from data. Similarly, we will make use of different kinds of cryptographic transformation in chapter three. Randomness is another key component used for building privacy preserving mechanisms. In encryption, the keys are chosen randomly and differential privacy relies on the mechanisms that are typically randomized by the use of random noise, sampling and shuffling. Unlike transformation and randomization which are operations, invariance is a property that is utilized to achieve a good privacy-utility trade-off. In chapter one we show how the utility can be preserved if applying transformation on the sensitive information is invariant to the utility. Similarly in homomorphic encryption based methods, we utilize homomorphisms that make computation possible on encrypted data. As we use these three components in the upcoming chapters, we see that every technique makes use of one more than the others, and the choice of which components to leverage results in corresponding trade-offs.

1.7 Outline

1.7.1 Private Collaborative Inference

In this chapter we address the first question: How can users protect privacy when using machine learning models during the prediction phase?

Recent deep learning models have shown remarkable performance in image classification. While these deep learning systems are getting closer to practical deployment,

the common assumption made about data is that it does not carry any sensitive information. This assumption may not hold for many practical cases, especially in the domain where an individual's personal information is embedded within the data, a common phenomenon in domains like healthcare and facial recognition systems. In this chapter, we posit that selectively removing features in a feature space can protect the sensitive information and provide a better privacy-utility trade-off. Consequently, we propose DISCO which learns a dynamic and data driven pruning filter to selectively obfuscate sensitive information in the feature space. We propose diverse attack schemes for sensitive inputs & attributes and demonstrate the effectiveness of DISCO against state-of-the-art methods through quantitative and qualitative evaluation.

1.7.2 Private Data Release

In this chapter we discuss the second question: How can individuals crowd-source data without revealing their sensitive information present in the data?
A majority of ML models are trained on datasets where one or more data samples are obtained from different users. Currently, these data samples are obtained from users and aggregated for analysis but this would not be feasible if the data contain sensitive information about individuals. Existing works often address this problem by using distributed machine learning methods where each user trains their model locally instead of sharing their raw data. However, we show that privacy preserving data aggregation has a much broader scope than what can be achieved by distributed machine learning techniques. We propose a method for sharing desensitized data by applying a transformation in the feature space such that the shared dataset is useful for learning any arbitrary downstream task. To preserve representations for an arbitrary downstream task, we first disentangle the sensitive set of features from the non-sensitive ones using a variational autoencoder and then perform feature manipulation to anonymize individual linkage with sensitive attributes. We compare our method against a state of the art baseline and demonstrate multiple benefits of the technique empirically.

1.7.3 Private Set Intersection

In this chapter, we shift the focus on the third question: How can users compare their private data with other groups of users without leaking any data to either of the parties?

One of the ways to compare data between individuals is to perform set intersection where the elements of the set are items to be compared. When the elements of the set are private, the problem is called as Private Set Intersection(PSI). PSI is a powerful tool for many applications such as genomic matching, network and contact discovery, and many more [57]. In this work, we focus on contact tracing as a downstream application. Exposure notification in GPS based contact tracing platform requires performing intersection of at least two GPS trails, however, these GPS trails carry sensitive information about a user and hence requires performing intersection in a private manner. Low entropy in the GPS trails makes this problem even more challenging as brute force exposure of identifying information becomes feasible. In this chapter, we discuss the limitations of the existing methods for the use-case of contact tracing and propose a method that utilizes homomorphic encryption and secure system design for performing private GPS trail intersections. We discuss various aspects of its security, threat model and highlight potential attacks that can be performed on the proposed techniques.

Chapter 2

Background

In this chapter, I review the broad set of concepts of methods that will be used or compared within the upcoming chapters. All of the methods and concepts mentioned here serve the purpose of setting up the context under which the contributions of this thesis fall. However, a more detailed discussion of related works is made in the upcoming chapters themselves.

2.1 Machine learning

The main idea in machine learning is to identify patterns in a dataset $X \in \mathbb{R}^{n \times m}$. Here n is the number of data samples in the database and m is the number of dimensions of every data sample. Each data sample $x \in \mathbb{R}^m$ is a vector representing a certain measurement and also referred as features. We can consider the example of X-ray images as seen in chapter 1. The pixel values of the x-ray images can be represented as features. In this case, the goal of a ML algorithm would be to identify likelihood of pneumonia of the individual from the pixel values of their x-ray images. There are different categories of problems in ML and the category broadly determines the input-output space of a ML problem. One common category of problems in ML is called supervised ML where the objective is to learn a mapping from $X \in \mathbb{R}^{n \times m}$ to $Y \in \mathbb{R}^{n \times k}$. In supervised ML, the dataset $\{X, Y\}$ is used for obtaining a function $f : R^m \rightarrow R^k$. When the output is a categorical variable, the problem is called classification and f

is called a classifier. The entries of the categorical variable are referred as classes and defined by experts based on some meaningful concept underlying the corresponding sample's x. Typical examples of supervised ML include regression, decision trees, support vector machines, and neural networks. Other classes of problems include unsupervised learning, reinforcement learning and etc. For a more comprehensive reading in machine learning, we refer the reader to Bishop [19]. In this thesis, I will be building upon several existing works in deep learning and hence we discuss it further.

2.1.1 Deep learning

Deep learning is a special case of neural networks where a large number of layers are used inside the neural networks. As discussed in the X-ray example, the input features are pixel values and large number of layers allow a heirarchical process of the input feature where different patterns can be learned at different layers. Zeiler and Fergus [181] showed this composition of multiple layer can be interpreted as going from pixels to curves and lines to abstract objects. The idea of representing information processing from data through connections between "artificial neurons" is commonly referred as connectionism in cognitive sciences [118]. Deep learning has shown remarkable performance for the problems where datasets are unstructured such as computer vision [170] and natural language processing [163]. For a more in-depth commentary on deep learning, we refer the reader to Goodfellow et al. [68].

2.2 Distributed Machine Learning

Distributed Machine Learning is the frontier of Machine Learning where multiple devices collaborate together to perform machine learning on a given dataset. What constitutes a device can vary significantly based on the setup. In some instances, it is multiple GPUs that are spread across a single machine, multiple machines spread in a cluster, or multiple machines spread across geographic boundaries. There are two important paradigms in distributed machine learning that we will be studying

in more depth - data parallel learning and model parallel learning. The idea behind data parallel learning is to have multiple copies of the same machine learning model to be run in different machines, while model parallel learning requires different parts of the machine learning model to be run across different machines.

2.2.1 Split Learning

Split learning is a technique introduced by Gupta and Raskar [70] that leverages a model parallel approach of distributed machine learning to allow multiple agents to train a neural network in a round robin fashion. From a privacy standpoint, it is required that the representation shared by the agents does not allow inferring sensitive information in the data. From a distributed computing point of view, the traditional split learning architecture requires round-robin scheduling between the clients that prevents it from being asynchronous.

2.2.2 Federated Learning

Federated learning introduced by McMahan et al. [121] is a distributed ML approach where a central server orchestrates training across multiple clients and each client communicates their model update instead of raw data. From a privacy standpoint, the model updates should carry the least amount of information about private data while still carrying information about the underlying data distribution. For a more detailed discussion and open problems in federated learning, we refer the reader to Kairouz et al. [87].

Singh et al. [158] compare the communication efficiency of Split and Federated learning. Their analysis indicates that the *big dataset - small model* regime is favored by Federated learning while the *small dataset - big model* regime is favored by Split learning. Thapa et al. [162] propose a hybrid approach that aims to combine the best of both worlds by enabling federated learning over multiple agents' client model while a split learning server remains intact.

2.3 Data Privacy and Security

Data privacy is a well studied topic with different categories of solutions. At a very high level, its techniques can be segregated into two broad categories - 1) techniques for computation on private data and 2) techniques for private data sharing. Homomorphic encryption, secure multi-party computation, and secure hardware fall in the first category while differential privacy and information obfuscation fall under the second.

2.3.1 Differential Privacy

Differential privacy was introduced by Dwork et al. [48] and is a highly generalizable yet powerful definition of privacy. The definition has two key components - a randomized mechanism and a family of different possible databases. Intuitively, the definition says that a randomized mechanism is differentially private if the output of the mechanism does not change significantly by the presence or absence of any data point from any database from the family of databases. More formally differential privacy can be defined as follows:
A randomized mechanism $\mathcal{M}$ satisfies (ϵ,δ)-differential privacy if for any two neighbouring databases X and X'

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

Here Pr is the probability density function. The original definition by Dwork et al. [48] introduced the definition where $\delta = 0$ and the neighbouring database was defined using an ℓ_1 distance such that $||X - X'|| \leq 1$. For more detailed discussion and analysis, we refer the reader to Dwork and Roth [49].

2.3.2 Homomorphic encryption

Homomorphic encryption is a powerful idea of performing any arbitrary arithmetic operation to be evaluated on encrypted data without requiring the encrypted data to

be decrypted before evaluation. Homomorphic encryption comes in two categories - somewhat homomorphic encryption (SHE) and fully homomorphic encryption (FHE). SHE focuses on homomorphic encryption for a fixed set of arithmetic operations while FHE schemes are built for arbitrary operation evaluation. These operations are represented through circuits in many cases. Typically SHE schemes are faster but limited in their scope and vice-versa for FHE schemes. The idea was first introduced by Rivest et al. [149] and a full practically plausible construction was shown by Gentry [67]. While the earliest proposals suffer from inefficient design, recent works [25, 24, 21] have proposed efficient schemes to make the circuits evaluation practically realizable. For a more comprehensive discussion we refer the reader to a survey by Abbas et al. [6].

Chapter 3

Private Collaborative Inference

3.1 Introduction

While large deep neural networks have resulted in breakthroughs across computer vision [179], speech recognition [12] and reinforcement learning [15] their deployment in critical application domains such as healthcare and face-recognition has motivated a research focus on learning censored, unbiased and fair data representations to mitigate misuse by adversarial agents. Alternately, there can also be *sensitive* information in data which the user would like to keep private but the learned representations may inadvertently encode. This sensitive information may manifest as sensitive inputs or attributes (such as race, age and etc). Consider a setup where citizens consent to usage of face recognition in public spaces for identifying criminals. During inference, feature representations are extracted for faces and identification is performed by matching in the feature space over an indexed database. While this may be a well-intended initiative, a malicious adversary may seek to intercept the feature representations to i) reconstruct the input face image or ii) extract personal attributes such as race, age, gender etc. The citizens did not consent to sharing this sensitive information which could be used to compromise their privacy and in a way that is biased or unfair to them. Exploring methods of improving privacy of the sensitive information (image, race, age, gender etc.) while preserving utility (identifying criminals) is the focus of this work.

Conventionally, research in privacy-aware machine learning has primarily focused on protecting training data from membership inference [156] and model inversion attacks [62], when i) training data is distributed over clients and ii) computation of training the model is out-sourced. For the former, distributed learning techniques such as federated learning [87, 97] and split learning [70, 168] are used, where clients communicate with a centralized server using weights and activations, the latter relies on homomorphic encryption [67, 24] and secure enclaves [182, 61]. Additionally, techniques such as multi-party computation [139, 55] and differential privacy [47, 50, 49, 34] have been employed to improve the privacy in federated-learning. While effective for training, scaling these methods for deployment at inference is a challenge for a variety of reasons. First, in several cases computational limitations and intellectual property considerations limit keeping the entire model on a client device. Secondly, cryptographic methods for training deep networks [74, 86, 129] are computationally very expensive operations which makes deploying models on the server infeasible when working with sensitive data. We posit that **private collaborative inference**, where the inference network is distributed between client devices (client network) and a server (server network) which communicate via the *split activations*, presents a viable alternative. While amenable to scalability, it is important to encode explicit measures of security in the intermediate activations to protect privacy of the sensitive inputs and attributes.

While not motivating private collaborative inference, a few recent works [104, 150, 133, 16, 78] have attempted the related problem of attribute leakage [150, 52, 120, 16, 151] by focusing on adversarial representation learning (ARL). This couples together two entities, i) an adversarial network that seeks to extract a sensitive attribute from a given activation and, ii) a predictor network that intends to extract compact activations for accurate prediction of a task attribute (utility) while preventing the adversary from leaking the sensitive attribute (privacy). To balance this privacy-utility, Roy and Boddeti [150] designed an objective to maximize entropy of the prediction by the adversary network, other techniques include maximizing cross entropy loss [92, 104] to minimize likelihood of the predictor on the sensitive at-

Figure 3-1: a) Input Image and Grad-CAM visualization from ResNet-18 classifier b) Corresponding convolution representations which encode inter-channel redundancy and preserve intra-channel semantic integrity.

tributes.

Motivated by the above observations, in this work, we first examine existing ARL methods which reveal the presence of high redundancy in learned representations. We posit that selectively removing features in this latent space can protect the sensitive information and provide a better privacy-utility trade-off than ARL based techniques. Consequently, we propose DISCO which learns a dynamic and data driven pruning filter to selectively obfuscate sensitive information in the feature space. We validate DISCO and other baseline methods with multiple attacks on inputs and attributes. We observe that DISCO consistently achieves superior performance by disentangling representation learning from privacy using the pruning filter.

To this end, the contributions of this chapter can be summarized as follows:

- We introduce DISCO, a dynamic scheme for obfuscation of sensitive channels to protect sensitive information in collaborative inference. DISCO provides a steerable and transferable privacy-utility trade-off at inference.

- We propose diverse attack schemes for sensitive inputs and attributes and show that DISCO achieves significant performance gains over existing state-of-the-art methods across multiple datasets.

- To encourage rigorous exploration of attack and defense schemes for private collaborative inference, we release a benchmark dataset of **1 million** sensitive representations.

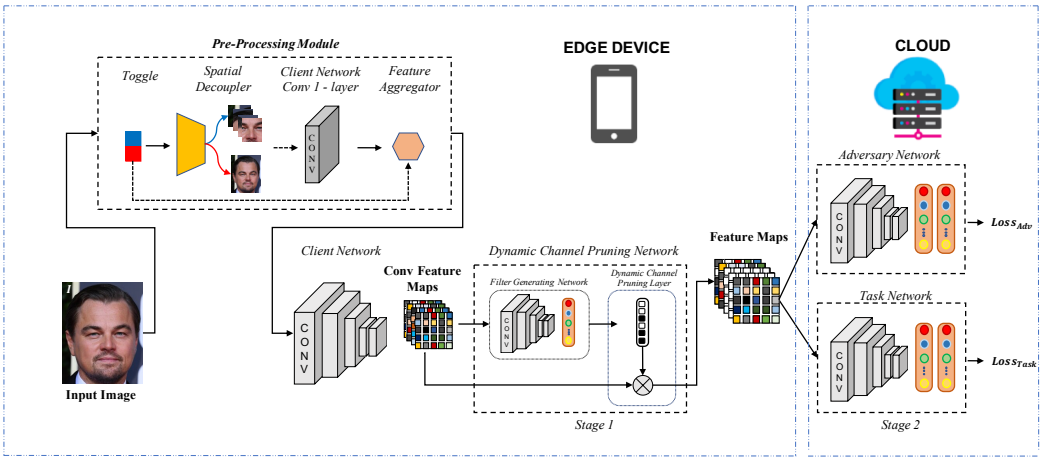

Figure 3-2: **DISCO for Privacy**. Input to the network is an image, as well as task labels and attribute labels to hide. The network is jointly optimized with a task objective to adaptively hide a given attribute without causing a drop in performance of the target task.

3.2 Related Work

Private Representation Learning [97, 70] provides mechanisms that allow for learning on data distributed across multiple agents with raw training data never leaving the corresponding client device. DP-SGD[4] further improves [97] by adding differentially private noise to weights of the trained model to prevent reconstruction of training data by inversion attacks. That said, techniques such as DP-SGD are largely optimized to protect training data. In contrast, there is limited research on methods for privacy during inference via privatized activations. The majority of the work on private inference use ARL [150, 151, 92, 16, 104, 169] to learn a feature extractor that minimizes sensitive information leakage. Bertran et al. [16] apply transformation in the image space to ensure server's input remains an image. Vepakomma et al. [169] introduced a distance correlation based regularization to decouple intermediate activations from input data while preserving performance on task attribute. While efficacy of these methods depend upon the convergence of min-max optimization, our work separates the feature extraction and privatization module giving guaranteed reduction in mutual information. In this work, we explore methods that seek to reduce redundancy and semantic integrity of activations to mitigate attacks on sensitive information.

Natural Pre-Image is a class of diagnostic techniques that are designed to re-

construct an input image from intermediate activation values; it finds utilization in computer vision tasks such as denoising, super-resolution etc. Deep image prior [167] leverages a randomly-initialized neural network and a hand crafted prior to invert deep neural representations and reconstruct the input. Dosovitskiy et al. [44] seeks to train a decoder offline to learn to predict the input distribution. We leverage expected pre-image methods to formalize diverse attack schemes on sensitive inputs.

Bias in Machine Learning is a recent direction of ML research focused on two key problems: identifying and quantifying bias in datasets, and mitigating its harmful effects. The bias routinely manifests as some attributes of the input (eg. age, race, gender for faces). A popular category of techniques involves adversarial representation learning [173, 92, 11] to mitigate the impact of the bias attribute on the task attribute. This family of adversarial mitigation techniques aligns with our work on selective privacy, with the private attribute analogous to the bias attribute, and a corresponding state-of-the-art [92] forms one baseline for our study.

Part-based Representation Learning involves splitting an image into several stripes to learn local representations and has achieved promising performance on computer vision tasks such as person re-identification which involves image retrieval under occlusions and partial observability. While sophisticated learning based partitioning methods have been explored [106, 109, 183, 180], methods proposed in Wang et al. [172] have achieved outstanding performance with trivial deterministic splitting. In this work, we adapt the static part-based techniques to decouple the intra-channel semantic consistency of convolutional activations for improving privacy-utility trade-offs in collaborative inference.

Channel Pruning is a prevalent technique for deep network compression to minimize computational complexity and accelerate inference [33]. While most methods interleave pruning with the training phase [8, 128, 185], there has been recent focus on pruning at inference [65]. Gradual pruning of channels [8] is another method for pruning at fixed intervals during training using a feature relevance score to minimize compute cost. Gao et al. [65] propose dynamic feature boosting and suppression (FBS) to predictively amplify salient convolutional channels and skip unimportant

ones at run-time for accelerated inference. In this chapter, our proposed method can be aligned with channel pruning but optimizes for a different objective of preventing leakage of sensitive information.

Filter Generating Networks (FGN) [84, 82] are special neural network modules that generate filtering scores for the intermediate output of a standard neural network. One such module, the "Spatial Transformer" network, is proposed by Jaderberg et al. [82]. This spatial transformer module applies an affine transformation to feature maps to do translation and rotation for improved classification. Following spatial transformers [82], all these recent works [84, 96, 155] utilize the same concept to learn a steerable filter [84], weather prediction filter [96], an image enhancement filter [155], and a dynamic motion motion representation filter [41] using source-target image pairs. In contrast to these works, our focus is to learn dynamic filters that selectively prune channels which leak sensitive attributes without harming the performance on the target task. The output of our dynamic channel pruning filters are binary (0 or 1) in nature, where 0 masks (or deactivates) channels that contribute to sensitive attributes, and 1 unmasks channels that contribute to the target task at hand.

3.3 Methodology

First, we introduce the attack and threat models and then define the privacy considerations for our work. Finally, we formalize our privacy evaluation setup and delineate our proposed method DISCO: *Dynamic and Invariant Sensitive Channel Obfuscation* for protecting sensitive information in latent representation.

3.3.1 Formulation

Setup. Consider a parameterized model $f(\theta; \cdot)$ trained to estimate the target attribute $y \in \mathcal{Y}_1$ for a given input image $x \in \mathcal{X}$. In many scenarios, x may be a sensitive input or have a sensitive attribute $\hat{y} \in \hat{\mathcal{Y}}$. Consideration for balancing computational feasibility and privacy have motivated private collaborative inference

schemes [150, 104] that split $f(\theta;\cdot)$ into $f_1(\theta_1;\cdot)$ and $f_2(\theta_2;\cdot)$ where:

$$f_1(\theta_1; x) \in F_1 : \mathcal{X} \times \Theta_1 \rightarrow \mathcal{Z}$$

$$f_2(\theta_2; z) \in F_2 : \mathcal{Z} \times \Theta_2 \rightarrow \mathcal{Y}_1$$

such that $f_2 = (\theta_2; f_1(\theta_1; x))$ and $\theta = \{\theta_1, \theta_2\}$. We refer to this as the *traditional* setup for collaborative inference. We formalize f_1 as the *client network* that is executed on a trusted device and f_2 as the *task network* that executes on an untrusted server using the *client activation* $z = f_1(\theta_1; x)$.

Threat Model. Under our threat model, the untrusted server could attempt to learn sensitive information about x by inferring an arbitrary sensitive attribute $\hat{y}$ or by reconstructing x itself. As a concrete example, x may be a face image with y as gender and $\hat{y}$ as racial identity. For the evaluation and algorithm design purposes, we build a proxy adversary that attempts to approximate the real world adversary. This proxy adversary is parameterized with an *adversarial network* $f_3(\theta_3;\cdot)$ that may intercept the payload z to extract the sensitive input x or the attribute $\hat{y}$. **Attack Model.** The adversary may utilize the activation z to perform a *reconstruction attack* to recover the sensitive input or a *leakage attack* to extract the sensitive attribute. We define the following attack models for the sensitive information z:

- *Supervised Decoder:* In this attack setting, the adversary leverages a small number of $(z, \hat{y})$ pairs to train a neural network $\hat{f}(\hat{\theta})$ such that $\hat{y} = \hat{f}(z)$. The practical validity of this attack is in the scenarios where some finite number of pairs $(z, \hat{y})$ is obtained through a malicious or colluding client who is also participating in the collaborative inference setting. This attack scheme is inspired from the feature inversion work in the computer vision community [44, 116, 42]. Another practical scenario for this attack is where the pair $(x, \hat{y})$ from a similar distribution are publicly available; in such a case, the client can train an auto-encoder and use the trained decoder for the attack. This technique can be utilized for both a *reconstruction attack* and a *leakage attack*.

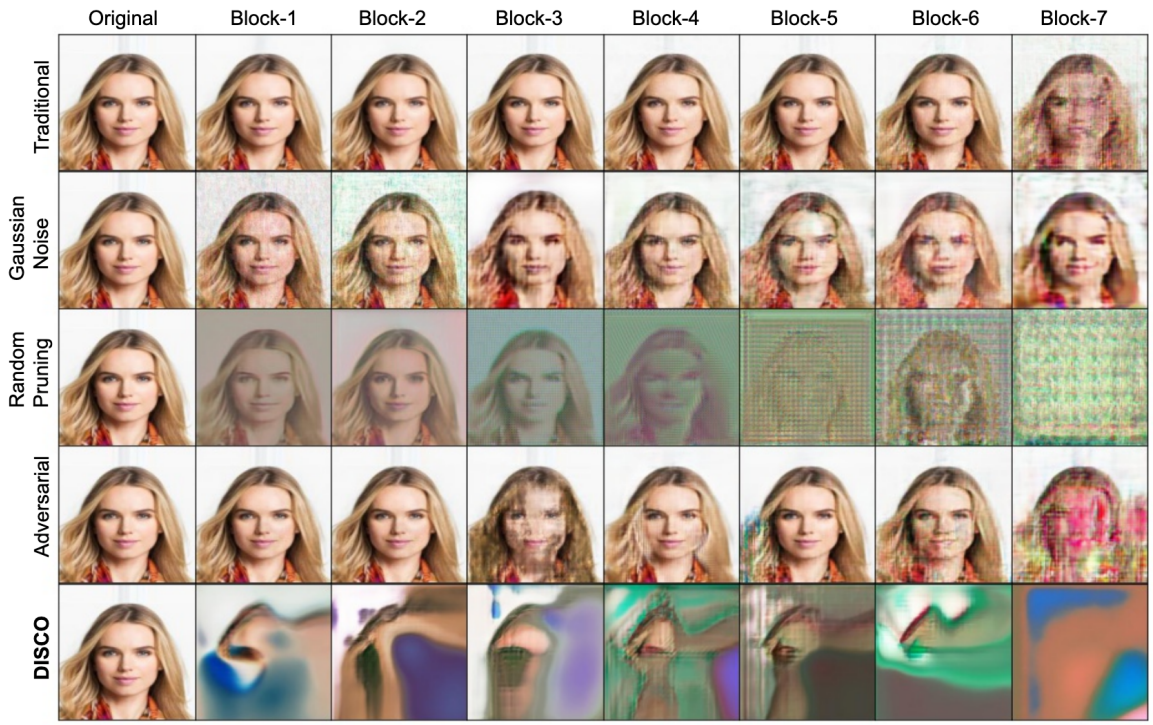

Figure 3-3: **Reconstruction results on CelebA [111]**: All of the reconstructed images are obtained from the activations using the likelihood maximization attack. We generate activations from the ResNet-18 [72] architecture where a set of convolution, batch normalization, and activation layers are grouped under a block. The first column shows the original sensitive input and remaining columns show its reconstruction across different blocks. For gaussian noise we use $\mu = -1., \sigma = 400$, this is the amount of noise at which the learning network gets utility close down to random chance. *Adversarial* refers to the set of techniques for filtering sensitive information using adversarial learning [104, 92]. For *DISCO* and *Random Pruning* we use a pruning ratio of $R = 0.6$.

- *Likelihood Maximization:* Unlike the above scheme, here $(z, \hat{y})$ pairs are not needed to reconstruct the sensitive input, instead, the attacker uses the weights θ_1 of the *client network* and a randomly initialized network $\hat{f}(\hat{\theta}; \cdot)$ that generates an image $\hat{x}$ to produce $\hat{z} = f_1(\theta_1, \hat{x})$. Then the loss $\ell_2(\hat{z}, z)$ between random and sensitive activation is minimized by optimizing $\hat{\theta}$. This attack scheme is inspired by the deep image prior [167] for feature inversion. This attack is only applicable to the sensitive input protection and not to a sensitive attribute. This attack setting is stronger and harder to defend against because it does not require access to the $(z, \hat{y})$ pairs.

Privacy. Following the setup described in Hamm *et al.* [71], we measure privacy as the expected loss over the estimation of sensitive information by the adversary. This privacy loss L_{priv}, given ℓ_p norm, for an adversary can be stated as:

$$L_{priv}(\cdot) \triangleq E[\ell_p(\hat{f}(z), \hat{y})]$$

Under this definition, releasing sensitive information while preserving privacy manifests as a min-max optimization between the data owner and the attacker. For training the model parameters, we use a proxy adversary from which gradients can be propagated. We formalise our setup as an analogue but relax the non-invertibility assumption made by Hamm *et al.* [71] for the client f_1, following [14], to generalize the attack surface to sensitive inputs.

3.3.2 Premise Validation

Adversarial representation learning (ARL) is the existing state-of-the-art approach for performing private inference [150, 104, 92] on sensitive data. Consider Figure 3-1 which visualizes the face image and the learned client activation in ARL [92]. We note the following observations: a) the learned activations have high inter-channel redundancy, and b) individual feature maps preserve semantic integrity of the input image, especially with shallower client networks. Since gradient attribution in convnets is spatially localized [154], we posit that reducing this inter-channel redundancy and

perturbing the intra-channel integrity of *client activations* can help achieve better privacy-utility trade-offs.

3.3.3 DISCO

We now introduce DISCO, depicted in Figure 3-2, which is composed of three key entities: a client, a predictor, and an adversary. The client transforms the input image to generate *client activations* which are communicated to the predictor for inferring the task attribute but can be collected by an adversary.

a) Client owns the sensitive information. Given an input image $x \in R^{3\times H\times W}$ where H and W are the height and width of the input image x, this entity participates in the collaborative inference and intends to achieve privacy in the *client activations* z it communicates.

Initially, x is passed through the *pre-processing module* where the *spatial decoupler* first decomposes it into d^2 disjoint spatial partitions $P_i \in R^{3\times\hat{H}\times\hat{W}}$ for $i = \{0, 1, 2, ..., d^2\}$ with $\hat{H} = H/d, \hat{W} = W/d$. Next, each of the partitions P_i is resized back to $H \times W$ and passed through a convolutional layer (with F filters) to generate $\hat{P}_i \in R^{F\times H'\times W'}$. Finally, the *feature aggregator* generates an aggregated representation $A \in R^{d^2\times H'\times W'}$ by averaging across channels and re-stacking each $\hat{P}_i$. A is then communicated to the client network. Here, we note that $d^2 = F$ in our pre-processing module so that the spatial decoupler can be easily bypassed (toggled-off) without altering the rest of the network architecture. The underlying idea to use a *pre-processing module* is to ensure pruning of channels in z leads to removal of unique spatial information. If not performed, the redundancy present across channels in z would allow an attacker to recover the full image even from a pruned z.

Next, the *client network* takes A as input and generates an intermediate activation $\hat{z} \in R^{C''\times H''\times W''}$. Finally, the *filter generating network* $g(\phi, \hat{z})$ parameterized by ϕ takes $\hat{z}$ as input and generates a feature map score $F \in R^{C''}$ for each channel in $\hat{z}$. The F channel pruning filters are weakly discretized using a sigmoid with temperature (to avoid introducing discontinuity) and then thresholded to obtain a binary vector b. Then b is multiplied channel wise with $\hat{z}$ to produce a pruned feature volume z,

the *client activation*, with channels leaking the sensitive information masked out (or deactivated) in the latent space. Note that F, the feature map score, is conditioned on $\hat{z}$ (hence x) and is thus generated dynamically at run-time on a per sample basis. A key idea of DISCO is to disentangle representation learning from privacy via the learned pruning filter. The hyper-parameter pruning ratio R governs the number of active channels in the pruning filter and helps regulate the privacy-utility trade-off.

b) Predictor is an untrusted entity that receives the *client activations* (z) and executes the *task network* (f_2) to estimate the task attribute (y). The task network is optimized on the conventional loss function (ℓ_u) used for the task. In this paper we consider image classification as the task and hence use cross entropy loss (ℓ_{cce}).

c) Adversary also receives the client activations (z) and executes the *adversarial network* (f_3) with the intent of extracting *sensitive* information - input or attribute. The adversary performs reconstruction attacks for obtaining the sensitive inputs or attribute leakage attacks to infer sensitive attributes. During the training, we design a proxy adversary that has access to the sensitive inputs (x) and attributes ($\hat{y}$). For reconstruction attacks, the adversarial network is a decoder module optimized using ℓ_1 loss against the input x. For attribute leakage attacks, the adversarial network is a convolutional classifier module optimized using ℓ_{cce} loss against the sensitive attribute $\hat{y}$. The adversary loss can be summarized as :

$$l_a = \begin{cases} \ell_1(f_3(z), x) & mode = SI \\ \ell_{cce}(f_3(z), \hat{y})) & mode = SA \end{cases}$$

where, mode $\in$ [SI, SA] represents an attack on either the sensitive input (SI) or the sensitive attribute (SA). Note that f_3 is a proxy adversary used for training purposes while $\hat{f}$ is the real world adversary which will be used for attack during evaluation.

3.3.4 Training

The utility of the task during inference depends upon parameters θ_1, ϕ, θ_2 learned during the training stage and can be expressed as

$$L_{util}(\theta_1, \phi, \theta_2) \triangleq E[\ell_u(f_2(g(f_1(x; \theta_1); \phi); \theta_2), y)] \tag{3.1}$$

As described previously we use a proxy adversary during the training and evaluation of our setup as described by the evaluation function L_{priv} to train the pruning network.

$$L_{priv}(\theta_1, \phi, \theta_3) \triangleq E[\ell_a(f_3(g(f_1(x; \theta_1); \phi); \theta_3), \hat{y})] \tag{3.2}$$

θ_3 is the parameters for the proxy adversary used during the training and evaluation. ℓ_u and ℓ_a is the loss function used for evaluating utility and privacy respectively. The adversary network and task network have access to supervised data and attempt to minimize their losses L_{util} and L_{priv} respectively. The filter generating network is trained to minimize L_{util} and maximize L_{priv}, simulating an implicit min-max optimization for these two components. The client network parameters are only optimized to minimize L_{util}. We deliberately restrict θ_1 for minimizing L_{util} and do not maximize L_{priv} to ensure that the filter generating network generalizes and does not trivially utilize representations learned by the θ_1. This makes our explicit privatizing module $g(\phi, \cdot)$ one of the big differentiating factors of our work from existing ARL based methods [104, 150, 92]. We posit that this facilitates the filter generating network to specialize at pruning by identifying the privacy leaking channels. This overall objective can be summarized as:

$$\min_{\phi} \left[\max_{\theta_3} -L_{priv}(\theta_1, \phi, \theta_3) + \rho \min_{\theta_1, \theta_2} L_{util}(\theta_1, \phi, \theta_2) \right] \tag{3.3}$$

Here, ρ is chosen as a hyper-parameter to trade-off between accuracy and privacy.

3.3.5 Prediction

During the inference stage, computation for feature extraction $\hat{z} = f_1(x; \theta_1^*)$ and pruning $z = g(\hat{z}; \phi^*; R)$ is performed on the trusted system and z is sent to the untrusted party. The value of pruning ratio R governs the total number of channels to be pruned from z and allows adjusting for the privacy and utility trade-off during runtime.

3.3.6 Generalization

The setup described in the main text is as follows for optimizing the parameters -

$$\min_{\phi} \left[\max_{\theta_3} -L_{priv}(\theta_1, \phi, \theta_3) + \min_{\theta_1}(-\rho \max_{\theta_2} -L_{util}(\theta_1, \phi, \theta_2)) \right] \tag{3.4}$$

where $\theta_1, \phi, \theta_2, \theta_3$ are the parameters of the *filter generating network*, *client network*, and *server network* respectively. Let $\theta_1^*, \phi^*, \theta_2^*, \theta_3^*$ be the solution for the parameters we obtain by minimizing the expected loss. Let $\hat{\theta}_1, \hat{\phi}, \hat{\theta}_2, \hat{\theta}_3$ refers to the empirical minimizer of the above mentioned joint optimization. As noted before, we adapt to the setup described by Hamm [71]. However, a significant difference lies in the fact that θ_1 is not trained to minimize L_{priv} as this is to improve generalization of the ϕ across a different set of θ_1. The remaining parameters remain analogous to the min-max filters described in [71] .Following on that, we describe the joint loss as follows

$$L_J(\theta_1, \phi, \theta_2, \theta_3) = L_{util}(\theta_1, \phi, \theta_2) - \rho L_{priv}(\theta_1, \phi, \theta_3) \tag{3.5}$$

Let D be the original unknown data distribution and S be a set of samples obtained from the true distribution for calculating empirical loss then the empirical and expected loss can be bounded as follows, giving a generalization bound.

$$|E_D(L_J(\theta_1^*, \phi^*, \theta_2^*, \theta_3^*)) - E_S(L_J(\hat{\theta}_1, \hat{\phi}, \hat{\theta}_2, \hat{\theta}_3))| \leq 2 \sup_{\theta_1, \phi, \theta_2, \theta_3} |E_D(L_J(\theta_1, \phi, \theta_2, \theta_3)) - E_S(L_J(\theta_1, \phi, \theta_2, \theta_3))| \tag{3.6}$$

For more details, we refer the reader to the proof of theorem 1 shown in [71]. The equation above gives the bound on generalization error.

3.3.7 Effect of channel pruning on mutual information

We now study the effect of applying channel pruning of activations at the output of the client network with regards to the mutual information between the raw sample and the pruned activations. Inspired by the theoretical analysis in [125], we extend and adapt it to our setup of analyzing the reduction in mutual information between the *sensitive input* and *client activations* upon performing random pruning. We use the superscript notation $f_1^k(\theta_1^k; x)$ to denote the output of k'th layer of client network. We compare this with regards to no pruning and random pruning at the k'th layer of the client network as shown below.

Pre-pruning: The negative of the mutual information between the raw data and the output of 1'st layer prior to applying the pruning is given by

$$\begin{aligned} -\mathcal{I}(x; f_1^1(\theta_1^1; x)) &= -\mathcal{H}(f_1^1(\theta_1^1; x)) - \mathcal{H}(f_1^1(\theta_1^1; x)|x) \\ &= -\mathcal{H}(f_1^1(\theta_1; x)) \end{aligned}$$

as $-\mathcal{H}(f_1^1(\theta_1; x)|x) = 0$, due to $f_1^1(\cdot)$ being a deterministic function. Upon applying the data processing inequality, we have that the mutual information between the output of the k'th layer and the raw data satisfies:

$$\mathcal{I}(x; f_1^k(\theta_1^k; x)) \leq \mathcal{I}(x; f_1^{k-1}(\theta_1^k - 1; x)) \leq \ldots \leq \mathcal{I}(x; f_1^1(\theta_1^1; x)) \tag{3.7}$$

where, we have the following relation $\mathcal{I}(x; f_1^k(\theta_1^k; x)) = \mathcal{H}(f_1^k(\theta_1^k; x))$.

Post-pruning: The mutual information after pruning channels randomly can be represented as a multiplication of the outputs at the k'th layer with a Bernoulli random variable $\mathcal{P}$ as $\mathcal{I}(x; f_1^k(x, \theta_1^k).\mathcal{P})$. In addition to the form of data processing inequality used in analysis of pre-pruning; there is an equivalent form of the classical

data processing inequality given by

$$-\mathcal{I}(x; f_1^k(x, \theta_1^k).\mathcal{P}) \geq -\mathcal{I}(f_1^k(x, \theta_1^k); f_1^k(x, \theta_1^k).\mathcal{P})$$

Upon expanding this upper bound using entropy terms we get

$$\mathcal{I}(x; f_1^k(x, \theta_1^k).\mathcal{P}) \leq \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x)|f_1^k(\theta_1^k; x).\mathcal{P}) \tag{3.8}$$

But $\mathcal{H}(f_1^k(\theta_1^k; x))$ is the mutual information in the case of pre-pruning as analyzed above. Therefore the decrease in information about raw data post-pruning is given by the term $\mathcal{H}(f_1^k(\theta_1^k; x)|f_1^k(\theta_1^k; x).\mathcal{P})$. Upon applying the Bayes rule (for conditional entropy), this term exactly equals:

$$\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) + \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}) \tag{3.9}$$

Since the term $f_1^k(\theta_1^k; x)$ is independent of the random variable $\mathcal{P}$, the above can be further rearranged as

$$\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) + \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(f_1^k(\theta_1^k; x)) - \mathcal{H}(\mathcal{P}) \tag{3.10}$$

which simplifies to $\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) - \mathcal{H}(\mathcal{P})$. As we chose $\mathcal{H}(\mathcal{P})$ to be a Bernoulli random variable; upon considering its success probability to be p (lower-case) and probability of failure to be $q = 1 - p$, we have -$\mathcal{H}(\mathcal{P}) = plog(p) + qlog(q)$. Therefore, upon performing random pruning the decrease in mutual information amounts to

$$\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x)) + plog(p) + qlog(q) \tag{3.11}$$

while the mutual information post-pruning is upper bounded by $\mathcal{H}(f_1^k(\theta_1^k; x).\mathcal{P}|f_1^k(\theta_1^k; x))+$ $plog(p) + qlog(q)$.

3.4 Discussion: Dynamic Design of *DISCO*

A key idea behind DISCO is the decoupling of privacy considerations from representation learning using the dynamic pruning filter. We analyse the dynamic formulation of this design along the following dimensions:

- Dynamic Private Representations: The filter generating network in DISCO estimates the pruning filter for each input, independently at run-time. Since different convolutional filters are known to activate differently [65], the dynamic channel pruning in DISCO enables more personalized identification of sensitive channels for each input resulting in better privacy-utility trade-offs.

- Dynamic Integration: We train DISCO in two phases as i) train the client and the predictor networks to maximize utility ii) train filter generating network with predictor and the (proxy) adversary to minimize privacy leakage and preserve utility. Decoupling of g from f_1 enables private *expert filters* that can obfuscate sensitive attributes and be employed by a network running DISCO. For example, one can build a dictionary of DISCO modules for different sensitive attributes for faces such as race, gender, eyeglasses, and etc. can be trained and used by different vendors based on their context for privacy and utility.

- Dynamic Privacy Utility Trade-offs: All previous methods weight seek to balance privacy-utility during training by weighting the corresponding losses. However, once the model is trained, the privacy-utility trade-off is frozen. In contrast, DISCO can allow dynamically varying privacy-utility at inference by tweaking the pruning ratio (R). However, this would also require the server's parameters (θ_2) to be trained with different R. This dynamic adjustment enables one to continuously control the privacy offered by deployed systems without having to interrupt or retrain the machine learning service from scratch.

3.5 Experiments

Datasets We conduct experiments with the following datasets:

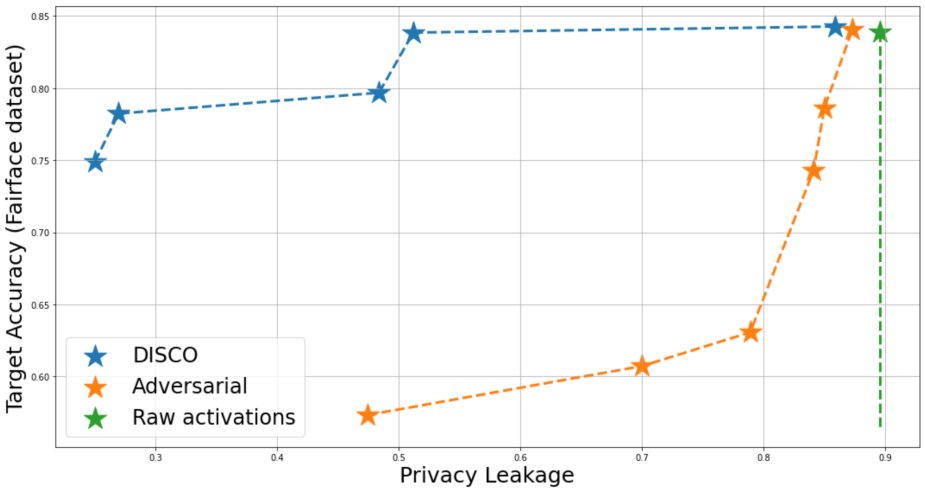

Figure 3-4: **Privacy-Utility Trade-off**: We vary the pruning ratio R for DISCO and λ trade-off parameter for ARL [104, 92]. Leakage is measured as SSIM score between inputs and reconstruction.

- Fairface [100] dataset consists of 108,501 images, with race, gender, and age groups. The dataset is designed with the emphasis of balanced race composition which we preserve in our experimental train and test sets. For our experiments, the task attribute is gender and the sensitive attribute is race.

- CelebA [111] consists of 202,599 celebrity face images across 10,177 identities, each with 40 attribute annotations. For our experiments, we define the task attribute as emotion and sensitive attribute as gender.

- CIFAR [98] consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. We manually label each of the 10 classes as living or non-living. For our experiments, the task attribute is the class label and sensitive attribute is living/non-living, as introduced in Roy and Boddeti [150].

Implementation Details Experiments are implemented using Pytorch and conducted using NVIDIA Tesla V100 GPUs. The backbone network is ResNet-18 [72] with the *client activations* obtained from the block-4, unless specified otherwise.

Evaluation Metrics We measure utility using top-1 accuracy on the task attribute. For attacks on sensitive inputs, we measure privacy using ℓ_1 loss, SSIM and

Method	Privacy (Fairface) ↓	Utility (Fairface) ↑	Privacy (CelebA) ↓	Utility (CelebA) ↑	Privacy (CIFAR10) ↓	Utility (CIFAR10) ↑
[133]	0.319	**0.824**	0.729	**0.916**	0.912	0.498
DISCO	**0.190**	0.815	**0.612**	0.910	**0.223**	0.9198
[150]	0.236	0.802	0.780	0.880	0.358	0.915
[92]	0.193	0.815	0.675	0.905	0.526	**0.924**

Table 3.1: **Comparison for sensitive attribute leakage:** We compare our approach on sensitive attribute leakage with the existing works. For the fairface dataset, sensitive attribute is race and task attribute is gender. In the CelebA dataset, sensitive attribute is gender and task attribute is smiling. The adversary accuracy is reported on the supervised reconstruction attack as described in 3.3.1. For all three methods, adversary accuracy is close to random chance, indicating that evaluation of privacy just by analyzing the adversary proxy during the training may give a false sense of privacy.

	SSIM ↓	PSNR ↓	ℓ_1 ↑	Utility ↑
Traditional [133]	0.88 ± 0.03	31.58 ± 2.44	108.82 ± 8.92	**97.35**
Adversarial [92]	0.68 ± 0.12	20.49 ± 5.94	123.33 ± 20.67	97.15
DISCO	**0.38±0.09**	**11.61±1.91**	**125.34±15.29**	95.66

Table 3.2: **Comparison for sensitive input leakage:** We compare our approach on sensitive input reconstruction task and compare with our baselines and the existing works.

PSNR [77] between reconstructed and input image and top-1 accuracy on the private attribute for attacks on sensitive attributes.

Baselines For attacks on sensitive attributes, we baseline with ARL based methods [92, 150, 104] which are state-of-the-art on attribute leakage. For attacks on sensitive inputs, we baseline with ARL methods [92, 104] relevant for sensitive inputs and two randomized variants of DISCO where we perform: i) random pruning ii) gaussian noise added in the feature space. Finally, for both sensitive inputs and attributes, we also compare with a traditional CNN model, denoted as*traditional*, with no activation privacy; this has been studied in Osia et al [133].

We present more reconstruction results for the qualitative comparison. Our results indicate that supervised decoder based attack model performs significantly better than likelihood maximization attack for *DISCO*; however, for all other techniques,

a likelihood maximization attack provides much better reconstruction quality. The figure can be found on the next page.

3.5.1 Hyper-parameters and Experimental Setup

All of the experimental setup is implemented in PyTorch and we will be releasing the codebase for all of different quantitative and qualitative experiments, with the random seeds used in all of the experiments.

Network architecture: We describe four distinct networks in the section 3, *client network*, *filter generating network*, *adversary network*, *task network*. We use ResNet-18 [72] as the base architecture for all of the four networks. For alignment of the architecture we experiment with the different blocks of the ResNet architecture and split the network such that output of the *client network* is fed to all three *filter generating network*, *adversary network*, and *task network*. The *filter generating network* has same number of neurons in the final fully connected layer as number of channels in the output produced by *client network*. The sigmoid temperature is 0.03 for the filter generating network. We adapt the ResNet backbone for *adversary network* when the protected attribute is sensitive input since it requires to build a generative model conditioned on *client activations*. We use a transpose convolution based architecture that upsamples the feature map to a higher dimensionality resulting in final image.

Pre-processing module described in the section 3.3.a is composed of a single convolution layer and a *spatial decoupler* that splits the feature-map into d^2 spatially disjoint partitions. For an image size of 112 and target d^2 to be 64, the resulting featuremap size is 14×14 that gets rescaled back to 112×112 using bilinear interpolation. We keep the value of the d^2 as 64 to make sure that the averaging in the channel space results in 64 distinct feature maps that can be fed into the remaining of the architecture, this allows compatibility of the *pre-processing module* with off the shelf architectures.

Optimizer: We use SGD optimizer with momentum [140] for all of the networks with a learning rate of 0.01

Figure 3-5: Qualitative comparison for different techniques using the supervised decoder attack. *DISCO (Off)* refers to DISCO with pre-processing module's toggle turned off. This technique results in a different yet realistic reconstruction for even *DISCO* compared to deep image prior results shown in the Figure 3-3 .

Sensitive Attribute	Method	Privacy (↓)	Utility (↑)
Mouth Open (S1)	[133]	0.814	0.893
	DISCO	**0.783**	**0.907**
Big Nose (S2)	[133]	0.616	**0.896**
	DISCO	**0.559**	0.893

Table 3.3: Privacy-utility trade-offs is influenced by correlation of task and sensitive attribute. The task attribute here is *Smiling* (yes/no). Both sensitive attributes are binary.

3.6 Discussion

In this section, we present the motivation and analyse the implication of various design choices for *DISCO*.

i) Privacy-Utility for Correlated Attributes While users idealize high privacy-utility guarantees, we posit that what level can be empirically realized is conditioned on the similarity of the task and sensitive attribute. To corroborate this position, we conduct leakage attacks using DISCO and *traditional* with the following attribute configuration: corroborate this with observations from the following experiments on the celebA dataset:

- **S1**: Sensitive Attribute is *Mouth Open* (yes/no) and the Task Attribute is *Smiling* (yes/no)
- **S2**: Sensitive Attribute is *Nose Size* and Task Attribute is *Smiling* (yes/no)

Results in Table 3.3 indicate DISCO achieves near-perfect privacy and high utility in S2, the privacy-utility worsens for S1 where the sensitive attribute (*mouth open*) is strongly correlated with task attribute (*smiling*) due to spatial overlap of the corresponding regions of interest.

ii) Comparing with Activation Noise for Privacy Adding noise to the output of a statistical query (*client activations* in this case) is a well known mechanism for privatizing sensitive data. These mechanisms are sometimes built under the framework of differential privacy [47] or its derivatives [126, 88]. While we do not compare or operate under a strict differentially private mechanism, we posit that preventing

sensitive input reconstruction requires a heavy amount of noise. To validate this, we design an experiment where we add Gaussian noise to the *client activations* and incrementally increase σ until the reconstruction is prevented. We also measure the difference in utility obtained by these noise based mechanisms. Compared to the learning based approaches like adversarial and DISCO, achieving privacy through random noise comes at a heavy cost of deteriorating utility to the extent that utility gets close to random chance with noise that is empirically capable of preventing reconstruction attack $\mu = -1, \sigma = 400$.

3.7 Conclusion

In this chapter, we show that sensitive information present in the latent representations of a deep learning model can be removed by selectively identifying a subset of features in the latent space and obfuscating them. Our analysis reveals that for some tasks, utility might be invariant to the obfuscation of sensitive features. This could yield an ideal utility-privacy trade-off by zeroing out a subset of activations from the hidden layer. To identify such a subset we use a learnable channel pruning and spatial pruning mechanism to remove sensitive features. Empirical results show the efficacy of our proposed method against existing state of the art works.

The proposed method DISCO, a dynamic scheme for obfuscation of sensitive channels to protect sensitive information can be used for enabling private collaborative inference. DISCO provides a steerable and transferable privacy-utility trade-off at inference without any retraining. We propose diverse attack schemes for sensitive inputs and attributes and achieve significant performance gain over existing methods on multiple datasets. To encourage rigorous exploration of attack schemes for private collaborative inference, we also release a benchmark dataset of 1 million sensitive representations.

3.8 Future work

There are multiple directions to improve DISCO and Private Collaborative Inference in general. First, the overhead to remove sensitive information on the edge device is high since a pruner model is used to remove that information. Second, a more rigorous theoretical analysis of optimal privacy-utility trade-off that factors in mutual information between the competing tasks would allow designing obfuscation mechanisms that get closer to the optimal trade-off. In this work, our focus was primarily on vision tasks but the idea of pruning sensitive information from representation space can be applied to other tasks and architectures also and we consider it as part of the future work. A formal guarantee on privacy and information leakage would be the next step before this application can be deployed to the real world.

Chapter 4

Private Data Release

4.1 Introduction

In the previous chapter we discussed private collaborative inference that aims to reduce sensitive information flow when using a trained model for prediction. One might ask, what about the privacy of users who contributed their data for training the model. Addressing the question of training data is the focus of this chapter. Over the last decade, research in deep neural networks has caused several sub-fields of computer vision, speech and natural language processing to leap forward for a variety of problems such as image classification, image captioning, face recognition, video understanding, automatic speech recognition, language modelling and many more. Even if unprecedented progress was made, the performance of deep learning is contingent upon the availability of large-scale datasets (e.g. image classification benchmark [39, 99], video classification benchmark [5, 90, 40], language corpus benchmark [30, 184, 171]). These large-scale datasets are often crowdsourced from individual users and often contain *sensitive* information in data that the user would like to keep private. This problem poses a fundamental dichotomy between the idea of centralizing datasets for machine learning and hiding sensitive information about the users present in the dataset. The sensitive information in the dataset may manifest as sensitive inputs or attributes, such as gender, race, age, etc for facial recognition systems. Unfortunately, the learned model is trained on crowd-sourced data that inherits unintended possibly sensitive

information. This, in turn, can expose serious privacy risks and can also be misused by the adversary to intercept sensitive information. In this context, great laws are being established to protect data security and privacy, such as General Data Protection Regulation (EU GDPR) and the California Consumer Privacy Act (CCPA). These laws hinder the aggregation of crowdsourced datasets. Therefore, there is a need for techniques with which one can obtain privacy-preserved data that cleverly anonymize sensitive attributes while preserving utility. Thus, how to obtain anonymized privacy-preserved data, allowing training of state-of-the-art models without any significant drop in utility, is the focus of this work.

At a broader scale, there are two categories of techniques that attempt to address this problem. The first category aims at using distributed machine learning instead of doing machine learning on centralized data. At this time, there are two prevalent techniques in distributed machine learning - 1) federated learning [121] and 2) split learning [70]. This category has multiple drawbacks, 1) It assumes individuals participating in crowdsourcing the dataset can perform machine learning at their end which mostly holds true for big institutions participating in distributed ML. 2) It is only applicable for a fixed target task, i.e., what model has to be learned has to be pre-specified before the training begins. However, if the dataset is centralized then a researcher could test multiple hypothesis.

The second category of solutions apply privacy-preserving transformation to the data samples before sharing it with an untrusted party. In the differential privacy literature this definition is commonly referred to as local differential privacy [143, 56]. This category suffers from a heavy privacy-utility trade-off. This trade-off gets exacerbated when the dataset is an unstructured modality like images, natural language and etc. Our proposed technique closely resembles this category while improving the privacy-utility trade-off by introducing the idea of replacing private attributes instead of removing them.

Conventionally, research in privacy-utility machine learning has primarily focused on protecting the privacy of the sensitive information contained in the data. In the context of preserving privacy, recent developments in model inversion attacks [43, 44,

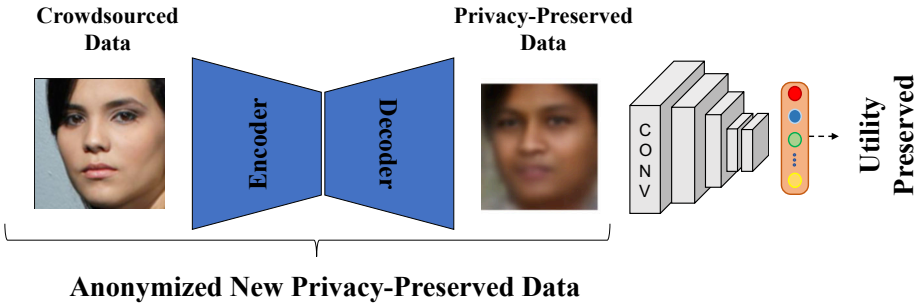

Figure 4-1: Our proposed approach applies a privacy-preserving encoder and decoder such that sensitive information from crowdsourced data can be replaced with a randomly sampled and synthetically generated attribute while preserving non-sensitive information.

117] have demonstrated that even with intermediate activations of a neural network, adversaries can reconstruct the raw input back or infer the sensitive private attributes such as race, age, gender etc. However, such methods predominantly apply transformation by dimensionality reduction followed by noise injection [133] to the features or exploiting local differential privacy [46, 54, 141] to obfuscate the data before sharing it with the service provider. Such an approach often leads to a significant reduction in utility. More recently, generative adversarial networks learning have been studied extensively to learn obfuscate [93, 104, 110, 130, 157] features from the raw image. Thus, this allows to decorrelate sensitive and non-sensitive data but is limited to setups when the primary utility task is known [157, 161], which limits its applicability to setups when the primary utility task is unknown [102].

In the following section, we introduce the idea of desensitization which is the key component of our proposed technique: **What is desensitization?** The process of removing sensitive information from a data sample by transforming it in such a way that the sensitive information is not recoverable from the transformed version. In this work we leverage the insight that data modality can be compressed into feature space, and this feature space, with appropriate training, can be decomposed into different semantic concepts. Given a sensitive attribute (like race, gender etc), only a subset of the semantic concepts would be causally responsible for the inference of that sensitive attribute. Therefore we perform training using variational autoencoder and other decorrelaton objectives such that these sensitive semantic concepts can be separated out

from the remaining semantic concepts. In order to desensitize the images, we sample the sensitive semantic concept from the distribution of sensitive semantic concepts such that the new sampled semantic concept is statistically independent from the original. This compressed representation where the sensitive information is replaced with a synthetic sensitive information is shared with the untrusted party.

Existing desensitizing techniques can be categorized under two broad categories, desensitization by information removal and desensitization by sampling from a data distribution. The first category of works aims to remove information by adversarial learning [104, 102, 178, 78], adding noise to the sample or representations [46, 28, 133]. However, utility-privacy trade-off is the main disadvantage of these set of techniques. The second category of approaches typically learns a generative model [85, 164, 27, 18] to synthesize new samples. There are two main disadvantages with this category of approaches: first, learning a data distribution from a dataset may or may not be tractable, and usually a model trained on synthetic datasets result in inferior performance compared to models trained on authentic data [144]. The second disadvantage emerges from the fact that the dataset obtained from synthetic samples might not be relevant for practical applications like facial recognition where the identity of individual needs to be preserved in order to obtain good prediction performance. Finally, the synthetically sampled dataset might allow for a distribution specific query but not a dataset query, i.e. questions specific to the dataset like how many faces are smiling in the given dataset may no longer be answered accurately. Our proposed approach, can be seen as the intersection of the two aforementioned categories: For every sample, we remove the sensitive information from the feature space and then replace it with a synthetically generated sensitive information such that *any* arbitrary downstream task except the one corresponding to the sensitive information is possible. We start by building upon one of the common assumptions in disentangled representation learning - every image can be decomposed into a disentangled set of features that carry different semantic information associated with the image. Once we have this set of features, we replace the ones correlated with sensitive user information and reconstruct the data back from the net set of features. This new dataset

still has sensitive information for representation learning, however, it is not tied to any particular individual and hence can be interpreted as an anonymization scheme. To the best of our knowledge, this is the first technique which can get rid of the privacy-utility frontier because of its unique approach.

4.2 Related Work

Privacy and anonymization has been a focus of many recent works in the machine learning community for different threat models and protection of different entities. These works can be broadly categorized into the following categories:

Private model release aims to learn a parametric model on private training data such that the existence of a particular data point can not be inferred with a reasonable high accuracy, also known as membership inference. Protection against membership inference directly comes as a consequence of the definition of differential privacy. A majority of the works [3, 134, 177, 122] in this direction give protection against the membership inference or model inversion attack. Our work focuses on the release of data itself rather than a learned model.

Synthetic data generation aims to learn a generative model for sampling data from a distribution that can be learned from a dataset [85, 164, 27]. Most of these works have only shown results for a tabular dataset with the exception of a few papers like Chang et al. [27], which provide results without any formal privacy guarantee. In comparison to this line of work, our goal is to apply a transformation over data points instead of sampling from the data distribution. These type of methods would not allow learning general data queries like how many people are smiling in the dataset. In addition, this would not enable building facial recognition software where the identity of an individual needs to be preserved in the new synthetic dataset. These are the two key differences of these methods in comparison to our work. Furthermore, in the method section we show that this category is a special case of the privatizing mechanism our work proposes.

Private Collaborative Inference is a body of work [157, 104, 151, 125] discussed

in the previous chapter where a privatized representation of data is shared in such a way that sensitive information is hidden while task-dependent information can be inferred about a sample. The key differentiating factor between private data release and private collaborative inference is the ability of private data release mechanisms to be domain agnostic.

Attribute privacy This category of work focuses on suppressing only the sensitive attributes. These methods [45, 91, 73, 108, 103] solely focus on privatizing with respect to a predefined set of sensitive attributes in the dataset while giving a differential privacy guarantee. However, they have been only studied for tabular datasets with the exception of GAP [79]. This line is closest to our proposed solution for data release as we also attempt to provide privacy for specific sensitive attributes.

4.3 Method

Consider a scenario where data holder A has a dataset $D = \{X, Y\}$ where X is a set of N images and $Y = \{Y_S, Y_{NS}\}$ where Y_S represents the set of labels carrying sensitive information and Y_{NS} represents the set of labels for non-sensitive information. A simple example could be where x, y_S, y_{NS} corresponds to the face, ethnicity, and expression of a person respectively. Let us say A wants to share D with analyst B without revealing sensitive information present in X that maps to Y_S. In brief, we propose a method where the user A maps its dataset D to a representation Z such that $Z = \{Z_S, Z_{NS}\}$ where Z_S,Z_{NS} carries information only relevant for Y_S,Y_{NS} respectively. We term this process of obtaining such a Z as *semantic separation*. After obtaining the Z, A applies a desensitizing transformation $f(Z_S) = Z'_S$ such that the transformed dataset $D' = \{X', \{Y'_S, Y_{NS}\}\}$ does not carry recoverable information about linkability between X' and Y_S. We term this process as *desensitization*. In order to perform *semantic separation* and *desensitization*, we require a disentangled representation and a latent space interpolation. We utilize a VAE [94] for learning network representations that can be used for the aforementioned requirements.

4.3.1 β-VAE

Building upon the definition of classic VAEs introduced in Kingma and Welling [94] where the data samples X are used to model the distribution of samples $x \in X$. This is accomplished by learning ϕ of approximate posterior function $q_\phi(z|x)$ and θ for the likelihood function $p_\theta(x|z)$. In this work, we use β-VAE [75] that regularizes the KL divergence between prior $p_\theta(z)$ and posterior $q_\phi(z|x)$. When the prior is chosen as isotropic gaussian, minimizing the KL divergence encourages disentanglement between the components z. The value of β gives a trade-off between reconstruction fidelity and disentanglement. The overall learning objective can be formulated as:

$$\max_{\theta,\phi}[\mathbb{E}_{x\sim D}[\mathbb{E}_{q_\phi(z|x)}[logp_\theta(x|z)]] - \beta D_{KL}(q_\phi(z|x)||p(z))] \tag{4.1}$$

Semantic separation is achieved by segmenting the components of z sampled from the approximate posterior in two mutually disjoint sets $\{z_S, z_{NS}\}$ where z_S carries, loosely speaking, information relevant to the sensitive task and z_{NS} carries the rest of the information about the image. The number of components k belonging to z_S out of the total m components of z is a hyperparameter that depends upon the cardinality of y_S, m, and the complexity of the sensitive task like identity, emotion prediction etc. The goal of *semantic separation* is to ensure two goals - i) z_{NS} should not carry any information about y_S and ii) z_S should not carry any information about Y_{NS}. The two goals are achieved by making use of three distinct mechanisms. First, we train the β-VAE as shown in 4.1 to encourage disentanglement in the latent space. In contrast to standard VAE [94], β-VAE [75] provides a controllable trade-off on disentanglement and reconstruction quality. Second, we train two predictor networks g (*sensitive predictor*) and h (*adversarial predictor*) parameterized by u and v such that u and ϕ are optimized to minimize the loss ℓ_1 with respect to y_S, while v and ϕ are optimized to respectively minimize and maximize the loss ℓ_2 with respect to y_S. The role of the sensitive predictor is to enforce certain components z_S of the vector z to carry sensitive information and the role of the adversarial predictor is to ensure that the sensitive information does not flow to z_{NS}. The loss functions ℓ could

be categorical cross entropy or ℓ_2 error depending upon the $\{y_S, y_{NS}\}$. The learning objective can be written down as:

$$\min_{\phi,u} \max_{v} [\ell_1(g_u(q_\phi(x)_i|_{i\leq k}), y_S) - \ell_2(h_v(q_\phi(x)_i|_{k<i\leq m}), y_S)] \tag{4.2}$$

Here k is the cardinality of the private feature z_S set which is decided beforehand and can be viewed as a hyperparameter for the main algorithm. Third, to ensure minimum information leakage about z_S in z_{NS}, we use distance correlation for minimizing the mutual information between the two vectors. Usage of distance correlation for reducing the information leakage was first introduced by Vepakomma et al. [169]. While they minimize distance correlation between activations and inputs, we minimize it between the two disjoint subsets of activations. Mutual information estimation for high dimensional variables is inefficient and hence different measures are used for estimating the information like HSIC, MMD and etc. Distance correlation captures both linear and non-linear relationships and allows efficient computation for backpropogation. The learning objective can be written down as:

$$\min_{\phi} dcorr(q_\phi(x)_i|_{i\leq k}, q_\phi(x)_i|_{k<i\leq m}) \tag{4.3}$$

The parameters ϕ, θ, u, v are trained jointly as specified in the equations 4.1, 4.2, and 4.3 under a simultaneous optimization [124]. In order to balance the relative importance of different modules, we use four hyper-parameters - $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ controlling the relative importance of β-VAE, *sensitive predictor*, *adversarial predictor*, and distance correlation.

Desensitization for a sample x is achieved by obtaining the range of values for each component in Z_S and randomly sampling z'_S and replacing it with z_S such that $z = \{z'_S, z_{NS}\}$. More formally, by the definition of VAEs we know that each latent variable is standard normal; thus, random sampling the sensitive attributes with $q_\phi(X)_i|_{i\leq k}$ would ensure that Z'_S is statistically independent from Z_S. With the newly obtained Z'_S we compute $Y'_S = g(Z'_S)$ and $X'_S = p_\theta(Z'_S))$. Finally the desensitized dataset $D' = \{X', \{Y'_S, Y_{NS}\}\}$ is shared with B. In comparison to the

existing methods that attempt to remove sensitive information through adversarial learning or adding noise [79, 103, 108], the **advantage** of this technique comes from sharing Z'_S and Y'_S that allows B to build predictors even on Y'_S. In addition, B can also get access to q_{ϕ^*}, and does not need to train a new CNN model, but just a single layer fully connected network for the prediction of any subset of Y'.

Threat Model: Under our threat model the untrusted receiver B can act as an adversary with access to a set of pairs of leaked dataset $\tilde{D} = \{X', \{Y_S, Y_{NS}\}\}$. Note that the leaked dataset carries the mapping from desensitized input samples X' to their sensitive labels Y_S and not Y'_S. Therefore, the attacker can learn a f_{adv} which attempts to leak actual sensitive labels of a query x'. Therefore, the goal of our desensitization scheme is to maximize the error on the sensitive label prediction task during the test phase. For the evaluation purpose in this paper, we simulate the adversary's f_{adv} by training a CNN based architecture on X', Y_S. Unlike threat models considered in differential privacy like central or local DP which offer a post-processing invariance property, here the goal is to prevent sensitive information leakage and hence under this threat model, we consider sensitive information leakage as the target threat with access to only $\tilde{D}$.

4.4 Experiments and Results

We conduct quantitative and qualitative experiments for evaluating the efficacy of our proposed method. For quantitative evaluation, we measure privacy-utility trade-off where privacy refers to the adversary's capability to infer sensitive performance. The goal here is to examine whether Z_{NS} carries sensitive information about Y_S. Our adversary is trained on a dataset obtained from desensitizing (Z_{NS}, Y_S), we do not include Z_{NS} since it gets randomly sampled and hence would only make an uninformed adversary mine spurious correlation. This would indicate lack of perfect disentanglement between the sensitive and non-sensitive attributes. The work which introduced β-VAE [75] observed decrease in the sample quality as the disentanglement is increased. This observation indicates that there will be some privacy-utility trade-

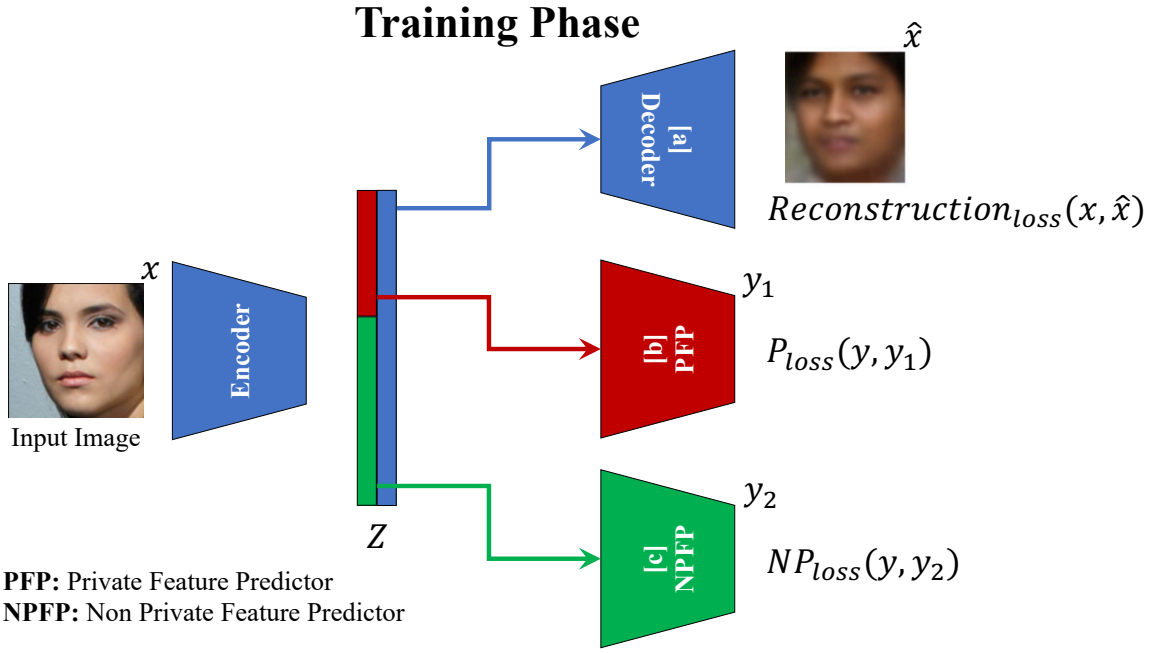

Figure 4-2: Main architecture of the proposed technique is based on VAE (blue colored encoder and decoder). We partition the latent space of the VAE Z into two disjoint sets (denoted by red and green color), the red set is trained to carry sensitive information while the green set is trained to carry the non-sensitive information. This constraint is enforced by training red and green classifiers. Post-training, we randomly sample the sensitive set and generate a non-private image x'.

off as we strive towards high disentanglement (for privacy) while high sample quality (for utility).

Dataset: We evaluate our technique and baselines on the FairFace [100] dataset, which is a face image dataset of 108,501 face images with three categorical labels - ethnicity, gender, and age. The dataset has been developed in a way such that each demographic is represented in the dataset uniformly. In our experiments, we use race as a sensitive attribute and measure utility by training a classifier to predict gender on the dataset that has sensitive information about ethnicity replaced.

Baseline: We choose adversarial training as our baseline. The idea behind adversarial training is to train two networks, one to minimize the sensitive information leakage and the other to maximize the reconstruction quality. In this way, the network should be able to learn representations that hide the sensitive information while

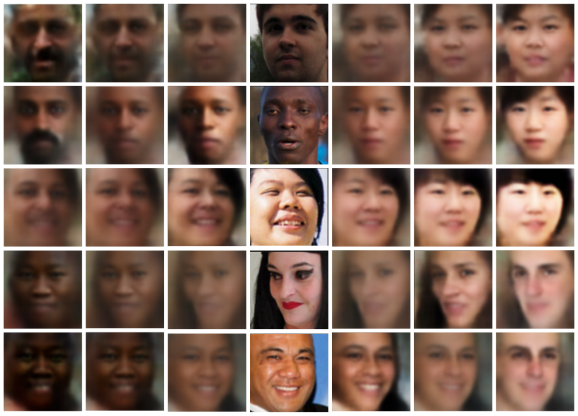

Figure 4-3: **Latent Space Interpolation in private dimension.** The image in the middle column is the original image taken from the Fairface dataset [100]. The sensitive attribute corresponding to the private dimension is ethnicity.

preserving semantic information in the generated image so as to be useful for downstream tasks encountered after the data release. Adversarial training is the main foundational block for TIPRDC [103] and GAP [79]. This ideas has been widely used in the private collaborative inference literature [151, 16, 104].

First, we measure the sensitive information leakage in the absence of any privatizing mechanism. In our experiment setup we treat ethnicity of a person as sensitive information. Without any privatizing mechanism, the adversary obtains the accuracy of **57.39%**. Using our approach we find that the adversary is able to recover the sensitive information with an accuracy of **22.1%**, While using the adversarial training approach the adversary's performance is **33.3%**. Note that the lowest possible performance on this six-class classification problem is **19.0%** due to a small class imbalance. Therefore our approach obtains a better leakage reduction in comparison to the adversarial training approach.

Second we measure the utility by measuring the performance on a binary gender classification on this privatized dataset. On the non-private version, we obtain an accuracy of **86.51%**. Our approach results in an accuracy of **69.65%** while the adversarial training baseline obtains the accuracy of **71.84%**. Here adversarial training obtains a higher performance, however, the relative difference between our technique and adversarial is nominal.

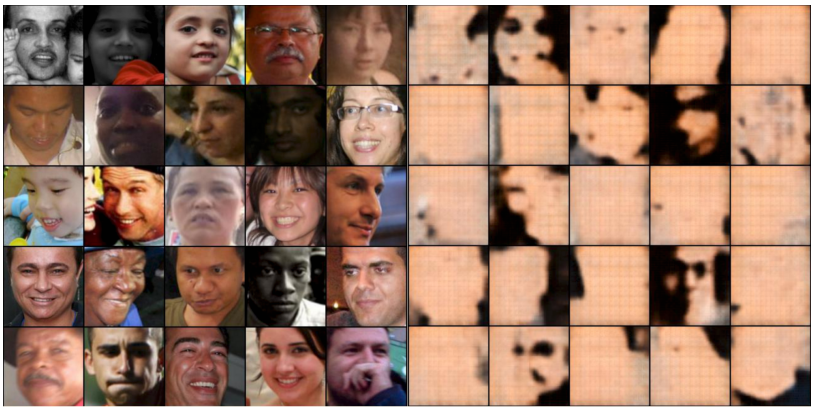

Figure 4-4: **Visualization of images produced by adversarial training.** We visualize the images generated by the decoder used in the adversarial training by plotting it on the grid on the right. The trained model and original image grid on the left is taken from Fairface dataset [100]. The sensitive attribute here is ethnicity.

The overall performance difference in the sensitive information leakage as well as utility evaluation indicates that our proposed technique can result in better privacy protection while maintaining similar levels of utility. We also visualize the images generated by our method and adversarial training. To further analyze the quality of the released samples, we visualize the multiple instances of different individuals and perform latent space interpolation. The goal of latent space interpolation is to show different random samples possible for a given face image. As seen in fig. 4-3, different samples indeed produce different instances of sensitive information (chosen as race for the experiments) while reasonably preserving other aspects of face like azimuth, emotion, expression and etc. However, not all features get disentangled and hence we see some undesirable associations as seen in fig. 4-3 where gender also appears to be affected when adjusting ethnicity. In contrast, adversarial training based representations result in the images shown in figure 4-4, where the network appears to be obfuscating majority of the information from the image in order to remove the sensitive information. The main difference between our technique and adversarial training is that instead of removing sensitive information we replace it with synthetically sampled sensitive information.

4.4.1 Ablation Study

We evaluate the performance contribution of each component described in the architecture in Fig. 4-2. We measure the change in sensitive information leakage by comparing performance with and without each component in the loss function. This can be interpreted as setting up $\alpha_i = 0$ for the i'th component during the training phase.

Distance correlation: Removing distance correlation results in the sensitive information leakage as 29.7% depicting that removing distance correlation increases the information leakage of the sensitive attribute (y_S) in z_{NS}.

Adversarial predictor (h): Similar to ablating the distance correlation loss function, we see increase in the information leakage of sensitive attribute in the z_{NS} as the accuracy of sensitive attribute prediction goes to 28.04%.

Sensitive predictor (g): The sensitive predictor is tasked to make sure information relevant to y_S is present.

The sensitive attribute prediction accuracy is 22.16%, indicating that all three components help in preventing leakage together.

4.5 Conclusion

In this chapter, we discussed the requirement for privacy-preserving data aggregation mechanisms and the existing works in this direction. Then we presented a method for desensitizing images by learning a generative model. We showed that every image can be segmented into sensitive and non-sensitive sets of variables in the latent space and that replacing the sensitive set of features with a randomly sampled sensitive set of features allows us to generate a new image which does not carry sensitive information linked to any individual. It is important to highlight the notion of *replace* which our proposed technique uses instead of *remove* which a majority of the existing techniques use. The notion of *remove* suffers from relatively high privacy-utility trade-off especially when the private attribute and sensitive attribute are correlated.

4.6 Future Work

This work presented a technique for desensitizing data that removes sensitive information from dataset. While the technique is data agnostic, the evaluation was only performed for image datasets. Therefore evaluation of this technique on other modalities like natural language would be important to assess its efficacy. While the focus of this work is on the privacy mechanism, improving the sample quality using hierarchical models would allow us to truly cross the barrier of privacy-utility trade-off as discussed in the section 4.3. Other future directions include formalizing the privacy definition for the proposed class of technique and giving bounds on the information leakage. It appears that due to the approximate posterior estimation, the objective of VAEs dovetails nicely with the objective perturbation methods that were introduced in differential privacy [29] but could not gain momentum due to inferior privacy-utility trade-off. Therefore more investigation of VAEs and their connection with the objective perturbation might result in differential privacy mechanisms for empirical risk minimization framework.

Chapter 5

Private Set Intersection

5.1 Introduction

In the last two chapters we focused on designing a privacy-preserving mechanism for machine learning and general statistics. In this chapter we shift the discussion towards a more fundamental problem of computing multi-party set intersection in a secure manner. While the highest level of privacy can be achieved by performing all of the computation on the edge device, multi-party set intersection is the special case where it is not possible for any single party to do on-device computation. Hence, this problem fundamentally requires both privacy and distributed computing as the computation needs to be done jointly among multiple parties. The Private Set Intersection (PSI) solution allows two or more entities to know if their respective sets have common elements. PSI comes in different flavors like giving a binary output about whether there is an intersection or not, providing the cardinality of the intersected set, or providing the intersected set itself as the output. Most of the existing work in PSI has assumed that both parties hosting their own private sets work interactively. However, we present a compelling use case where this can not be realized practically and hence, existing algorithms and protocols may not help because the private set intersection task is delegated to a server. In the scenario where the party which goes offline does not trust any other entity, we need to distribute the computation and the encrypted data hosting service across different nodes. To motivate the underlying

use-case, we present the following COVID-19 based digital contact tracing problem statement -

- Every App user records their GPS data every five minutes and stores it in the following format (latitude, longitude, ts) where latitude and longitude correspond to the geo-coordinates at timestamp ts. In this format, every day there are 288 points stored in the local database.
- A set of users Q among all the users get tested positive and upload their dataset of location trails.
- The remaining users P want to compute the intersection to check whether they have been in the same location as an infected user.
- However, Q and P do not want to leak their privacy by submitting the data in raw format; hence, they cryptographically share this data and perform some additional computations in order to compute the intersection of their trajectories in a secure manner.
- A match is found between User i and User j if -

  $dist((lat_i, long_i), (lat_j, long_j)) < dist\_threshold$ and $(time_i - time_j) < time\_threshold$

  To avoid the exact computation, we can discretize the latitude and longitude values into bins and the extent of discretization can be chosen based on the $dist\_threshold$.

Desirable properties:

- Q can go offline after securely uploading their data.
- Server should not know the outcome of the results.

The problem can be cast as a set intersection protocol [80, 63, 58] in which a certain number of the parties go offline once they upload their encrypted sets. In our solutions we require 2 non-colluding servers where the second server allows us to achieve

a solution in which the infected people can go offline. There are further attacks which do target the privacy of users but instead maliciously inject wrong data. For example - An attacker who acts as an infected person and spoofs their GPS and sends coordinates of a different place to create panic. Our proposed method can be extended to circumvent some of these malicious adversaries and should be considered as part of future work.

5.2 Related Work

Digital Contact tracing has emerged to be one of the promising solutions for curbing the COVID-19 disease spread [113, 145, 10, 142, 9, 36]. Most papers that have proposed the usage of digital contact tracing have based it on the peer to peer protocols like Bluetooth [69, 165, 160, 13], Ultrasound [112], and other similar broadcast based sensors [123]. For a more comprehensive survey of various existing protocols, we refer the reader to Martin et al. [119]. One of the most notable protocols deployed widely is the Google/Apple protocol [69] due to its hardware level support for enabling the smooth functioning of the protocol. While the bluetooth-based approaches have been widely studied, there are several limitations associated with such approaches [101, 166] that make a strong case of substituting or augmenting these peer-to-peer approaches with the location data [138, 32, 159, 89].

Unlike all of the existing work, our work presents the Private Set Intersection problem where the entropy in the data point is low to prevent a brute force attack. Kato et al. [89] propose a relatively similar setup; however, they use trusted hardware to achieve privacy goals. The assumption of an available trusted hardware might not be applicable always and also brings scaling issues since the user has to rely on a dedicated piece of hardware. Another practical constraint presented in this work is that the client can go offline after uploading their data and an untrusted server holds this data in encrypted format for the curious clients who want to perform the set intersection to check their health status.

Private Set Intersection(PSI) is one of the key building blocks in design of all existing private digital contact tracing systems. PSI protocols allow interaction between two clients, each with their own private set of data, such that by the end of the protocol they can find out *some particular* information about the intersected set. PSI can come in different flavors based on what information needs to be disclosed. In some cases, it is the cardinality of the intersected set or the elements of the intersected set itself. The focus of this work is to learn a single bit indicating whether the cadinality of the intersected set is greater than 0 or not. There is a significant amount of work done in this domain [136, 31, 137, 146, 51, 132, 135, 147, 95]. In comparison to these existing works, our focus is on the design of a PSI protocol that can allow one party to go offline after uploading their set to servers such that the server can not learn information about these sets while allowing individual clients to perform PSI. This design constraint originates from the practical requirements associated with digital contact tracing systems.

5.3 Preliminaries

5.3.1 Naive hashing based protocol

In simplistic settings, the user Q can hash its data and upload the hashes. The user P downloads this hash from the central server and compares with its own hash. This scheme is very simple and computationally efficient. However, in our proposed use case as well as in general this scheme is insecure due to the P's capability to enumerate over all the possible data points and hence find Q's actual trajectory.

5.3.2 RSA encryption

RSA [148] is one of the widely known public-key crypto system. The security guarantees of the RSA system are based on the hardness of the factorization problem. The protocol operates under the following definition

Definition 5.3.1. *The RSA encryption scheme*

$$\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Eval}, \mathsf{Decrypt})$$

is an asymmetric encryption algorithm defined as follows:

- $\mathsf{KeyGen}(n, e, d)$

  Compute $n = pq$ where p and q are prime numbers. Now e is chosen in a way that $1 < e < \lambda(n)$, and e and $\lambda(n)$ are co-prime. Here $\lambda(n) = lcm(p-1, q-1)$. The parameter e acts as the public key and $d \equiv e^{-1} mod(\lambda(n))$ is the private key.

- $\mathsf{Encrypt}(e, x)$

  The encryption can be performed on a message x by computing $c = x^e$.

- $\mathsf{Eval}(c_1, c_2)$

  Two ciphertexts c_1 and c_2 encrypted with the same public key e can be compared against each other to perform equality test in the encrypted space.

- $\mathsf{Decrypt}(d, x)$

  The decryption can only be performed by the owner of the private key by computing $(x^e)^d \equiv x(mod(n))$.

5.3.3 ElGamal encryption

The el-gamal cryptosystem [53] is also a well known public key cryptosystem which bases its security guarantees on the hardness of the discrete logarithm problem.

Definition 5.3.2 (ElGamal Encryption)**.** *The ElGamal encryption scheme*

$$\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Eval}, \mathsf{Decrypt})$$

is a probabilistic encryption algorithm defined as follows:

- $\mathsf{KeyGen}(G, q, g, e) \rightarrow (\mathsf{sk}, \mathsf{pk})$

  Construct group G of order q with generator g. The private key is an integer e randomly sampled from the group G. The public key is g^e and the group G.

- $\mathsf{Encrypt}(x, \mathsf{sk}) \rightarrow (c_1, c_2)$

  a third party can encrypt a message x by first sampling random d from the group G and computing $c_1 = g^d$ and $c_2 = g^x.(g^e)^d$.

- $\mathsf{Eval}(c_1, c_2, \mathsf{pk}) \rightarrow \mathsf{ct}$

  Given two ciphertext, $c_1 \times c_2$ results in $g^{e.d(x_1+x_2)}$

- $\mathsf{Decrypt}(c_1, c_2, \mathsf{pk}) \rightarrow m$

  is performed by the receiving party only if they know the private key e as $x \equiv (c_2).((c_1)^e)^{-1}(mod(n))$. The inverse can be efficiently computed by using the value of q.

5.3.4 Leveled fully homomorphic encryption

In this section, we give high level definitions for the algorithms comprising a leveled homomorphic encryption scheme as well as necessary security definitions.

Definition 5.3.3 (Leveled Homomorphic Encryption)**.** *A leveled homomorphic encryption scheme*

$$\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Eval}, \mathsf{Decrypt})$$

is a set of PPT algorithms defined as follows:

- $\mathsf{KeyGen}(1^\lambda, 1^L) \rightarrow (\mathsf{sk}, \mathsf{pk}, \mathsf{evk})$

  Given the security parameter λ and a maximum circuit depth L, outputs a key pair consisting of a public encryption key pk*, a secret decryption key* sk*, and an evaluation key* evk*.*

- $\mathsf{Encrypt}(\mathsf{pk}, m) \rightarrow \mathsf{ct}$

  Given a message $m \in \mathcal{M}$ and an encryption key pk*, outputs a ciphertext* ct*.*

- $\mathsf{Eval}(\mathsf{evk}, f, \mathsf{ct}_1, \mathsf{ct}_2, \ldots, \mathsf{ct}_n) \to \mathsf{ct}'$

  Given the evaluation key, a description of a function $f\colon \mathcal{M}^n \to \mathcal{M}$ *with multiplicative depth at most* L*, and* n *ciphertexts encrypting messages* $m_1, \ldots, m_n$*, outputs the result ciphertext* ct' *encrypting* $m' = f(m_1, \ldots, m_n)$*.*

- $\mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct}) = m$ *Given the secret decryption key and a ciphertext* ct *encrypting* m*, outputs* m*.*

Optionally the scheme $\mathcal{E}$ may be extended with a PPT algorithm $\mathsf{EncryptSK}(\mathsf{sk}, m) \to \mathsf{ct}$ which uses the secret key sk rather than the public key pk, to compute the ciphertext ct from the message m.

When returning a ciphertext output by the Eval function, it is often desirable for this ciphertext to hide the function f that was used to produce it. This property of the scheme is called circuit privacy, which we formally define below.

Definition 5.3.4. *Circuit Privacy ([81] definition 7, [22] definition 5.1)*

Let $\mathcal{E}$ *be a leveled homomorphic encryption scheme as defined in definition 5.3.3. Define the following:*

$$
\begin{aligned}
(\mathsf{sk}, \mathsf{pk}, \mathsf{evk}) &\xleftarrow{\$} \mathcal{E}.\mathsf{KeyGen}(1^\lambda, 1^L) \\
\mathsf{ct}_i &\xleftarrow{\$} \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}, m_i) \quad i \in [1 \ldots k] \\
\mathsf{ct}_i &\xleftarrow{\$} \mathcal{E}.\mathsf{EncryptSK}(\mathsf{sk}, m_i) \quad i \in [k+1 \ldots n] \\
\langle \mathsf{ct}_i \rangle &:= \{\mathsf{ct}_i\}_{i=1}^n \\
m_{out} &= f(m_1, \ldots, m_n)
\end{aligned}
$$

We say that $\mathcal{E}$ *is* ϵ-circuit private *for functions* f *of depth* $\ell \leq L$ *if there exists a PPT simulator algorithm* Sim *such that for all PPT distinguishing algorithms* $\mathcal{D}$ *the*

following holds:

$$\Bigg| \Pr\Big[\mathcal{D}\Big(\mathcal{E}.\mathsf{Eval}\big(\mathsf{pk}, \mathsf{evk}, f, \langle \mathsf{ct}_i \rangle\big), \langle \mathsf{ct}_i \rangle, \mathsf{sk}, \mathsf{pk}, \mathsf{evk}\Big) = 1\Big] - \Pr\Big[\mathcal{D}\Big(\mathsf{Sim}\big(1^\ell, \mathsf{pk}, \mathsf{sk}, \mathsf{evk}, m_{out}\big), \langle \mathsf{ct}_i \rangle, \mathsf{sk}, \mathsf{pk}, \mathsf{evk}\Big) = 1\Big] \Bigg| \leq \epsilon$$

In other words, definition 5.3.4 says that the output of the evaluation algorithm of a circuit private leveled homomorphic encryption scheme should be indistinguishable from a simulated output, where the simulator is given no information about the function f used to compute the result ciphertext other than the function output. We can view the simulator in definition 5.3.4 as an alternate encryption procedure that produces a fresh ciphertext that is indistinguishable from the output of the real $\mathcal{E}.\mathsf{Eval}$ algorithm. We only consider functions f that will result in $\mathcal{E}.\mathsf{Eval}$ outputting a correct ciphertext. This can be implemented by having both $\mathcal{E}.\mathsf{Eval}$ and Sim output $\perp$ for functions that are exceed the number of levels supported by $\mathcal{E}$.

We will use the homomorphic encryption scheme of Brakerski, Fan, and Vercuteran [23, 59], denoted the BFV scheme.

In order to effectively choose parameters for our leveled homomorphic encryption scheme, we must accurately upper bound the noise growth due to homomorphic operations. When multiplying two elements of $\mathcal{R}_q$, we need an upper bound on the norm of the product by a function of the norms of the operands as well as properties of $\mathcal{R}_q$ itself.

Definition 5.3.5 (Ring Expansion Factor [114, 66])**.** *The expansion factor $\delta_\mathcal{R}$ for a ring $\mathcal{R}$ is defined as follows:*

$$\delta_\mathcal{R} = \max_{a,b \in \mathcal{R}} \frac{||a \cdot b||}{||a|| \cdot ||b||}$$

Lemma 5.3.1 (Ring Expansion Upper Bound)**.** *For a ring $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$, we*

can upper bound the ring expansion factor $\delta_{\mathcal{R}}$ for the norm $||\cdot||$ by

$$\delta_{\mathcal{R}} \leq n$$

Proof. In the worst case for the expansion of the norm $||\cdot||$ is when both polynomials $a, b \in \mathcal{R}$ have coefficients of all the same magnitude. In this case, the maximum coefficient of the product will be n times the product of the maximum coefficient of the operands. Therefore, we have

$$||a|| \cdot ||b|| = n \cdot ||a \cdot b|| = \delta_{\mathcal{R}} \cdot ||a \cdot b||$$

□

Remark 5.3.1. *The upper bound in lemma 5.3.1 extends to $\mathcal{R}_q$ for any modulus q.*

5.4 Method

Apart from the cryptographic method based security, we add another layer of defense by adding rate limiting for all communication between server to client and server to server. This is to make sure that brute force attacks are infeasible. While the network communication latency already makes some of the brute force attacks infeasible, we still keep the rate limiting as a measure to strictly enforce the necessary query limit. This system level solution to improve the security of the system is discussed in more detail in the section 5.5.1.

Protocol Description Every protocol defined in the following sections can be segmented into four phases as follows:
Registration Phase - q goes offline after communicating and registering its data to the servers and doesn't interact at all with any server or client during the online phase.
Request Phase - p requests for data/computation to be performed in order to obtain the result and get exposure notification.
Computation Phase involves distributed computation between p, S_1, and S_2 where

P	Set of healthy person
Q	Set of infected person
S_1	Server 1
S_2	Server 2
$\|P\|$	number of healthy person
$\|Q\|$	number of infected person
X	location data of healthy person, can be thought of as a set
x	an arbitrary element in the set X
Y	location data of infected person, can be thought of as a set
y	an arbitrary element in the set Y
$\|Y\|$	total number of entries in Q's dataset
$\|X\|$	total number of entries in P's dataset
n	total number of locations

Table 5.1: Table for notation followed in this chapter.

INPUTS: Set X from party $p \in P$ and Y from party $q \in Q$
FUNCTIONALITY:

1. q provides input Y. q then immediately goes offline.
2. p inputs X
3. p receives a single bit indicating if $|X \cap Y| > 0$.

Figure 5-1: Protocol definition for Private Set Intersection with offline parties. PSI-CA is PSI-cardinality. PSI-ICA short for PSI-indicator cardinality.

the set intersection is performed.

Results Phase - p obtains the final output of the protocol in this phase and interprets the result. This phase may or may not include the communication with one or more servers based on the protocol.

Attacker taxonomy: In the following section we describe different attackers and all the ways in which they can attempt to obtain secret information about either an infected user q or a healthy user p. Note that in the current protocol, we only focus on the semi-honest adversaries as the attackers.

Semihonest User - An attacker could be posing as a healthy person with the intent to leak an infected person's trail using GPS. There are potentially different

attack targets for this user. For example - they might be targeting a single person to know whether they are infected or not. They might try to decrypt the location information of all individuals they receive.

Semihonest Server - One of the servers S_1 or S_2 could also act as an attacker and try to take sensitive information out of the encrypted data or try to interpret encrypted results. The goal of the rogue server could be to decrypt location information or obtain the results of the intersection.

Server collusion - Both Server 1 and Server 2 could try to collude in order to leak information about either a healthy or infected person.

Client-Server collusion - The Malicious App described in i) could team up with a malicious server in ii) to target user's privacy. Another way of manifestation of this attacker is that the server also acts as an App provider. Another potential attack is where the server can deliberately inject a few points as an infected person to obtain the Healthy person's information.

Malicious Query Attack - In this attack scheme, the attacker sends a maliciously crafted query that would leak more information about the infected individuals that would not be revealed otherwise. One such example is the *timestamp attack* in which the attacker has the knowledge about potential visiting points of an individual and they query whether this user is present in the set of infected individuals or not. This is quite practical in the instances where the attacker knows the set of points in one which the target user will be present at any given point of time (ex. - if the attacker knows target user's workplace and house location, they can keep all timestamps enumerated for these two given locations). Since we have imposed a limit on the number of queries, the attacker uses the same timestamp for all the location instances they want to query.

5.4.1 Hashing scheme

Method In the hashing scheme described in the Figure 5-2, the infected client q first gets their data encrypted by a key signing server S_1 by sending the blinded version of its data to the server and then uploads this encrypted data to the query server S_2 that hosts this encrypted data. By encryption, here we simply refer to raising a given number x by key k. The healthy/susceptible individual p then first gets their data encrypted with the key known to the server S_1 and then obtains the data held by the server S_2 and computes set intersection. In this protocol, the underlying security comes from the fact that the data is encrypted by S_1 while the set intersection is performed by p who does not know the encrypting key and hence can not perform the brute force attack.

INPUTS: Two sets X and Y.
PARAMETERS: Cyclic group of order q, hashing function h.
REGISTRATION PHASE:

1. q sends $h(y)^r$ to S_1 for a random r where $h()$ is a collision resistance hash function. ($\forall y \in Y$ and similarly $\forall x \in X$, in the following steps as well)
2. S_1 returns $(h(y)^r)^k$ for a random k
3. q computes $((h(y)^r)^k)^s$, such that $s.r = 1$ and sends the result to q $(h(y)^k)$ to S_2
4. q goes offline

REQUEST PHASE:

1. p sends $h(x)^r$ for another random r to S_1.
2. S_1 returns $(h(x)^r)^k$
3. p computes $((h(x)^r)^k)^s$, where $s.r = 1$
4. p downloads all $h(y)^k$ from S_2 and compares it with its own $h(x)^k$

Figure 5-2: Description of the hashing scheme based PSI. Note that computing inverse in the exponent would require euler totient functions, hence, we use RSA protocol as the building block for this scheme.

Security The security of this protocol depends upon the discrete logarithm problem. Given the user p and q's location information, the malicious server S_1 has to know the discrete log of $h(x)^k$. The hash function would not improve the security here from a malicious server's point of view because the entropy in the location information x is very low and hence it is prone to dictionary attack by itself. However, the hash function serves improves the security against malicious client by ensuring that a client who computes $h(x)^k$, should not learn anything additional about k. For instance, if we do not use h, then the client can learn $x^{m.k}$, for any integer m, from x^k by raising the power. The protocol does not protect against the *client-server* collusion and *server* collusion.

While the definition in the Figure 5-1 requires p to learn only a single bit of the information about the intersection of the two sets, this protocol allows a healthy client p to learn the cardinality of the intersected set, hence leaking extra information. This extra leakage may or may not be acceptable dependent on the practical application. For instance, a majority of the digital contact tracing systems require p to learn a risk score which is a function of the cardinality of the intersected set.

Computation Efficiency The computational power of the system proposed here does not require any real-time communication given the nature of the digital exposure notification technology and hence can be spread across time with a significant delay (e.g. 6-12 hours). However there could be other constraints like battery usage in the phone that should be taken care of. The scheme offers a high computational efficiency for the infected client q since it performs two exponentiations for every element in the set. The curious client p has to perform two exponentiations operation for every element. Additionally, they also need to perform an equality check with each element in the infected set. This can be optimized to a certain degree by q sorting the two set which would give the computational complexity of $\mathcal{O}(|Q| \cdot |Y| \cdot \log(|Q| \cdot |Y|))$ assuming the exponent operation to be a constant time operation for a fixed number of bits of the elements of a set. For the server S_1, the protocol requires it to take each element x and raise it to the power k and hence requires a single round of

exponentiation of each element over the sets to be intersected. Therefore, it has the computational complexity of $\mathcal{O}(|Q| \cdot |Y| + |P| \cdot |X|)$ and for the server S_2, the computational complexity would be $\mathcal{O}(1)$ because the server only acts as a file hosting server and does not perform any computation.

Communication Efficiency The communication efficiency of the client p would be sub-optimal since it has to download all of the infected person's dataset. More precisely it would be $\mathcal{O}(|Q| \cdot |Y|)$. The communication efficiency of the server S_2 also would be sub-optimal on the similar lines with upper bound as $\mathcal{O}(|P| \cdot |Q| \cdot |Y|)$.

5.4.2 ElGamal Encryption Scheme

Method: The el-gamal based encryption scheme is similar in design to the previous hashing scheme; however, the key difference lies in the computation performed by the server S_2. Here the server S_2 is able to perform the computation on encrypted data by performing the re-keying operation and utilizing the additive homomorphic nature of el-gamal scheme. Under this protocol, the result of the set intersection is sent to p unlike the full encrypted set to the client. This transfer of computation on the server provides higher computational efficiency on the health person side p and also improves the communication efficiency since only the result of set intersection is sent to the client.

Security The security of this protocol depends upon the security of the underlying building block of the system which is an el-gamal cryptosystem [53]. For the el-gamal system the security relies on the decisional diffie-helman problem [20]. Furthermore, the server S_2 also gets to see x^t for an arbitrary t where the security relies on the discrete log problem. Like the previously described method, this scheme is also prone to the *client-server collusion* with the minor difference being that only server S_2 can perform this collusion attack and not server S_1.

INPUTS: Two sets X and Y
PARAMETERS: Cyclic group G of order q.
REGISTRATION PHASE:

1. S_2 chooses a key pair for ElGamal encryption (pk_q, sk_q) and sends pk_q to q
2. q encrypts each location in Y using ElGamal Encryption $C_q = (c_1, c_2) = \mathsf{Encrypt}(pk_q, y)^t$ for a random mask t (use different t if we don't want to leak the ratio of locations) and sends t to S_1 and C_q to S_2. $[c_1 = (g^y)h^r, c_2 = g^r]$ q goes offline

REQUEST PHASE

1. p receives t from S_1
2. p chooses (pk_p, sk_p) and encrypts his location$C_p = \mathsf{Encrypt}(pk_p, -x)^t$ and sends C_p and pk_p to Server 2.
3. Server 2 decrypts C_q using sk_q, to get g^{y*t} and encrypts it again under the key pk_p such that
$$C'_q = \mathsf{Encrypt}(pk_p, Dec(sk_q, C_q))$$
4. $ct_i = (C_p * C'_q)^s$ for a random s and sends it to p
5. p decrypts ct_i with sk_p to get $g^{s*t(y-x)}$ i.e. $\mathsf{Decrypt}(sk_p, ct_i) = \mathsf{Decrypt}(sk_p, (C_p + C'_q)^s)$
$= \mathsf{Decrypt}(sk_p, (d * \mathsf{Encrypt}(pk_p, \mathsf{Decrypt}(sk_q, C_q)))^s)$
$= g^{s*t(y-x)}$

Figure 5-3: Description of the ElGamal scheme. The answer is 1 if at least single element in the two private sets is the same, otherwise the answer is a non-zero element which does not reveal any information about the elements in Y.

Computation Efficiency The computation efficiency of the user p and q remain the same; however, the constant time operation has been shifted to the server side in this protocol. The server S_2 performs the majority of the computation and the server S_1 performs the secret t distribution for the commitment; therefore, the computational cost for S_1 is $\mathcal{O}(1)$. The computational complexity of other participants of the protocol can be found in the Table 5.3.

Communication Efficiency In comparison to the previous protocol, the communication efficiency of p increases to $\mathcal{O}(|X| \cdot |Q| \cdot |Y|)$ due to the computation load shift from p to S_2. The server S_1 has the communication upper bound of $\mathcal{O}(|P| \cdot |X| + |Q| \cdot |Y|)$

5.4.3 Using leveled and low depth FHE

Method: Unlike the previous two protocols, here we bring the notion of communication between two non-colluding servers in order to circumvent the attacks where a healthy person p colludes with one of the servers. This is a very practical attack setting since any server can behave in a semi-honest manner and still act as a healthy person by having an access to query the other server as a legitimate user. S_1 here is used for FHE computation and for storing the encrypted locations by the infected user. S_2 does the job of generating keys and performing partial decryption of the results and returning it to p.

Security This protocol improves the security coverage of the system by giving protection against the *client-server* collusion. The *server* collusion attack still remains possible.

Computation Efficiency The computational complexity of this protocol is dominated by the homomorphic multiplications. For a polynomial of degree n, we need $O(n)$ multiplications to raise the input to all n powers. The multiplicative depth of this circuit is $O(\log\log(n))$. This gives us a computational complexity of $O\left(n\left(\log\log(n)\right)^2\right)$, since the multiplicative depth affects the size of the ciphertext for all multiplications.

Communication Efficiency The communication of this protocol consists of the client sending the powers of the fresh ciphertext to server 1, server 1 sending the result ciphertext to server 2, and server 2 returning the masked result to the client. Suppose a ciphertext can support ℓ slots. The amortized size of the initial message across the ℓ values is $O(\rho \cdot \log\log(n))$, where ρ is a parameter of the homomorphic encryption

INPUTS: Two sets X and Y.
PARAMETERS:
REGISTRATION PHASE:

1. Server 2 chooses a key pair for packed FHE encryption $(\mathsf{pk}_q, \mathsf{sk}_q)$ and sends pk_q to q.
2. q encrypts each location $y_i, c_i = \mathsf{Encrypt}(\mathsf{pk}_q, y_i)$ and sends the cipher text to Server 1 (q can go completely offline).

REQUEST PHASE:

1. p receives pk_q from Server 2
2. p encrypts his location $d = \mathsf{Encrypt}(\mathsf{pk}, x)$ and $R = \mathsf{Encrypt}(\mathsf{pk}, r)$ for random r and sends (d, R) to Server 1
3. Server 1 computes $ct_i = c_i - d = \mathsf{Encrypt}(\mathsf{pk}, y_i) - \mathsf{Encrypt}(\mathsf{pk}, x)$ and $CT = \mathsf{Encrypt}(\mathsf{pk}, s) * ct_1 * ct_2 * \ldots * ct_{|Y|} + R = \mathsf{Encrypt}(s((x-y_1)*\ldots*(x-y_{|Y|})) + R)$ for random s and sends C_T to server 2 (C_T is the sum of ALL the infected locations).
4. S_2 decrypts $\mathsf{Decrypt}(C_T, \mathsf{sk}_q)$ and gets either $0+R$ or $random+R$ and sends the answer to p
5. p checks if solution $= R$

Figure 5-4: Description of the FHE based protocol.

scheme that represents the overhead for each multiplication. The amortized size of the message server 1 sends to server 2 is $O(\rho)$. The amortized size of the message server 2 sends to the client is $O(1)$. Overall, this gives an amortized communication complexity of $O(\rho \cdot \log\log(n)) = O(\log\log(n))$ for a constant ρ.

Multiplicative depth of FHE: Server 1 homomorphically evaluates the polynomial $F(x, Y) = s$ of degree $|Y|$. Naive homomorphic computation of this polynomial would require depth $O(log|Y|)$. We can reduce the depth as follows: p sends the encrypted coefficients of the polynomial based on

$$x : d_j = Enc(x^{2^j}) \quad \forall \ 0 \leq j \leq log|Y|$$

Attack \ Method	Hashing	El-gamal	FHE
Malicious App	✓	✓	✓
Malicious Server	✓	✓	✓
Collusion S1 and S2	x	x	x
Collusion P and S1	x	x	✓
Collusion P and S2	x	✓	✓

Table 5.2: Table for comparing different proposed methods for protection against different attacks described in the attacker taxonomy.

, i.e., encryptions of x, x^2, x^4 etc... Server 1 computes the encrypted polynomial coefficients which are based on Y using c_i received in step iv)* and it also computes the encryptions of all the powers of x (i.e., encryptions of x^3, x^5, x^6 etc...) using the binary representation of the exponents and d_j Server 1 evaluates the polynomial using d_j and c_i The depth of the FHE is reduced to $log(log|Y|)$. Moreover, if we batch the locations of q in buckets of size n then the depth is $log(log(|Y|/n))$.

5.5 Discussion

5.5.1 Systems solution to the security

The proposed protocols use cryptographic primitives to protect user's data from the server while allowing computation on it. However, they do not protect against brute-force attacks that can be performed by a malicious client. Bruteforce attacks are practical in our setup due to the low entropy of spatio-temporal data. Any cryptographic tool will not prevent such an attack because it is indistinguishable from a generic user query. Therefore, we propose to limit such number of queries to a number that would be functionally normal for an honest user but prevents dishonest users from querying extra data. Furthermore, we employ additional layers of security at the system layer such as safety nets [2, 1] that perform App verification and device verification to ensure that a dishonest user can not make queries from an original app and original signed-in device. Note that these defense mechanisms only alleviate the bruteforce attack issue but do not eliminate it in its entirety.

5.5.2 Efficiency

The communication efficiency of the system for different protocols and different entities is described in table 5.4. The hashing protocol requires a considerable bandwidth with the server S_2 because it is proportionate to the total number of infected users ($|Q|$) which could be very large in a practical scenario.
The computation efficiency is described in Table 5.3. In general, leveled FHE provides the best computation complexity because the sets are shared as polynomials which can be evaluated more efficiently than iterating over each and every element in the set as done in the remaining two protocols. The computational task is heavily offloaded to the client p in the hashing protocol while the other two protocols offload it to the servers. Therefore, the choice of algorithm might depend upon the real world hardware setup. For the application of digital contact tracing, the goal should be to keep minimal computation on the client side since the device performing the computation is a mobile phone in comparison to the servers which could have better compute capabilities.

5.5.3 Security of the proposed protocols

Bruteforce attacks We assume the servers and clients follow the honest but curious model where each entity tries to obtain the maximum information from the available and encrypted data and keys but does not perform or provide any incorrect output. In Table 5.2 we compare all the proposed protocols against different possible attacks.

Attacks described by Dar et. al. [38]

Jamming Attack Protecting against physical hardware attack is not in the scope of this work. However, it is possible to use a notch filtering mechanism or solutions like BLISS for protection against GPS jammers.

Spoofing Attack Like the previous attack, this is also a hardware based attack where the attacker is trying to interfere in the received signal such that spurious locations are received. A solution to this attack also requires dedicated hardware for detecting interference or a smart GPS capture algorithm which can detect an anomaly in the single read.

Resource Drain Attack Our mode of operation mandates the use of only 288 (every 5 minutes for 24 hour) GPS points in a single day, hence when a user uploads their data it consists of no more than 14 days of history and any violation of this rule will essentially result in the packet drop.

Trolling Attack We propose to build the system with an audit mechanism. In cases when there is a suspicion of potential trolling, government authorities can inspect the individual's point by requesting their keys and verifying whether there is any misuse or not. Furthermore, we tightly couple the App ecosystem with secure upload so that no one can reverse engineer the app and even if they do they should not be able to modify the GPS readings which they upload.

Proximity App Attack Similar to the attack above, this attack also is only possible in Bluetooth based systems and not GPS because no location information is shared apriori.

Tracking and Deanonymization Attacks Since there is no identity exchange between any individual, the only information which can be deanonymized is location information. Most places can have multiple individuals at the same time; which makes this attack infeasible as it would require distinguishing a particular user from a group of users which can be present at a given geolocation. For sensitive locations such as a house, we propose to perform user-level redaction so that it does not leave any individual's device.

Method / User	Hashing	ElGamal	Leveled FHE
Healthy Person (P)	$\mathcal{O}(\|Q\| \cdot \|Y\| \log(\|Q\| \cdot \|Y\|))$	$\mathcal{O}(\|X\| \cdot \|Q\| \cdot \|Y\|)$	$\mathcal{O}(\|X\|)$
Infected Person (Q)	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(\|Y\|)$
Server 1	$\mathcal{O}(\|P\| \cdot \|X\| + \|Q\| \cdot \|Y\|)$	$\mathcal{O}(1)$	$\mathcal{O}(\|P\| \cdot \|Q\|)$
Server 2	$\mathcal{O}(1)$	$\mathcal{O}(\|P\| \cdot \|X\| \cdot \|Q\| \cdot \|Y\|)$	$\mathcal{O}(\|P\| \cdot \|Q\|)$

Table 5.3: Table for comparing different proposed methods for the computational efficiency.

Method / User	Hashing	ElGamal	leveled FHE
P and S1	$\mathcal{O}(\|X\|)$	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(\|P\| \cdot log(\|X\|))$
P and S2	$\mathcal{O}(\|X\| \cdot \|Q\| \cdot \|Y\|)$	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(1)$
Q and S1	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(1)$	$\mathcal{O}(\|Q\| \cdot \|Y\|)$
Q and S2	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(\|Y\|)$	$\mathcal{O}(1)$
Between S1 and S2	N/A	N/A	$\mathcal{O}(\|P\| \cdot log(\|X\|) \cdot \|Q\|)$

Table 5.4: Table for comparing different proposed methods for the communication efficiency. N/A refers to the instances when there is no communication between two systems.

False Injection or False Report Attack There is only one path of communication between an attacker and individual. This makes it difficult for an attacker to intrude in-between. For every communication, our assumption is that traffic will be protected end to end using presentation layer security such as SSL.

5.6 Conclusion

In this chapter we discuss the problem of private set intersection with a focus on contact tracing as an application. While conventional methods have focused on bluetooth based options that allow arbitrary large entropy in the random tokens, our proposed schemes focus on privately computing intersections of GPS trails that have limited entropy. This results in a different system design as well as different security constraints in comparison to the bluetooth based approaches. We propose three techniques that leverage privacy homomorphisms and multi-server protocols to securely compute the intersection between two set of users. We elucidate all three key trade-offs of these

techniques - Privacy, Communication and Computation efficiency.

5.7 Future Work

The proposed protocols in this work require iterating over every infected user's dataset. However, a secure approach to aggregating the infected user's spatio-temporal data can reduce the size of the whole dataset significantly and hence improve the overall performance. One of the fundamental drawbacks for GPS-based contact tracing is the attacker's capability to perform a brute force attack. While we circumvent this attack through rate limiting and a systems level solution discussed in section 5.5.1, an algorithmic solution can address this problem in a more secure manner. Finally, privacy in the context of private set intersection usually aims at secure matching of two sets, but does not satisfy the definition of differential privacy. Hence, exploring differentially private set intersection protocols could be an interesting direction.

Bibliography

[1] *Apple Device Check*.

[2] *Safety Net Device attestation*.

[3] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2016.

[4] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[5] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[6] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)*, 51(4):1–35, 2018.

[7] Nabil R Adam and John C Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989.

[8] Sai Aparna Aketi, Sourjya Roy, Anand Raghunathan, and Kaushik Roy. Gradual channel pruning while training using feature relevance scores for convolutional neural networks. abs/2002.09958, 2020.

[9] Jesslyn Alekseyev, Erica Dixon, Vilhelm L. Andersen Woltz, and Danny Weitzner. Realizing the promise of automated exposure notification (AEN) technology to control the spread of COVID-19: Recommendations for smartphone app deployment, use, and iterative assessment.

[10] Jonatan Almagor and Stefano Picascia. Exploring the effectiveness of a COVID-19 contact tracing app using an agent-based model. 10(1):22235.

[11] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[12] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[13] Nick Angelou, Ayoub Benaissa, Bogdan Cebere, William Clark, Adam James Hall, Michael A. Hoeh, Daniel Liu, Pavlos Papadopoulos, Robin Roehm, Robert Sandmann, Phillipp Schoppmann, and Tom Titcombe. Asymmetric private set intersection with applications to contact tracing and private vertical federated machine learning.

[14] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. Why are deep nets reversible: A simple theory, with implications for training. *CoRR*, abs/1511.05653, 2015.

[15] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[16] Martin Bertran, Natalia Martinez, Afroditi Papadaki, Qiang Qiu, Miguel Rodrigues, Galen Reeves, and Guillermo Sapiro. Adversarially learned representations for information obfuscation and inference. In *International Conference on Machine Learning*, pages 614–623. PMLR, 2019.

[17] Kishor Bharti, Tobias Haug, Vlatko Vedral, and Leong-Chuan Kwek. Machine learning meets quantum foundations: A brief survey. *AVS Quantum Science*, 2(3):034101, 2020.

[18] Binod Bhattarai, Seungryul Baek, Rumeysa Bodur, and Tae-Kyun Kim. Sampling strategies for gan synthetic data. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2303–2307. IEEE, 2020.

[19] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[20] Dan Boneh. The decision diffie-hellman problem. In *International Algorithmic Number Theory Symposium*, pages 48–63. Springer, 1998.

[21] Joppe W Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In *IMA International Conference on Cryptography and Coding*, pages 45–64. Springer, 2013.

[22] Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee. Fhe circuit privacy almost for free. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 62–89, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[23] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. *Proceedings of Advances in Cryptology-Crypto*, 7417, 08 2012.

[24] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[25] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[26] Eva Cetinic and James She. Understanding and creating art with ai: Review and outlook. *arXiv preprint arXiv:2102.09109*, 2021.

[27] Qi Chang, Hui Qu, Yikai Zhang, Mert Sabuncu, Chao Chen, Tong Zhang, and Dimitris N Metaxas. Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13856–13866, 2020.

[28] Thee Chanyaswad, Alex Dytso, H Vincent Poor, and Prateek Mittal. Mvg mechanism: Differential privacy under matrix-valued query. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 230–246, 2018.

[29] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[30] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[31] Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1243–1255, 2017.

[32] Xiang Cheng, Hanchao Yang, Archanaa S. Krishnan, Patrick Schaumont, and Yaling Yang. KHOVID: Interoperable privacy preserving digital contact tracing.

[33] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

[34] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

[35] Pai-Cheng Chu. Cell suppression methodology: The importance of suppressing marginal totals. *IEEE transactions on knowledge and data engineering*, 9(4):513–523, 1997.

[36] Fabrizio Cicala, Weicheng Wang, Tianhao Wang, Ninghui Li, Elisa Bertino, Faming Liang, and Yang Yang. PURE: A framework for analyzing proximity-based contact tracing protocols.

[37] Lawrence H Cox. Suppression methodology and statistical disclosure control. *Journal of the American Statistical Association*, 75(370):377–385, 1980.

[38] Aaqib Bashir Dar, Auqib Hamid Lone, Saniya Zahoor, Afshan Amin Khan, and Roohie Naaz. Applicability of mobile contact tracing in fighting pandemic (covid-19): Issues, challenges and solutions. Cryptology ePrint Archive, Report 2020/484, 2020. `https://eprint.iacr.org/2020/484`.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[40] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jürgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Large scale holistic video understanding. In *European Conference on Computer Vision*, pages 593–610. Springer, 2020.

[41] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6192–6201, 2019.

[42] Alexey Dosovitskiy and Thomas Brox. Inverting convolutional networks with convolutional networks. *CoRR*, abs/1506.02753, 2015.

[43] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *arXiv preprint arXiv:1602.02644*, 2016.

[44] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016.

[45] Stelios Doudalis, Ios Kotsogiannis, Samuel Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. One-sided differential privacy. *arXiv preprint arXiv:1712.05888*, 2017.

[46] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[47] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[48] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

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

[50] Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2), 2010.

[51] Rolf Egert, Marc Fischlin, David Gens, Sven Jacob, Matthias Senker, and Jörn Tillmanns. Privately computing set-union and set-intersection cardinality via bloom filters. In *Australasian Conference on Information Security and Privacy*, pages 413–430. Springer, 2015.

[52] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. *arXiv preprint arXiv:1808.06640*, 2018.

[53] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 10–18, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[54] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[55] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2(2-3), 2017.

[56] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the*

Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '03, page 211–222, New York, NY, USA, 2003. Association for Computing Machinery.

[57] Sky Justin Faber. *Variants of Privacy Preserving Set Intersection and their Practical Applications*. PhD thesis, UC Irvine, 2016.

[58] Ronald Fagin, Moni Naor, and Peter Winkler. Comparing information without leaking it. *Commun. ACM*, 39(5):77–85, May 1996.

[59] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption, 2012.

[60] Matthew Feickert and Benjamin Nachman. A living review of machine learning for particle physics. *arXiv preprint arXiv:2102.02770*, 2021.

[61] Andrew Ferraiuolo, Andrew Baumann, Chris Hawblitzel, and Bryan Parno. Komodo: Using verification to disentangle secure-enclave hardware from software. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 287–305, 2017.

[62] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[63] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 1–19, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[64] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273, 2008.

[65] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng zhong Xu. Dynamic channel pruning: Feature boosting and suppression. In *International Conference on Learning Representations*, 2019.

[66] Craig Gentry. Fully homomorphic encryption using ideal lattices. 2009.

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

[68] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[69] Google. Apple and google: Exposure notifications: Using technology to help public health authorities fight covid-19, 2020. `https://www.google.com/covid19/exposurenotifications`.

[70] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *CoRR*, abs/1810.06060, 2018.

[71] Jihun Hamm. Minimax filter: Learning to preserve privacy from inference attacks. *Journal of Machine Learning Research*, 18(129):1–31, 2017.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[73] Xi He, Ashwin Machanavajjhala, and Bolin Ding. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1447–1458, 2014.

[74] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*, 2017.

[75] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[76] Jan Holvast. History of privacy. In *The History of Information Security*, pages 737–769. Elsevier, 2007.

[77] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[78] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Context-aware generative adversarial privacy. *Entropy*, 19(12):656, 2017.

[79] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Generative adversarial privacy. *CoRR*, abs/1807.05306, 2018.

[80] Bernardo A. Huberman, Matt Franklin, and Tad Hogg. Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, page 78–86, New York, NY, USA, 1999. Association for Computing Machinery.

[81] Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In Salil P. Vadhan, editor, *Theory of Cryptography*, pages 575–594, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[82] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015.

[83] Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

[84] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in neural information processing systems*, pages 667–675, 2016.

[85] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2018.

[86] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1651–1669, 2018.

[87] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

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

[89] Fumiyuki Kato, Yang Cao, and Yoshikawa Masatoshi. PCT-TEE: Trajectory-based private contact tracing system with trusted execution environment.

[90] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[91] Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):1–36, 2014.

[92] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[93] Tae-hoon Kim, Dongmin Kang, Kari Pulli, and Jonghyun Choi. Training with the invisibles: Obfuscating images to share safely for learning visual recognition models. *arXiv preprint arXiv:1901.00098*, 2019.

[94] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[95] Ágnes Kiss, Jian Liu, Thomas Schneider, N Asokan, and Benny Pinkas. Private set intersection for unequal set sizes with mobile applications. *Proceedings on Privacy Enhancing Technologies*, 2017(4):177–197, 2017.

[96] Benjamin Klein, Lior Wolf, and Yehuda Afek. A dynamic convolutional layer for short range weather prediction. In *CVPR*, 2015.

[97] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[98] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

[99] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

[100] Kimmo Kärkkäinen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age, 2019.

[101] Douglas J Leith and Stephen Farrell. Measurement-based evaluation of google/apple exposure notification api for proximity detection in a light-rail tram. *Plos one*, 15(9):e0239943, 2020.

[102] Ang Li, Yixiao Duan, Huanrui Yang, Yiran Chen, and Jianlei Yang. Tiprdc: task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 824–832, 2020.

[103] Ang Li, Yixiao Duan, Huanrui Yang, Yiran Chen, and Jianlei Yang. Tiprdc: Task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and; Data Mining*, KDD '20, page 824–832, New York, NY, USA, 2020. Association for Computing Machinery.

[104] Ang Li, Jiayi Guo, Huanrui Yang, and Yiran Chen. Deepobfuscator: Adversarial training framework for privacy-preserving image classification, 2019.

[105] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[106] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification, 2018.

[107] Jiachun Liao, Oliver Kosut, Lalitha Sankar, and Flavio P Calmon. A tunable measure for information leakage. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 701–705. IEEE, 2018.

[108] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Deeprotect: Enabling inference-based access control on mobile sensing applications. *CoRR*, abs/1702.06159, 2017.

[109] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, Jul 2017.

[110] Sicong Liu, Junzhao Du, Anshumali Shrivastava, and Lin Zhong. Privacy adversarial network: representation learning for mobile data privacy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–18, 2019.

[111] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

[112] Po-Shen Loh. Flipping the perspective in contact tracing. *arXiv preprint arXiv:2010.03806*, 2020.

[113] Ted Londner, Jonathan Saunders, Dieter W. Schuldt, and Bill Streilein. SimAEN – simulated automated exposure notification.

[114] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 144–155, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[115] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[116] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. *CoRR*, abs/1412.0035, 2014.

[117] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[118] Gary F Marcus. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press, 2019.

[119] Tania Martin, Georgios Karopoulos, José L. Hernández-Ramos, Georgios Kambourakis, and Igor Nai Fovino. Demystifying COVID-19 digital contact tracing: A survey on frameworks and mobile apps.

[120] John Martinsson, Edvin Listo Zec, Daniel Gillblad, and Olof Mogren. Adversarial representation learning for synthetic replacement of private attributes. *arXiv preprint arXiv:2006.08039*, 2020.

[121] H McMahan. Brendan et al."communication-efficient learning of deep networks from decentralized data.". *Proceedings of the 20th AISTATS*, 2016.

[122] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[123] John Meklenburg, Michael Specter, Michael Wentz, Hari Balakrishnan, Anantha Chandrakasan, John Cohn, Gary Hatke, Louise Ivers, Ronald Rivest, Gerald Jay Sussman, and Daniel Weitzner. SonicPACT: An ultrasonic ranging method for the private automated contact tracing (PACT) protocol.

[124] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *arXiv preprint arXiv:1705.10461*, 2017.

[125] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Prakash Ramrakhyani, Dean M. Tullsen, and Hadi Esmaeilzadeh. Shredder: Learning noise to protect privacy with partial DNN inference on the edge. *CoRR*, abs/1905.11814, 2019.

[126] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[127] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021.

[128] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[129] Karthik Nandakumar, Nalini Ratha, Sharath Pankanti, and Shai Halevi. Towards deep neural network training on encrypted data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[130] Seong Joon Oh, Mario Fritz, and Bernt Schiele. Adversarial image perturbation for privacy protection a game theory perspective. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1491–1500. IEEE, 2017.

[131] Graziella Orrù, Merylin Monaro, Ciro Conversano, Angelo Gemignani, and Giuseppe Sartori. Machine learning in psychometrics and psychological research. *Frontiers in psychology*, 10:2970, 2020.

[132] Michele Orrù, Emmanuela Orsini, and Peter Scholl. Actively secure 1-out-of-n ot extension with application to private set intersection. In *Cryptographers' Track at the RSA Conference*, pages 381–396. Springer, 2017.

[133] Seyed Ali Osia, Ali Shahin Shamsabadi, Sina Sajadmanesh, Ali Taheri, Kleomenis Katevas, Hamid R Rabiee, Nicholas D Lane, and Hamed Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *IEEE Internet of Things Journal*, 7(5):4505–4518, 2020.

[134] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

[135] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Spot-light: Lightweight private set intersection from sparse ot extension. In *Annual International Cryptology Conference*, pages 401–431. Springer, 2019.

[136] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. Phasing: Private set intersection using permutation-based hashing. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 515–530, Washington, D.C., August 2015. USENIX Association.

[137] Benny Pinkas, Thomas Schneider, and Michael Zohner. Faster private set intersection based on {OT} extension. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 797–812, 2014.

[138] Atul Pokharel, Robert Soulé, and Avi Silberschatz. A case for location based contact tracing.

[139] Manoj M Prabhakaran and Amit Sahai. *Secure multi-party computation*, volume 10. IOS press, 2013.

[140] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[141] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 192–203, 2016.

[142] Ramesh Raskar, Ranu Dhillon, Suraj Kapa, Deepti Pahwa, Renaud Falgas, Lagnojita Sinha, Aarathi Prasad, Abhishek Singh, Andrea Nuzzo, Rohan Iyer, and Vivek Sharma. Comparing manual contact tracing and digital contact advice.

[143] Sofya Raskhodnikova, Adam Smith, Homin K Lee, Kobbi Nissim, and Shiva Prasad Kasiviswanathan. What can we learn privately. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pages 531–540, 2008.

[144] Suman V. Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *CoRR*, abs/1905.10887, 2019.

[145] Jens Helge Reelfs, Oliver Hohlfeld, and Ingmar Poese. Corona-warn-app: Tracing the start of the official COVID-19 exposure notification app for germany.

[146] Peter Rindal and Mike Rosulek. Improved private set intersection against malicious adversaries. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 235–259. Springer, 2017.

[147] Peter Rindal and Mike Rosulek. Malicious-secure private set intersection via dual execution. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1229–1242, 2017.

[148] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[149] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[150] Proteek Chandan Roy and Vishnu Naresh Boddeti. Mitigating information leakage in image representations: A maximum entropy approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[151] Bashir Sadeghi, Runyi Yu, and Vishnu Boddeti. On the global optima of kernelized adversarial representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7971–7979, 2019.

[152] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[153] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1–36, 2019.

[154] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019.

[155] Vivek Sharma, Ali Diba, Davy Neven, Michael S Brown, Luc Van Gool, and Rainer Stiefelhagen. Classification-driven dynamic image enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4033–4041, 2018.

[156] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[157] Abhishek Singh, Ayush Chopra, Vivek Sharma, Ethan Garza, Emily Zhang, Praneeth Vepakomma, and Ramesh Raskar. Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks. *arXiv preprint arXiv:2012.11025*, 2020.

[158] Abhishek Singh, Praneeth Vepakomma, Otkrist Gupta, and Ramesh Raskar. Detailed comparison of communication efficiency of split learning and federated learning. *arXiv preprint arXiv:1909.09145*, 2019.

[159] Priyanka Singh, Abhishek Singh, Gabriel Cojocaru, Praneeth Vepakomma, and Ramesh Raskar. PPContactTracing: A privacy-preserving contact tracing protocol for COVID-19 pandemic.

[160] Pietro Tedeschi, Spiridon Bakiras, and Roberto Di Pietro. SpreadMeNot: A provably secure and privacy-preserving contact tracing protocol.

[161] Takehiro Tezuka, Lihua Wang, Takuya Hayashi, and Seiichi Ozawa. A fast privacy-preserving multi-layer perceptron using ring-lwe-based homomorphic encryption. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 37–44. IEEE, 2019.

[162] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, and Seyit Camtepe. Splitfed: When federated learning meets split learning. *arXiv preprint arXiv:2004.12088*, 2020.

[163] Amirsina Torfi, Rouzbeh A Shirvani, Yaser Keneshloo, Nader Tavvaf, and Edward A Fox. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020.

[164] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[165] Ni Trieu, Kareem Shehata, Prateek Saxena, Reza Shokri, and Dawn Song. Epione: Lightweight contact tracing with strong privacy, 2020.

[166] Amee Trivedi and Deepak Vasisht. Digital contact tracing: Technologies, shortcomings, and the path forward. 50(4):75–81.

[167] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. *CoRR*, abs/1711.10925, 2017.

[168] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[169] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning, 2020.

[170] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[171] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[172] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou. Learning Discriminative Features with Multiple Granularities for Person Re-Identification. *ArXiv e-prints*, April 2018.

[173] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8919–8928, 2020.

[174] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[175] Samuel Warren and Louis Brandeis. *The right to privacy*. Columbia University Press, 1890.

[176] Sarah Webb. Deep learning for biology. *Nature*, 554(7693), 2018.

[177] Bingzhe Wu, Shiwan Zhao, Guangyu Sun, Xiaolu Zhang, Zhong Su, Caihong Zeng, and Zhihong Liu. P3sgd: Patient privacy preserving sgd for regularizing deep cnns in pathological image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2099–2108, 2019.

[178] Liyao Xiang, Haotian Ma, Hao Zhang, Yifan Zhang, Jie Ren, and Quanshi Zhang. Interpretable complex-valued neural networks for privacy protection. *arXiv preprint arXiv:1901.09546*, 2019.

[179] LC Yan, B Yoshua, and H Geoffrey. Deep learning. *nature*, 521(7553):436–444, 2015.

[180] Dong Yi, Zhen Lei, and Stan Z. Li. Deep metric learning for practical person re-identification, 2014.

[181] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks (2013). *arXiv preprint arXiv:1311.2901*, 2013.

[182] Fan Zhang, Ziyuan Liang, Cong Zuo, Jun Shao, Jianting Ning, Jun Sun, Joseph K Liu, and Yibao Bao. hpress: A hardware-enhanced proxy re-encryption scheme using secure enclave. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.

[183] Liming Zhao, Xi Li, Jingdong Wang, and Yueting Zhuang. Deeply-learned part-aligned representations for person re-identification, 2017.

[184] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[185] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 875–886. Curran Associates, Inc., 2018.