UNDERSTANDING AND CONTROLLING LEAKAGE IN MACHINE LEARNING

A dissertation submitted towards the degree
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Tribhuvanesh Orekondy, M.Sc.

Saarbrücken
2020

Day of Colloquium	18th December, 2020
Dean of the Faculty	Univ.-Prof. Dr. Thomas Schuster Saarland University, Germany

Examination Committee

Chair	Prof. Dr. Isabel Valera
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer, Co-advisor	Prof. Dr. Mario Fritz
Reviewer	Prof. Dr. Jan-Michael Frahm
Reviewer	Prof. Apu Kapadia, PhD
Academic Assistant	Dr. Paul Swoboda

ABSTRACT

Machine learning models are being increasingly adopted in a variety of real-world scenarios. However, the privacy and confidentiality implications introduced in these scenarios are not well understood. Towards better understanding such implications, we focus on scenarios involving interactions between numerous parties prior to, during, and after training relevant models. Central to these interactions is sharing information for a purpose e.g., contributing data samples towards a dataset, returning predictions via an API. This thesis takes a step toward understanding and controlling leakage of private information during such interactions.

In the first part of the thesis we investigate leakage of private information in visual data and specifically, photos representative of content shared on social networks. There is a long line of work to tackle leakage of personally identifiable information in social photos, especially using face- and body-level visual cues. However, we argue this presents only a narrow perspective as images reveal a wide spectrum of multimodal private information (e.g., disabilities, name-tags). Consequently, we work towards a Visual Privacy Advisor that aims to holistically identify and mitigate private risks when sharing social photos.

In the second part, we address leakage during training of ML models. We observe learning algorithms are being increasingly used to train models on rich decentralized datasets e.g., personal data on numerous mobile devices. In such cases, information in the form of high-dimensional model parameter updates are anonymously aggregated from participating individuals. However, we find that the updates encode sufficient identifiable information and allows them to be linked back to participating individuals. We additionally propose methods to mitigate this leakage while maintaining high utility of the updates.

In the third part, we discuss leakage of confidential information during inference time of black-box models. In particular, we find models lend themselves to model functionality stealing attacks: an adversary can interact with the black-box model towards creating a replica 'knock-off' model that exhibits similar test-set performances. As such attacks pose a severe threat to the intellectual property of the model owner, we also work towards effective defenses. Our defense strategy by introducing bounded and controlled perturbations to predictions can significantly amplify model stealing attackers' error rates.

In summary, this thesis advances understanding of privacy leakage when information is shared in raw visual forms, during training of models, and at inference time when deployed as black-boxes. In each of the cases, we further propose techniques to mitigate leakage of information to enable wide-spread adoption of techniques in real-world scenarios.

ZUSAMMENFASSUNG

Modelle für maschinelles Lernen werden zunehmend in einer Vielzahl realer Szenarien eingesetzt. Die in diesen Szenarien vorgestellten Auswirkungen auf Datenschutz und Vertraulichkeit wurden jedoch nicht vollständig untersucht. Um solche Implikationen besser zu verstehen, konzentrieren wir uns auf Szenarien, die Interaktionen zwischen mehreren Parteien vor, während und nach dem Training relevanter Modelle beinhalten. Das Teilen von Informationen für einen Zweck, z. B. das Einbringen von Datenproben in einen Datensatz oder die Rückgabe von Vorhersagen über eine API, ist zentral für diese Interaktionen. Diese Arbeit verhilft zu einem besseren Verständnis und zur Kontrolle des Verlusts privater Informationen während solcher Interaktionen.

Im ersten Teil dieser Arbeit untersuchen wir den Verlust privater Informationen bei visuellen Daten und insbesondere bei Fotos, die für Inhalte repräsentativ sind, die in sozialen Netzwerken geteilt werden. Es gibt eine lange Reihe von Arbeiten, die das Problem des Verlustes persönlich identifizierbarer Informationen in sozialen Fotos angehen, insbesondere mithilfe visueller Hinweise auf Gesichts- und Körperebene. Wir argumentieren jedoch, dass dies nur eine enge Perspektive darstellt, da Bilder ein breites Spektrum multimodaler privater Informationen (z. B. Behinderungen, Namensschilder) offenbaren. Aus diesem Grund arbeiten wir auf einen Visual Privacy Advisor hin, der darauf abzielt, private Risiken beim Teilen sozialer Fotos ganzheitlich zu identifizieren und zu minimieren.

Im zweiten Teil befassen wir uns mit Datenverlusten während des Trainings von ML-Modellen. Wir beobachten, dass zunehmend Lernalgorithmen verwendet werden, um Modelle auf umfangreichen dezentralen Datensätzen zu trainieren, z. B. persönlichen Daten auf zahlreichen Mobilgeräten. In solchen Fällen werden Informationen von teilnehmenden Personen in Form von hochdimensionalen Modellparameteraktualisierungen anonym verbunden. Wir stellen jedoch fest, dass die Aktualisierungen ausreichend identifizierbare Informationen codieren und es ermöglichen, sie mit teilnehmenden Personen zu verknüpfen. Wir schlagen zudem Methoden vor, um diesen Datenverlust zu verringern und gleichzeitig die hohe Nützlichkeit der Aktualisierungen zu erhalten.

Im dritten Teil diskutieren wir den Verlust vertraulicher Informationen während der Inferenzzeit von Black-Box-Modellen. Insbesondere finden wir, dass sich Modelle für die Entwicklung von Angriffen, die auf Funktionalitätsdiebstahl abzielen, eignen: Ein Gegner kann mit dem Black-Box-Modell interagieren, um ein Replikat-Knock-Off-Modell zu erstellen, das ähnliche Test-Set-Leistungen aufweist. Da solche Angriffe eine ernsthafte Bedrohung für das geistige Eigentum des Modellbesitzers darstellen, arbeiten wir auch an einer wirksamen Verteidigung. Unsere Verteidigungsstrategie

durch die Einführung begrenzter und kontrollierter Störungen in Vorhersagen kann die Fehlerraten von Modelldiebstahlangriffen erheblich verbessern.

Zusammenfassend lässt sich sagen, dass diese Arbeit das Verständnis von Datenschutzverlusten beim Informationsaustausch verbessert, sei es bei rohen visuellen Formen, während des Trainings von Modellen oder während der Inferenzzeit von Black-Box-Modellen. In jedem Fall schlagen wir ferner Techniken zur Verringerung des Informationsverlusts vor, um eine weit verbreitete Anwendung von Techniken in realen Szenarien zu ermöglichen.

CONTENTS

1

INTRODUCTION

There is an increasing push towards adopting machine learning systems to perform a range of real-world tasks e.g., agents capable of sensing, understanding, and interacting with the world. A standard notion to determine whether an ML system is capable of performing these tasks is by evaluating its generalization error i.e., how accurate the model is on previously unseen test data. Can a system be confidently deployed into the real-world based solely on low generalization error? We argue that in spite of low error, many open questions remain that hinders widespread adoption of ML systems: What is the predictive behaviour of the model on out-of-distribution examples? What is the rationale that led to a particular prediction? Can a malicious agent infer sensitive information by interacting with the system? It is crucial to work towards recognizing such challenges not captured by generalization errors, so as to enable *trustworthy* deployment of machine learning approaches in the real-world.

The thesis aims to improve understanding of trustworthiness of machine learning systems and in particular, their security, privacy, and confidentiality implications. These implications naturally arise since systems are rarely siloed, but rather involve complex interactions between various parties, from initial stages of gathering data to eventually deploying the model for making predictions. To motivate the interplay between parties, consider an internet service that uses machine learning techniques to manage (e.g., image search, face tagging) private photo collections of its users. Here, the interactions take the form of raw data (when users share tagged photos to the service) and predictions (when the service accurately suggests face tags). The primary focus of the thesis is scrutinizing the information exchange that occurs during such interactions at various stages of an ML system.

It is crucial to understand privacy and security implications surrounding machine learning techniques. Considering the example of the ML-based photo management service from before, a default assumption is a bi-directional trust between the service and its users. From the service's side, that they do not misuse the user data beyond what is required to provide meaningful predictions. In parallel, from the users' side, that they do not share malicious or tampered data to negatively influence the training of the model. Given the severely negative consequences (e.g., privacy violations) that entail when parties work outside the trust assumption, it is critical to preemptively identify risks and develop techniques to mitigate such risks.

In the specific case of information sharing, it is crucial that the information shared fulfills a particular objective (e.g., only making predictions) and does not leak any unnecessary private nor confidential information. This thesis takes a closer look and

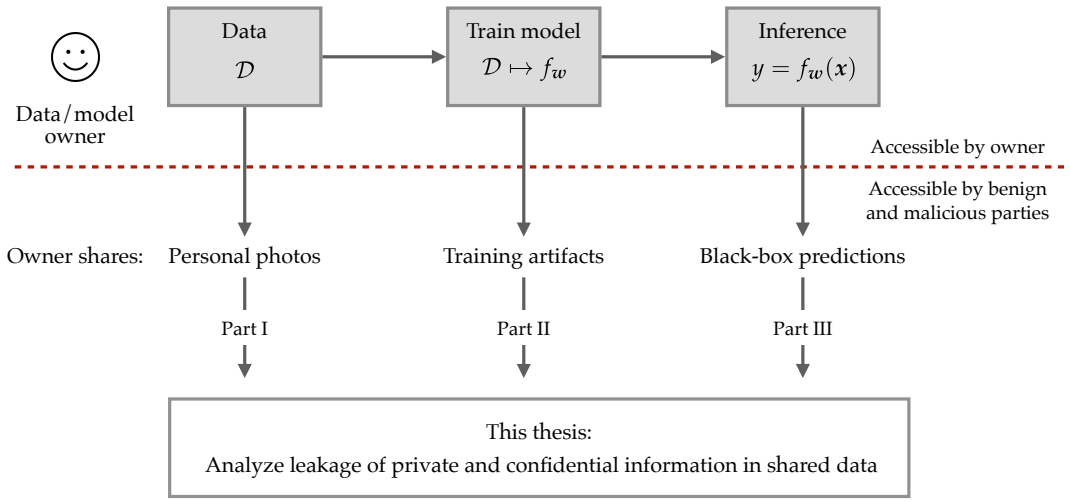


Figure 1.1: Overview of the three research directions investigated in this thesis.

finds that this is not entirely true. As a result, the goal of the thesis is to identify and prevent sources of leakage. To highlight this, the thesis investigates various parts of the pipeline. In Part I, we identify that raw visual data – most notably visual scenes resembling photos shared on social networks – reveals a wide spectrum of private information. Towards the goal of controlling this information we propose methods to identify and further obfuscate the corresponding pixels. In Part II, we focus on the training – for instance federated learning – where parameter updates are shared by participating individuals. We find that the updates reveal identity information and further propose methods to control this information. In Part III, our focus is the owner returning inference-time predictions, such as via black-box image prediction API. We observe that an API can be exploited to imitate the functionality and further propose ways to control it.

1.1 ANALYZING INFORMATION LEAKAGE IN MACHINE LEARNING

To provided a common basis for the research directions in the thesis, we consider a framework as shown in Figure 1.1. In this framework, we first consider an agent – the 'owner' – in possession of private and confidential data $\mathcal{D}$. The owner employs a typical machine learning pipeline: using data $\mathcal{D}$, a model f_w is trained and later used to return predictions at inference time.

Within this framework, we investigate the double-edged sword of sharing information relating to the data $\mathcal{D}$ or the model f_w. On the one hand, sharing certain pieces of information provides rewards. For instance, social (when sharing personal photos) or financial (when monetizing predictions) rewards. On the other hand, the shared information can also be exploited by malicious parties to achieve malicious

objectives such as recovering owner's personal details or confidential aspects of the owner's model.

In the following sections, we discuss the three research directions explored in the thesis. Each direction investigates on a specific source of leakage: personal photos (Part I), training artifacts (Part II), and predictions (Part III). We further highlight challenges when addressing the leakages and additionally place our contributions in the context of related literature.

1.1.1 *Analyzing Leakage in Visual Data*

One particular source that potentially leaks sensitive information is the raw data. Our primary focus is raw data comprising of visual scenes – representative of photos shared on social media – which captures complex relations between individuals, objects, and backgrounds from many viewpoints. Such scenes contrast other forms of visual data used in a privacy-context, such as frontal face images (single subject, good lighting) and surveillance footage (fixed viewpoint, low resolution). Understanding and controlling leakage of private information in visual data is particularly crucial in light of the massive amount of photos captured and disseminated on the internet everyday.

There is a long line of work that addresses privacy leaks in images. In the computer vision community, the notion of privacy in images is highly intertwined with identifiability of individuals. Studies indicate a variety of cues can be used to re-identify individuals such as facial features (Turk and Pentland, 1991; Parkhi et al., 2015), body parts (Bourdev and Malik, 2009; Oh et al., 2017), and clothing (Gallagher and Chen, 2008). However, apart from person identification cues, investigating complementary pieces of private information in images is limited. Some works have taken a promising first step by identifying that social relationships (Sun et al., 2017b), age (Liu et al., 2015), and gender (Wang et al., 2019) can also be inferred from images. However, there is scant research on identifying and controlling a wide spectrum of private information revealed in visual scenes (e.g., political opinions, occupation).

In parallel to studies that highlight privacy-sensitive information can be inferred in challenging scenarios, there is a good amount of literature dedicated to *preventing* such inferences. The primary focus of literature is manipulating images to prevent *automatic* person recognition. Recent works explore reducing effectiveness of recognition techniques using perceptible (Wilber et al., 2016) and imperceptible (Szegedy et al., 2013; Oh et al., 2017) image manipulations. Specific to manipulating images to prevent *automatic* person recognition, Wilber et al. (2016) explore effectiveness of various image manipulation strategies (e.g., adding noise, swirling). However, naturalness of images (the 'utility') is an important factor for sharing (obfuscated) images on social media. To obfuscate the image without significantly degrading its naturalness, recent works have explored leveraging adversarial perturbations (Oh et al., 2017) and advances in generative modelling (Sun et al., 2018a) to in-paint

facial identity, license plate, gender, ... | student ID, full name, age group, ... | full name, visited locations, timestamp, ... | fingerprints, ... | passport, credit card, identity card, ... | drivers license, full name, home address, ...

(a) Some examples of personal photos shared on the internet (on Flickr in this case). Such photos often contain a broad range of information; shown here using the privacy attribute taxonomy defined in Chapter 3. We additionally propose an approach to identify such attributes and evaluate privacy risks in visual content.

~~facial identity~~, ~~license plate~~, ~~gender~~, ... | ~~student ID~~, ~~full name~~, age group, ... | ~~full name~~, ~~timestamp~~, visited locations, ... | ~~fingerprints~~, ... | ~~passport~~, ~~credit card~~, ~~identity card~~, ... | ~~drivers license~~, ~~full name~~, ~~home address~~, ...

(b) In Chapter 4, we propose to further identify and redact privacy-sensitive regions in images, while preserving the utility of the image.

Figure 1.2: Analyzing leakage in visual data.

heads. While these approaches are promising, it is unclear whether they protect privacy against stronger adversaries (e.g., humans) and beyond facially-identifiable features (e.g., fingerprints).

Although literature to identify and control leakage of *person-recognition* features (especially faces) in visual scenes is abundant, we argue that the problem requires a broader perspective. Consider examples of publicly-available images as shown in Figure 1.2a. We enumerate some challenges in estimating privacy leakage in these examples: (i) *Personally Identifiable Information (PII) in visual content*: While notions of PII is well-established for explicit data (e.g., social user profile, health records), it is unclear what makes for image-specific PII, which we refer to as *visual privacy attributes*. Furthermore, only a narrow notion of privacy attributes has received attention in prior work e.g., facial identification features; (ii) *Visual Privacy Datasets*: Moreover, while datasets are abundant to enable recognition of specific privacy attributes (e.g., person identity, license plates), none capture all of them simultaneously; (iii) *Multimodality of Information*: Furthermore, identifying many attributes (e.g., student IDs) in images additionally requires reasoning over multi-modal content; (iv) *Utility*: Controlling privacy leakage in images, such as by obfuscation of selected content, comes at the price of impacting reducing the original utility. As a result, finding an optimal trade-off between privacy and utility is crucial to enable privacy-preserving photo sharing.

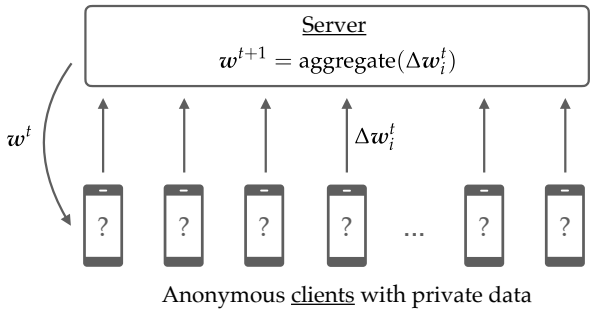


Figure 1.3: Analyzing leakage in learning algorithms. Some learning algorithms, such as Federated Averaging (McMahan et al., 2017), require anonymous clients (e.g., mobile devices) to intermittently share model updates w_i^t computed locally on private data. In Chapter 5, we investigate extent to which the updates leak user-identifiable information.

Contributions of the thesis. In Part I of the thesis we attempt to take a holistic view towards understanding and controlling privacy leakage in visual data by addressing the above challenges. We propose the first taxonomy of private attributes specific to visual content. Using these attributes, we build reasonably-sized image datasets of publicly-shared images annotated with image-level and pixel-level annotations. We leverage our datasets to work towards a 'Visual Privacy Advisor' (Chapter 3), which given an image, estimates the privacy risk in images taking into account the users' privacy preferences. In our follow-up work (Chapter 4), we extend our approach to additionally identify privacy-sensitive regions in images for obfuscation in a utility-constrained setting. Our work on Visual Privacy presents the first framework towards identifying and controlling leakage from a broad range sources in visual content.

1.1.2 *Analyzing Leakage during Learning*

Machine learning systems involve ingesting (potentially sensitive) raw training data and in many scenarios, also exposing certain *artifacts* when training the model. Identifying leakage from such artifacts (referred to as the 'attack surface') is crucial as it demonstrates vulnerabilities of ML systems from a privacy perspective. Studies indicate information related to training data can be recovered from artifacts such as activations (Yonetani et al., 2017), backpropagated gradients (Milli et al., 2018), and parameters (Nasr et al., 2019; Melis et al., 2019; Zhang et al., 2020). We specifically focus on information in the parameter space shared by the owner.

Federated Learning (McMahan et al., 2017) is a prototypical case where training artifacts are revealed during the training process. The main idea behind FL is to provide a framework to enable multiple data owners (e.g., hospitals) to collectively train a model using their private data (e.g., patient data) in a privacy-preserving manner. FL works on the principles of data minimization. Instead of the raw private

data, the owners contribute only the essence of the data relevant for training the model – the model parameter updates – as shown in Figure 1.3. Over multiple rounds, the server aggregates the parameter updates from a random anonymized subset of owners to improve a global model. The global model which is effectively learnt from the union of owners' private datasets is subsequently shared back to the owners. Consequently, as only the model parameter updates (the training artifact) are shared to untrusted parties, FL helps ensure confidentiality of the raw private data.

As model parameter updates in the presence of an untrusted server forms the primary channel of information sharing between data owners, it has received significant attention recently. In the specific case of modeling adversaries with access to the parameter updates, literature has explored adversarial objectives to maliciously *manipulate* the update, or alternatively *infer* private attributes. Manipulation objectives model clients as adversarial parties who wish to degrade the overall performance of the model (Bhagoji et al., 2019) or planting backdoors (Bagdasaryan et al., 2020; Xie et al., 2019). In contrast, extraction objectives involve an honest-but-curious server as the adversarial party who wishes to recover sensitive information from the model updates. Towards extraction goals, literature has investigated inferring membership of certain examples (Nasr et al., 2019), orthogonal sensitive attributes (Melis et al., 2019), and reconstructions under certain assumptions (Hitaj et al., 2017; Zhu et al., 2019). While there is some progress in understanding information leakage from model updates in FL scenarios, the full extent of leakage is still unclear.

Contributions of the thesis. Our work (Orekondy et al., 2020a) presented in Chapter 5 contributes to better understanding leakage in FL scenarios by investigating complementary extraction objectives. Specifically, we study deanonymization attacks, where a malicious server attempts to infer the identity from a model parameter update. Leakage of identities is problematic as it undermines existing de-identification mechanisms (McMahan et al., 2017; Hard et al., 2018). Such mechanisms help ensure a model parameter update is stripped of identifiable metadata before being made accessible to untrusted parties. Furthermore, this form of leakage adds to the risk of associating identities with other sensitive inferences (Melis et al., 2019) (e.g., gender). As we observe deanonymization risks is an artifact of inherent biases of individuals when generating data, we also study methods to mitigate these risks by adding adversarial biases to data.

1.1.3 *Analyzing Leakage in Black-box Models*

In the previous sections, we discussed information leakage *prior to* training a model (i.e., on raw data) and *during* the training (i.e., via intermediate parameters). Now we analyze information in a scenario *after* a model is trained i.e., at inference time. This scenario is becoming increasingly common as ML techniques are being

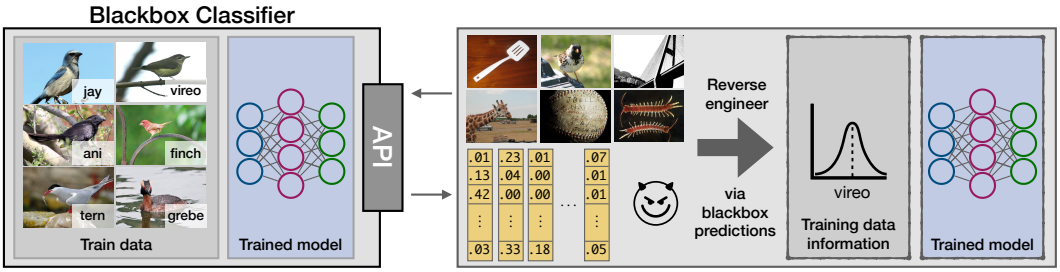


Figure 1.4: Analyzing leakage in black-box models. In Part III, we investigate leakage via black-box (left side of image) predictions. Specifically, we study how an adversary (right side of image) can potentially reverse-engineer certain confidential aspects of the model, such as its functionality.

deployed in the real-world such as on the internet (via prediction APIs), consumer electronics (e.g., smartphones), and healthcare products. Common to these scenarios is the owner obtaining a model by investing a large amount of effort, ranging from collecting and annotating a dataset to engineering a model's network architecture. The deployed model is typically accessed as a black-box to protect the intellectual property of these efforts. Given an input, the black-box interface to an ML model solely reveals the corresponding predictions while concealing all other information of the model (i.e., hyperparameters, training data). Concealing information outside of required predictions is crucial for the model owner to protect privacy of participating individuals in the training dataset and confidentiality of the intellectual property of the trained model.

However, recent studies indicate information about the training *data*, *gradients*, or *model* can be reverse-engineered solely using a black-box interface to trained models. Recovering each of these pieces of information often serve a different purpose. Approaches recovering information on the training *data* expose the privacy leakage such as by recovering membership of datapoints (Shokri et al., 2017; Salem et al., 2019) or attribute information (Fredrikson et al., 2015; Song and Shmatikov, 2020). In parallel, approaches (Chen et al., 2017; Guo et al., 2019) to recover *gradients* of predictions with respect to inputs threaten safety of the system, as the gradients are leveraged to craft adversarial examples. Complementing these are approaches (Lowd and Meek, 2005a) that extract information about the *model internals* (e.g., parameters) that compromise the confidentiality of the system.

For the rest of the section, we focus primarily on threats that exploit leakage of a confidential model's internal information by exploiting black-box access. Specifically, we focus on model stealing (or extraction) attacks (Lowd and Meek, 2005a; Tramèr et al., 2016), where an adversary exploits the black-box access to create a replica of the model. When performed sample-efficiently, such attacks raise severe concerns as the owner's model can be cloned bypassing the efforts (e.g., collecting and annotating the dataset). Furthermore, the stolen model using these approaches can also be leveraged to perform subsequent attacks, such as for crafting adversarial examples (Papernot

et al., 2017b) against the black-box model. While these approaches show remarkable performances when stealing simpler models (e.g., shallow MLPs, decision trees) (Tramèr et al., 2016), it was unclear whether equally effective attacks are possible on larger models (e.g., ResNet) without making underlying assumptions of the black-box model.

Contributions of the thesis. In Orekondy et al. (2019b) (Chapter 6), we demonstrate attacks that are effective in stealing complex neural networks. Unlike much of prior work, our approach does not require any knowledge of the model (e.g., model family) and training dataset (e.g., access to seed samples). Furthermore, we leverage feedback during stealing within a reinforcement framework to improve sample efficiency and additionally recover information of the training data.

In spite of recent advances in model stealing strategies since our publication (Jagielski et al., 2020; Krishna et al., 2020; Carlini et al., 2020), work on defending such attacks is limited. In Orekondy et al. (2020b) (Chapter 7) we additionally found existing attacks ineffective in defending against stealing techniques. As a result, we proposed the first defense that can withstand highly accurate model stealing attacks for up to tens of thousands of queries and further amplifying the attackers' error rates.

1.2 OUTLINE

In this section, we outline the content of the thesis.

Chapter 2: Related Work We review literature that sets the foundation for the contributions presented in the thesis. The discussion in this chapter is two-fold. First, we present a broad overview of advances in machine learning, computer vision, and certain aspects of trustworthy ML (e.g., safety, privacy) to help frame the content in the following chapters. Second, we present detailed discussions on two specific problems in trustworthy ML that is closely related to the thesis: privacy-preserving techniques and reverse-engineering ML models.

Part I: Leakage in Visual Data

In this part, we address privacy leakage in visual data. Our broad goal in this part is to work towards a 'Visual Privacy Advisor' to identify and control a wide-spectrum of private information revealed in visual content.

Chapter 3: Understanding and Predicting Privacy Risks in Images In this chapter, we present our work which takes the first step towards understanding and controlling privacy risks in visual content. Towards analyzing privacy leakage, we propose a visual privacy dataset (VISPR), present user studies to understand

the extent of privacy leakage, and develop approaches to identify privacy risks in visual content.

The content of this chapter corresponds to the ICCV 2017 publication *Towards a Visual Privacy Advisor: Understanding and Predicting Privacy Risks in Images* (Orekondy et al., 2017). Tribhuvanesh Orekondy was the lead author of this paper.

Chapter 4: Automatic Redaction of Private Information in Images In this chapter we extend our formulation in Chapter 3 to detecting privacy risks on a pixel-level. To enable pixel-level identification of privacy risks, we extend the VISPR dataset by additionally annotating a subset of the images with pixel- and instance-level annotations. We present approaches to identify and further obfuscate a variety of private information stemming from multiple modalities.

The content of this chapter corresponds to the CVPR 2018 (spotlight) publication *Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images* (Orekondy et al., 2018). Tribhuvanesh Orekondy was the lead author of this paper.

Part II: Leakage during Training

In this part, we move our focus from information in visual data to artifacts produced during training an ML model.

Chapter 5: Understanding and Controlling Deanonymization in Federated Learning In this chapter, we investigate privacy leakage arising from model iterates shared by individuals in a federated learning (McMahan et al., 2017) setting. Specifically, we identify that the de-identified model parameter updates intermittently communicated by individuals also encode user-specific identifiable information. To reduce the leakage, we present utility-preserving approaches which alters the data-distribution of the user.

The content of this chapter corresponds to the work *Gradient-Leaks: Understanding and Controlling Deanonymization in Federated Learning* (Orekondy et al., 2020a), which is currently under submission. A short-version of the work was presented as an oral presentation at the International Workshop on Federated Learning for User Privacy and Data Confidentiality in conjunction with NeurIPS 2019 (FL-NeurIPS'19). Tribhuvanesh Orekondy was the lead author of this paper.

Part III: Leakage during Inference

In this part, we move our focus to leakage of information during black-box inferences on a trained model. We specifically study leakage of *functionality*.

Chapter 6: Stealing Functionality of Black-Box Models In this chapter, we investigate whether the functionality complex CNNs can be 'knocked-off' or stolen by making minimal assumptions. We study sample-efficient strategies to understand whether such an objective is possible.

The content of this chapter corresponds to the CVPR 2019 publication *Knockoff Nets: Stealing Functionality of Black-Box Models* . Tribhuvanesh Orekondy was the lead author of this paper.

Chapter 7: Towards Defenses Against DNN Model Stealing Attacks In this chapter, we propose an approach to defend against model stealing attacks, such as the one proposed in Chapter 6. Investigation in such approaches are crucial given the asymmetry of performances between effective recent attacks and largely ineffective defenses. To tackle the problem by presenting a utility-constrained approach which introduces the first active defense by introducing calibrated noise to degrade the attackers' performance.

The content of this chapter corresponds to the ICLR 2020 publication *Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks*. Tribhuvanesh Orekondy was the lead author of this paper.

Chapter 8: Conclusion In this chapter, we summarize the findings of the thesis. In addition, we discuss future research problems and also present a broader outlook towards the goal of trustworthy machine learning.

2

RELATED WORK

In this chapter, we recap literature in the interdisciplinary topic of machine learning, computer vision, privacy, and security. As recent advances in machine learning and computer vision underpins all methods proposed in this thesis, we first begin with a brief review of related approaches in Section 2.1. In Section 2.2, we discuss trustworthy aspects entailing advances in ML, that enables wide-spread adoption of such advances into real-world applications. Trustworthy aspects typically involve societal challenges e.g., explaining decisions to humans, ensuring the technique is robust to malicious parties. We use Section 2.2 as a foundation for the following two chapters, which in detail addresses two specific trustworthy aspects: privacy of data and robustness against adversaries. In Section 2.3, we review privacy-preserving techniques to minimize leakage of individual-specific private information *before* the information is shared. This section helps frame our contributions in Part I to control leakage of information in personal photos shared on social networks. In the following Section 2.4, we discuss literature that highlights vulnerabilities of the ML system by viewing it from an adversarial lens. We discuss recent advances that exploits leakage to reverse-engineer specific private information of the model and the owner's training data. We build on top of these advances in Part II and Part III and paint a more complete picture on reverse-engineering risks.

2.1 ADVANCES IN MACHINE LEARNING

We begin the chapter by first discussing recent advances in machine learning techniques, especially when applied to solving challenging problems particularly in computer vision. Over the next sections in this chapter, we complement the literature reviewed here with privacy and security aspects.

In Section 2.1.1, we discuss recent advances in deep neural networks to understand visual scenes, such as associating objects in visual scenes to semantic categories. In Part I of this thesis, we leverage and extend these methods to detect private content in images. Alternatively, in Parts II and III, we view the trained models for visual recognition through an adversarial lens.

Training ML models also present challenges in the form of resource-limitations (e.g., data, computation, knowledge). In Section 2.1.2, we take a closer look at literature surrounding learning in such environments. This is particularly relevant in the thesis as we investigate methods for an adversary, who operates in resource (and especially knowledge) limited settings.

2.1.1 *Object Recognition and Scene Understanding*

A fundamental problem in image understanding is *image classification*: $f : \boldsymbol{x} \mapsto y$, where $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$ is an image and $y \in \mathbb{Y}$ is a semantic category (e.g., object type) best describing the image. A large body of work exists which models an image classifier by first obtaining a robust image-level *representation* $\phi(\boldsymbol{x}) \in \mathbb{R}^D$ and classifying such representations $f_{\boldsymbol{w}}(\phi(\boldsymbol{x}))$. Approaches following this framework (Schneiderman and Kanade, 1998; Lowe, 1999; Fergus et al., 2003; Dalal and Triggs, 2005) made considerable preliminary progress towards the goal of image understanding.

A parallel line of work on convolutional neural networks (CNNs), dating back to Fukushima and Miyake (1982) and LeCun et al. (1998), investigated jointly learning representation and classification in an end-to-end manner. Instead of extracting a set of hand-crafted features from images, CNNs instead *learn* spatially-invariant features (via parameterized convolution kernels) in a data-driven manner. Initial approaches which composed a sequence of such convolutions with non-linear activations demonstrated showed remarkable performances (LeCun et al., 1998). CNN-based approaches are currently de facto for image classification as they consistently demonstrate significant improvements (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016a) over their counterparts. The empirical gains of CNN-based approaches is accelerated by availability of larger datasets (Deng et al., 2009; Krasin et al., 2017; Lin et al., 2014), higher compute power, and an improved understanding on specific components of a CNN e.g., activation functions (Krizhevsky and Hinton, 2009), initializations (Glorot and Bengio, 2010), optimizers (Kingma and Ba, 2014) and regularization methods (Srivastava et al., 2014; Ioffe and Szegedy, 2015).

CNN-based approaches have also proven effective on a variety of tasks closely related to image classification: localizing objects via bounding box detections (Girshick, 2015; Ren et al., 2015), pixel-level labelling (Long et al., 2015; Ronneberger et al., 2015; He et al., 2017b). They key idea in these approaches is to use a pretrained CNN – such as a Resnet (He et al., 2016a) in Mask-RCNN (He et al., 2017b) – as a backbone in a network trained to predict location-specific information.

Apart from objects, visual scenes are also composed of text sequences and understanding them forms a crucial part in many applications. For instance, an agent assisting a visually-impaired person spotting a store of a given name on the street. Unlike optical character recognition (OCR) methods typically applied to documents, methods to spot and understand text sequences in visual scenes need to be robust to unconventional deformations of text and complex backgrounds. Towards understanding text sequences in visual scenes, recent approaches have explored detecting texts (He et al., 2017a; Wu and Natarajan, 2017; Neumann and Matas, 2012), recognizing the character sequence that compromises it (Sutton and McCallum, 2006; Shi et al., 2016), and to also perform these steps in an end-to-end manner (Wang et al., 2011; Li et al., 2017a; Luo et al., 2019; Zhan and Lu, 2019).

The approaches presented in this thesis is underpinned by CNN techniques. We leverage CNNs to identify private content leaked on an image-level (Chapter 3). In Chapter 4, we propose an approach based on recent advances in semantic segmentation to localize pixels that reveals private content (such as visual faces and textual names) with an objective of obfuscating corresponding pixels. In Chapter 6 and Chapter 7, we expose standard CNN models, such as VGG-16 (Simonyan and Zisserman, 2014) and ResNets (He et al., 2016a), via a black-box interface to study adversarial attacks.

2.1.2 *Learning with Limited Resources*

Training machine learning models, and especially deep neural networks, is often accompanied by challenges in resource-constrained settings such as compute (e.g., on lower-power devices) or knowledge limitations (e.g., availability of annotated data). In this section, we specifically focus on the latter.

Limited labeled data. A popular type of knowledge limitation is in the flavour of a scarcity of annotated data. In the extreme case to perform image classification, either none (zero-shot) or a handful (few-shot) of images are available for classes that will be encountered at test time. Techniques to overcome these challenges leverage additional knowledge such as domain-specific knowledge (Xian et al., 2016; Sung et al., 2018) (such as from wikipedia) or transferring knowledge from another similar task. Common approaches to transfer knowledge involve re-purposing good initializations from models pre-trained on closely-related annotated datasets (Yosinski et al., 2014; Donahue et al., 2014), or 'self-supervised' tasks (Doersch et al., 2015; Noroozi and Favaro, 2016) not requiring additional annotation. The latter is closely related to semi-supervised learning techniques (Chapelle et al., 2006; Zhu and Goldberg, 2009; Berthelot et al., 2019), for which the underlying scenario is that in addition to a small set of labeled data, a large pool of unlabeled data is also available. In such a scenario, one could also leverage advances in active learning Cohn et al., 1996; Settles and Craven, 2008; Ebert et al., 2012; Beluch et al., 2018 to find best candidate in the unlabeled pool for annotation. Such scenarios also reoccur in adversarial machine learning literature, as we will see in Part III of the thesis. Here, adversaries are typically data-limited and attempt to train models with access to a small set of labeled data (Papernot et al., 2017b; Juuti et al., 2019), or by selecting the best candidates from a large unlabeled pool (Chapter 6).

Limited model knowledge. Challenges in knowledge limitation, apart from data, could also arise from a poor understanding or opaqueness of a complex function that is being optimized. For instance, consider hyperparameter tuning of learning algorithms as an optimization problem. Here, the optimization objective is a complex functional mapping from the hyperparameters (the input) to the validation loss

(the output), and hence introducing a challenge in computing the exact gradient information. In the extreme case, the objective function could also be opaque i.e., a 'black-box' either due to its complexity (e.g., an operating system) or because it is simply unknown. The latter, where the objective function is unknown, is a typical situation in adversarial ML scenarios, as the attacker has limited knowledge on the target model (the objective function). As computing gradients over the objective function is often a requirement to perform certain attacks (e.g., introducing targeted misclassification), recent approaches (Chen et al., 2017; Ilyas et al., 2018) rely on derivative-free methods (Rios and Sahinidis, 2013) to approximately recover gradients over the black-box objective. In this thesis, these forms of partial or no knowledge on the model (the objective function) plays a key role. In Part III, where we study model stealing as a two-player game, each player only has a black-box view of the other player's model.

2.2 TRUSTWORTHY MACHINE LEARNING

Recent advances in ML models has led to remarkable performances on a variety of tasks (Krizhevsky et al., 2012; Vaswani et al., 2017) when evaluated on challenging held-out test sets. However, does this make the models 'trustworthy' enough to be confidently deployed to real-world systems? Answering this question often intersects with societal aspects: what is the explanation behind a model's prediction? Does the model exploit sensitive demographic attributes to make decisions? Could the model leak sensitive information related to training data during deployment? In this section, we discuss some topics that attempt to understand these questions.

Interpretability. A common criticism behind deep neural networks is that unlike humans, the reasoning behind predictions are opaque. This is problematic in situations where decisions are safety-critical e.g., should a patient be given cancer treatment? A popular approach to explain decisions, in the specific case of image classification, is to recover attribution masks (Simonyan et al., 2013; Selvaraju et al., 2017) for predictions. Alternatively, models can be trained to be interpretable-by-design by complementing predictions with natural language explanations (Hendricks et al., 2016). In contrast to explaining decisions, some approaches have also proposed 'debugging' pretrained black-box models using influence functions (Koh and Liang, 2017) and sensitivity analysis (Ribeiro et al., 2016).

Robustness to domain shifts. Design of ML algorithms is largely driven by an iid assumption: examples encountered at test-time arises from the same distribution as training. However, in most scenarios, it is extremely challenging to accurately capture the variations of the true test-distribution. This leads to a covariate (or domain) shift (Bickel et al., 2009) between the training data distribution and real-world (test) distribution. This domain shift can arise due to *sample bias* (Torralba, Efros, et al.,

2011) or through *corruptions* (Yin et al., 2019) during sensing. To tackle sample biases, a long line of work exists to learn domain-invariant models (Saenko et al., 2010; Tzeng et al., 2017). A special case is when the data is biased towards (or against) certain demographics learning to unfair decisions (Dwork et al., 2012; Hardt et al., 2016; Chouldechova, 2017). Domain shifts can also arise out of natural corruptions (e.g., faulty camera sensor) and perturbations in test data (Yin et al., 2019; Hendrycks et al., 2019). Corruptions could also be artificially introduced (Szegedy et al., 2013; Goodfellow et al., 2014a) leading to security concerns; this is discussed in the next paragraph.

Security. Deploying models in untrusted environments exposes them to be potentially interacted with malicious agents. Work around security and ML investigate whether such malicious agents can make the system misbehave and function outside of intended specification. The agent, i.e., the 'adversary', can interact with the system, by providing malicious inputs at inference-time or at training-time. In both cases, the malicious inputs are provided to make the system behave to achieve the adversary's objective (e.g., classifying a spam email as not-spam). At inference-time, a common goal investigated is to achieve (targeted) misclassification (Biggio et al., 2013; Szegedy et al., 2013) on an input by introducing imperceptible perturbations to inputs. The crafted inputs can also be used at training-time by introducing it in to the victim's training set so as to poison (Shafahi et al., 2018; Koh and Liang, 2017) the model. To complement such attacks, many works also focus on how to defend models against such malicious inputs such as by training robust models (Madry et al., 2018). We refer the reader to Papernot et al. (2018a) for a more comprehensive overview.

Privacy and confidentiality. In the previous paragraph, we looked at an adversary interacting with the model to make it misbehave. Now, we consider interacting with the system to infer or 'reverse-engineer' information about the model or its training data. We study this from two perspectives. First, a 'privacy-by-design' view, where the data/models are sanitized by the owner *before* it is shared to benign and malicious parties. Second, an 'adversarial' view, modeling an attacker who tries to infer information *after* it is shared by the owner. We take an in-depth look into both these aspects in Sections 2.3 and 2.4.

2.3 PRIVACY-PRESERVING MACHINE LEARNING

Having motivated the challenges to deploy ML systems in real-world environments, we take a close look at privacy and confidentiality aspects in this and the next section of the chapter. In this section, we consider the setting that the owner needs to share data to potentially untrusted parties for legitimate purposes (e.g., social networking, for collaboratively learning). We discuss techniques that enables such sharing, but

by minimizing leakage of private information related to himself, or of individuals corresponding to the data.

Privacy-enhancing techniques. A range of techniques in modern cryptography exist to enable sharing data from one party to another, such that only these two parties have access to the content. Techniques in secure multi party communication (MPC) (Yao, 1986; Goldreich, 2009) extend this idea to when multiple parties (each with their set of private information) need to compute a function (e.g., mean) jointly over their inputs without revealing individual inputs (e.g., their salaries). These ideas have also been extended to perform aggregation operations during collaborative learning (Bonawitz et al., 2017). An alternative to MPC is studying fully homomorphic encryption (FHE) techniques, such that common mathematical operations can be performed on the underlying data, while the data remains in an encrypted form (Yonetani et al., 2017). Both MPC and FHE enables computing an aggregate function of a dataset, without revealing the individual datapoints.

Privacy on aggregate statistics. However, what if the aggregated data by itself reveals private information? Differential Privacy (Dwork, 2006) provides a theoretical framework to ensure the aggregate data conveys useful population-level statistics, without revealing anything about individual datapoints (e.g., data of a particular person). As a motivating example, when training a classifier on patient records, it is crucial that the classifier is not overly sensitive to presence of a single individual record (a single training example here). In the specific case of training ML models, recent works (Abadi et al., 2016b; Papernot et al., 2018b) have explored variants of SGD to enable training ML models to accompany training with such rigorous guarantees. In such cases, the parameters of the model can be shared confidently knowing that the leakage (as per the definition of DP) is bounded. DP techniques can also be extended to the special case where a version of the sensitive dataset needs to be sanitized, prior to being publicly shared. This is the problem of privacy-preserving data publishing, where differentially private SGD can be used to train a generative model (Harder et al., 2020; Zhang et al., 2018a; Yoon et al., 2019) to serve as a private surrogate to the original data distribution.

Record-level privacy. DP techniques help ensure privacy of individual records (e.g., patient records) when publicly-sharing aggregated population-level statistics (e.g., mortality rates). However, many situations arise when an individual record itself needs to be released. For instance, when the owner wants to share a tweet, or a photograph on social media. In such cases, various non-explicit cues can encode a person's identifiable information such as clothing in images (Oh et al., 2016), writing style in sentences (Shetty et al., 2018b), and statistical correlations in movie preferences (Narayanan and Shmatikov, 2008). In the next section, we continue this discussion specific to mitigating private information encoded in visual data.

2.3.1 *Visual Privacy*

In this section we consider the problem of identifying and controlling private information in images. We begin by enumerating some of the challenges to tackle this problem. First, there is no established notion of *what* attributes in visual content encode privacy-related information. For instance, as highlighted in Part I, images can depict a wide range of private attributes of an individual e.g., political viewpoint, sexual orientation, relationships. Second, even if these attributes are well-defined, *automatically identifying* such attributes is a longstanding open-problem in image understanding. Third, post-identification of relevant private attributes, it is unclear *how to mitigate* leakage of this information which demonstrate local structural correlations, as traditional strategies (e.g., adding random noise) are insufficient to hide such attributes. Now, we discuss how existing research tackles these challenges.

Privacy goals. Our overarching research goal in Part I is studying techniques to enable users manage their privacy expectations when sharing personal photos on social media. We largely view privacy in the contextual integrity (CI) framework by Nissenbaum (2009). The CI framework considers privacy in terms of the appropriate information flow between a sender and a recipient within a specific context. A privacy violation occurs when the information flow between the parties deviates from the expected norm. Consider an example in our case where we study sharing personal photos on social networks: Alice (the subject) wishes to share a photograph of her new car to her friends (the recipients), but the additional visibility of the car's registration details in the photograph leaks unintended identifiable information and thereby deviates from Alice's privacy expectation. Consequently, our goal in Part I is to work towards techniques that identifies visual content not adhering to the expected privacy requirement of the individual. The content we study spans visual cues that encode identifiable information (e.g., facial features, car registration number) and additionally cues to manage impression (e.g., political opinions, hobbies) (Goffman et al., 1978). We refer the reader to Li (2020) for a more comprehensive discussion on the interplay between behavioral theories of privacy and photo sharing on social networks.

Visual cues encoding private information. There are a variety of cues in visual data that encodes personal information of individuals. The predominant cue addressed in literature is with respect to person's *identifiable* information, revealed by faces (Huang et al., 2007; Parkhi et al., 2015), and body (Gallagher and Chen, 2008; Oh et al., 2015). Such identifiable cues are sometimes a useful feature, such as for a person tagging faces to manage personal photo collections (Stone et al., 2008; Zhang et al., 2015) or to estimate demographics of strangers (Gallagher and Chen, 2009). However, such cues also raise privacy concerns (Besmer and Richter Lipford, 2010) when used by certain parties (e.g., law enforcement, advertisers) to investigate a per-

son's behaviour. Apart from body-specific identifiable information in images, there exists scant research in studying other visual cues that depict person's behavioural or private information. Some complementary visual cues that has received recent attention include display screens (Raguram et al., 2011; Xu et al., 2013; Korayem et al., 2016), relationships (Wang et al., 2010; Sun et al., 2017b), occupation (Shao et al., 2013), and visited locations (Li et al., 2009). In Part I, we present a significantly broader range of privacy-sensitive visual cues (which we refer to as the 'privacy attributes') that reveals personal information of an individual. Our proposed privacy attributes quantify visual privacy leakage using body-specific cues (e.g., facial features), other identifiable information (e.g., passport details), behaviours (e.g., hobbies, political opinion), and other sources of private information (e.g., disabilities) depicted in images. Some recent works (Li et al., 2018; Li et al., 2020) have also studied complementary attributes that additionally harm an individuals' impression management (e.g., embarrassing shots). Since our work, there is a growing interest (Vishwamitra et al., 2017; Li et al., 2020) towards recognizing privacy leakage from a broader perspective for images captured using mobile (Gurari et al., 2019), eye-tracking (Steil et al., 2019), and home assistant (Wu et al., 2018b) devices.

Understanding user privacy requirements. Identifying privacy attributes provides a reasonable framework to understand *what* content could be considered sensitive to an individual. However, *translating* the presence of such content to privacy risk depends on the context and the individual. In the case where the context is sharing personal photographs on social networks, recent works (Wisniewski et al., 2017; Knijnenburg, 2017; Li et al., 2018) (including our own in Chapter 3) suggest that individuals have diverse privacy requirements over different types of content appearing in their shared photographs. In particular, Wisniewski et al. (2017) demonstrate six patterns emerge (e.g., privacy maximizers, selective sharers, privacy minimalists) among the individuals' privacy requirements. In Chapter 3, we similarly find such patterns emerge in our user study conducted over our privacy attributes. Subsequently, our goal in Part I is to work towards estimating user-tailored privacy leakages by factoring in diverse privacy requirements.

Estimating privacy risks in images. In parallel to works identifying privacy-sensitive visual cues from images, works also estimate the privacy risk posed by an image. Tonge and Caragea (2015) were motivated to detect privacy violation in images prior to sharing on social media. Their approach classifies whether an image is public or private based on features extracted from a Convolutional Neural Network and user-generated tags for the image. However, we show in Chapter 3 that users have different notions of privacy and hence it cannot be modeled as a binary classification problem. Xioufis et al. (2016), similar to our work, factor in distinct user perceptions towards different privacy attributes. Unlike our approach, their model however requires user-specific image-level annotations to understand privacy

preferences. In contrast to these works, our work in Chapter 3 first tackles a more principled problem of predicting the privacy-sensitive elements present in images and use these in combination with users preferences to estimate privacy risk.

Controlling privacy leakage via obfuscation. After identifying sources of privacy leakage in visual content, a complementary set of techniques are required to mitigate the leakage of this information. The predominant mitigation strategy is via obfuscation: the visual input is manipulated such that it reduces an adversary's effectiveness in recovering the underlying private attributes (e.g., identity). There are multiple aspects to obfuscating information in images:

(i) against *whom* is the information obfuscated? For the case when the adversary is a deep neural network trained to infer certain attributes (e.g., identity, gender), a recent line of work (Oh et al., 2017; Wu et al., 2018a) proposes to leverage advances in adversarial learning to reduce attribute classification accuracies of the neural network. However, as these methods introduce bounded (and typically imperceptible) perturbations to the inputs, it is unclear whether they mitigate leakage against a stronger class of adversaries e.g., humans. Consequently, studies have in parallel addressed obfuscating privacy-sensitive regions such as by redacting (Chapter 4), encrypting (Boult, 2005; Chattopadhyay and Boult, 2007) or re-synthesizing localized regions (Sun et al., 2018a; Ma et al., 2018) in images to mitigate leakage;

(ii) *which* regions need to be obfuscated? Literature predominantly addresses obfuscating *facial* features (Bitouk et al., 2008; Wilber et al., 2016; Sun et al., 2018a). However, as Oh et al. (2016) demonstrate, a variety of other complementary identifiable cues (e.g., body) can nonetheless lead to leaking privacy of the obfuscated persons. Consequently, in Chapter 4, we extend obfuscation to a much broader set of localizable regions (e.g., fingerprints, location names) appearing in images whose cues from multiple modalities.

(iii) *how* to obfuscate the target regions? The de facto notion for obfuscation is destroying information content in the targeted region e.g., by blacking-out, blurring; see Wilber et al. (2016) and Hasan et al. (2018) for a more comprehensive study. However, these strategies can lead to introducing unnatural artifacts and reducing the image's utility, which is an important criteria for individuals sharing social photos. Consequently, recent studies propose content-preserving obfuscations such as by cartooning (Hassan et al., 2017) or re-synthesizing (Brkic et al., 2017; Sun et al., 2018b) the targeted regions with realistic substitutes by taking advantage of generative methods.

(iv) what are the privacy *implications* of imperfect obfuscations? Techniques proposed in literature typically fall short in *perfectly* identifying and obfuscating

privacy-sensitive regions due to a number of challenges, such as large variations of pose and illuminations of targeted content (e.g., faces). How do such imperfections in identifying and obfuscating content impact privacy? To help us answer the question, consider an example where a user, prior to sharing a set of photos, uses an algorithm to obfuscate her face to minimize the number of photos that reveals her identity. In this case, whether the algorithm is able to successfully enforce privacy of the user is determined by its effectiveness (e.g., identity misclassification accuracy rates) and more importantly, if this effectiveness metric at test-time surpasses the users' privacy requirements. Consequently, a major focus in literature (Wilber et al., 2016; Sun et al., 2018b) is advancing techniques to improve the effectiveness to accommodate stringent privacy requirements. In parallel, literature (Oh et al., 2016; Zhang et al., 2020) has also investigated techniques that studies a contrasting objective: to understand adversarial capabilities that degrades the effectiveness, such as by leveraging prior information to reconstruct obfuscated regions.

Some works (Jana et al., 2013; Templeman et al., 2014; Fernandes et al., 2016) simultaneously address all the above aspects by developing a system capable of controlling privacy for a specific domain. In particular, Jana et al. (2013) propose a privacy-preserving platform-level perceptual library where untrusted applications access data from visual sensors only via the proposed API; the API returns only the necessary information to applications by performing privacy-preserving transforms (e.g., sketching) on the raw visual data.

Privacy, utility, and user behaviors. In addition to methods to automatically identify and obfuscate content, effectively enforcing privacy constraints also requires a behavioral understanding of users. This is motivated by a large body of work (Barnes, 2006; Norberg et al., 2007) demonstrating a privacy paradox on social networks: users' information dissemination behavior does not reflect their intended privacy requirements. Similar paradoxical findings are also observed (Amon et al., 2020) when sharing visual data; we discuss similar findings in Chapter 3. By understanding user behavioral factors, studies present privacy-enhancing mechanisms to better control intended dissemination such as by designing effective users interfaces (Besmer and Richter Lipford, 2010), predicting content-specific privacy requirements (Liu et al., 2011), or by factoring-in personalized privacy requirements (Ahern et al., 2007; Hoyle et al., 2020). In parallel to studying user attitudes towards privacy, literature also addresses human perception towards manipulation strategies (e.g., blurring) traditionally used by image obfuscation techniques. While we briefly study the influence of redactions mask sizes in Chapter 4, our findings are complemented by many recent studies (Li et al., 2017d; Hasan et al., 2018; Hasan et al., 2019) that additionally analyze the influence of many other factors e.g., obfuscation filter strategy, location of the scene, type of object.

Automatic privacy advisor. The goal in Part I is to work towards a privacy advisor that can assist users in enforcing their privacy requirements when sharing data. Our work in Part I, along with recent works (Tonge and Caragea, 2019; Xioufis et al., 2016; Vishwamitra et al., 2017; Li et al., 2020), tackles the case where the data corresponds to personal photographs shared on social networks. More broadly, literature has proposed automated privacy advisors to assist users in a range of situations outside of visual privacy such as to auto-configure privacy permissions on mobile apps (Liu et al., 2016) and IoT devices (Das et al., 2018), predicting sharing policies for text-content Sinha et al. (2013), and sanitizing hashtags to obfuscate location information (Zhang et al., 2018b).

Datasets for visual privacy. Datasets to aid recognition of private information in images has evolved over time. Initial works, e.g., AT&T database (Samaria and Harter, 1994), Yale face database (Belhumeur et al., 1997), captured front faces of persons under lab conditions with restricted poses and constant illumination. Follow-up datasets for identity recognition, e.g., LFW (Huang et al., 2007), CelebA (Liu et al., 2015), PEViD (Korshunov and Ebrahimi, 2013), present additional recognition challenges as faces captured display varying poses, illumination, and occlusions. More recently, there is a push towards studying identity recognition beyond frontal face images. For instance, the PIPA (Zhang et al., 2015) dataset builds an image dataset capturing multiple individuals, often in social settings; these images are more representative of content shared on social networks. However, as we motivated earlier in the section, person identities capture only a narrow notion of private information contained in images. In parallel, PicAlert (Zerr et al., 2012) and YourAlert (Xioufis et al., 2016) contain binary labels of whether images are considered private by individuals. However, as they lack ground-truth annotations over visual cues that makes them privacy-sensitive, the reasoning behind privacy annotations is unclear. Consequently, our datasets proposed in Part I, apart from representing persons and crowds with large variations in poses and backgrounds, also contains: (a) images and annotations capturing a significantly broader range of privacy attributes (e.g., presence of name-tag), annotated at image- and pixel-level; (b) user-studies to capture privacy preferences over the attributes; (c) non-person centric images capturing private information (e.g., close-up photograph of an airline boarding pass); and (d) non-private images (e.g., generic image of a cat).

2.4 REVERSE-ENGINEERING MACHINE LEARNING MODELS

In the previous section, we discussed approaches that provides the owner control over the data *before* it is shared to untrusted parties. A concurrent research direction is to view the shared data by taking the role of a adversary to expose vulnerabilities in the system. The adversary observes outputs (e.g., posterior predictions, model parameters) *after* it is shared by the owner. By observing these outputs, the end-goal

of the adversary is to *reverse-engineer* private and confidential information about the owner's ML model or its training data.

2.4.1 *Inferring Training Data Attributes*

We being by looking at studies that highlight the adversarial attack techniques to infer certain properties of the training data. Central to the modeling such attacks is laying out: (a) the attack surface: what is the information observed by the adversary?; and (b) the attack objective: what private information of the training dataset does the attacker wish to recover? In the following paragraphs, we cover certain attack surfaces and objectives investigated in literature.

Attack surface. Specific to achieving attack objectives by interacting with ML models, attack surfaces in literature commonly fall in two extreme ends: (i) a 'white-box' setting, where the adversary observes the internals (e.g., parameters, hyperparameters) of the owner's model; and (ii) a 'black-box' setting with opaque internals, but allowing an API access (inputs in, predictions out) to the adversary. Recent studies have explored complementary attack surfaces such as intermediate features (Song and Shmatikov, 2020), gradient updates (Melis et al., 2019; Nasr et al., 2019), and difference in outputs (Salem et al., 2020). In this thesis, to highlight privacy risks, we use the gradient update information (Part II) and black-box access (Part III) as the attack surfaces.

Inference objectives. Literature indicates that a variety of private properties related to the training data can be inferred by interacting an ML model. Such inferences are especially problematic when the training data is sensitive e.g., health records of patients in a hospital. A basic privacy violation is inferring membership properties i.e., to determine whether a certain datapoint is part of the training data via black-box interactions with the target model. Studies (Shokri et al., 2017; Long et al., 2017; Salem et al., 2019; Jayaraman and Evans, 2019) demonstrate effectiveness of membership inferences by leveraging the insight that the target model returns overconfident predictions when overfit to its training data. Along these lines, attribute inferences (Ateniese et al., 2015; Fredrikson et al., 2015) additionally demonstrate that properties of training subsets can also be inferred by exploiting black-boxes. This is closely related to our work in Chapter 5, wherein an adversary uses gradient update information as an attack surface to infer the identity of participating individuals. While these works infer certain discrete properties related to the private training dataset, there is a recent push towards 'inverting' (Fredrikson et al., 2015) the model i.e., recovering prototypical training examples. Towards the goal of reconstructing training examples, Zhang et al. (2020) recently proposed to leverage class-specific loss signals from a target white-box classifier to guide a GAN to synthesize training inputs.

Defenses. Being able to infer certain properties of the training dataset by interacting with the model is predominantly an artifact of a generalization gap (Shokri et al., 2017) or biases in training data (Song and Shmatikov, 2020). Consequently, studies (Shokri et al., 2017; Salem et al., 2019) indicate employing conventional regularization techniques takes a step towards mitigating leakage of private properties. An alternate strategy involves explicitly bounding the influence of training instances e.g., using Differentially Private training (Abadi et al., 2016b; Papernot et al., 2017a), towards the learnt parameters. In our work in Chapter 5, instead of lowering the generalization gap along these lines, we introduce an adversarial bias in the data by adding decoy training instances to the training set, so as to mislead inferences of private properties.

2.4.2 *Recovering Model Information*

In this section, we switch focus from attack objectives that extract information of the training data to that of model internals. The attack surfaces to achieve extraction objectives are investigated with black-box access to the owner's ML model i.e., inputs in, predictions out. Exposing models via a black-box interface is especially common among 'Machine Learning as a service' (MLaaS) platforms which monetize prediction APIs to a trained model. Here, the trained model is a result of significant financial and human effort (e.g., annotating data, engineering hyperparameters). Consequently, any information of the model (e.g., parameters) reconstructed using extraction attacks threatens the intellectual property of the model's owner. The reconstructed knowledge can be further used to compromise the integrity of the model e.g., by improving effectiveness of adversarial perturbations (Papernot et al., 2017b; Oh et al., 2018).

Model extraction: Attack objectives. Literature investigates how black-box models leak information by studying attack objectives to extract various confidential aspects of model. One aspect are the *hyperparameters* (Wang and Gong, 2018), or specifics of the model architecture (Yan et al., 2018; Oh et al., 2018). In particular, Oh et al. (2018) show that a meta-classifier can be trained to predict certain model attributes (e.g., presence of a max-pool layer) from the output posterior probabilities of a black-box model. A complementary aspect is extracting the *fidelity* (Jagielski et al., 2020) of the target black-box model. Here, the attack objective is to obtain an 'extracted' model that mimics the prediction of the target model on *any* input. Tramèr et al. (2016) successfully demonstrate high-fidelity extraction of linear models, simple MLPs, and decision trees. In the specific case of linear models, their approach treats stealing as an equation-solving problem over unknown parameter variables. More recently, Jagielski et al. (2020) demonstrate successful high-fidelity extraction on two-layer ReLU-based neural networks. The key insight here is that the network is a piecewise-linear function and by strategically searching for inputs that cause a ReLU unit to change signs, one can exactly recover the parameters of the target network.

While these approaches have shown remarkable success in extracting parameters and decision boundaries of linear and simple neural-networks, due to their query complexity, it remains an open question whether complex neural networks can be *exactly* extracted. In Part III of the thesis, we argue for approximately extracting the target model by proposing functionality stealing attacks. Here, the attack objective is to obtain a 'knock-off' model to mimic the target model on test inputs, as opposed to mimicking all inputs in high-fidelity extraction. We argue that in many settings (e.g., MLaaS prediction APIs), an attacker achieving high performance on a test distribution threatens the business model of the owner. To highlight functionality stealing risks, in Chapter 6 we demonstrate a learning-based strategy where the attacker learns a knock-off (student) model to replicate predictions of a target (teacher) model. Remarkably, we find the strategy effective in spite of incomplete knowledge on the training distribution and the specifics of the model internals. In addition, we demonstrate that model stealing attacks (Lowd and Meek, 2005a; Tramèr et al., 2016; Correia-Silva et al., 2018) also extend to complex CNN architectures. Recently, (Krishna et al., 2020; Wallace et al., 2020) show strategies, in the spirit of our work in Part III, are effective in stealing language models.

Model extraction: Defenses. In light of the above attack studies, it is becoming increasingly evident that model extraction techniques pose a risk. However, work on defending against such extraction strategies is minimal. Existing defense strategies (Tramèr et al., 2016; Orekondy et al., 2019b; Lee et al., 2018) employ an information truncation approach such as by rounding-off predictions (i.e., the posterior probabilities over k classes) or revealing predictions over top-k classes. In particular, Lee et al. (2018) introduce ambiguities at the tail-end of the predicted posterior distribution to mitigate attacks. However, as we show in Part III, such defense strategies take a passive role against the attacker and are largely ineffective. Consequently, in Chapter 7, we work towards the first effective defense that actively attempts to attack the attacker. The key idea is introducing utility-constrained perturbations to predictions with an objective to poison the attacker's gradient signals when training the knock-off model. Such a strategy has also shown to be effective when mitigating attacks on language models (Wallace et al., 2020). Since our work, Kariyappa and Qureshi (2020) extended the line of defenses by perturbing output predictions in proportion to distance of the input to the expected test distribution.

Part I

LEAKAGE IN VISUAL DATA

The first part of the thesis investigates leakage of private information in visual data. In particular, *personal photos* representative of images disseminated on social media. Given the large amounts of such data that is disseminated on a daily basis (e.g., via smartphones), it is crucial to understand and control the extent of private information revealed by such images.

In Chapter 3, we aim to understand privacy risks at an image-level. Towards this goal, we present the first taxonomy of visual privacy attributes over a broad range of categories. We also perform real-world user studies to analyze the user perception of privacy risks over the attributes. Equipped with this information, we propose a 'Visual Privacy Advisor' to estimate privacy leakage from images.

In Chapter 4, we extend our previous approach from understanding risks on an image-level to pixel-level. Specifically, we leverage recent advances in segmentation-based approaches to obfuscate privacy-sensitive information while maintaining utility of the image.

3

TOWARDS A VISUAL PRIVACY ADVISOR

In this chapter, we take a step towards understanding and controlling private information disclosed in visual content. This is particularly important as massive amounts of personal visual data is captured (e.g., on smartphones) and disseminated on the internet (e.g., via social networks) and thereby raising major privacy concerns.

There exists a number of solutions to control disclosure of structured *explicit content* (e.g., personal details, GPS location). Most notably, devices (e.g., smartphones) and internet services (e.g., facebook) offer individuals to set privacy settings to control the disclosure of private information. In this chapter, we envision extending the concept of privacy settings to *image content* in the spirit of a *Visual Privacy Advisor*. Towards this goal, we first categorize personal information in images into 68 image attributes and collect a dataset, which allows us to train models that predict such information directly from images. We use the dataset to run a user study to understand the privacy preferences of different users w.r.t. the privacy attributes. Finally, we propose models that predict user specific privacy score from images in order to enforce the users' privacy preferences. Our model is trained to predict the user specific privacy risk and even outperforms the judgment of the users, who often fail to follow their own privacy preferences on image data.

The content of this chapter is based on Orekondy et al. (2017). As a first author, Tribhuvanesh Orekondy conducted all the experiments and was the main writer for the conference paper.

3.1 INTRODUCTION

As more people obtain access to the internet, a large amount of personal information becomes accessible to e.g. other users, web service providers and advertisers. To counter these problems, more and more devices (e.g. mobile phone) and web services (e.g. facebook) are equipped with mechanisms where the user can specify privacy settings to comply with his/her personal privacy preference.

While this has proven useful for explicit and textual information, we ask how this concept can generalize to visual content. While users can be asked (as we also do in our study) to specify how comfortable they are releasing a certain type of image content, the actual presence of such content is implicit in the image and not readily available for a privacy preference enforcing mechanism nor the user. In fact, as our study shows, people frequently misjudge the privacy relevant information content

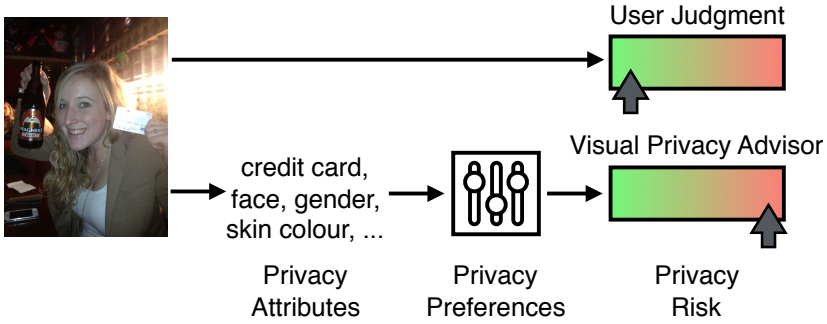


Figure 3.1: Users often fail to enforce their privacy preferences when sharing images online. We propose a first *Visual Privacy Advisor* to provide user-specific privacy feedback.

in an image. The misjudgement leads to the failure of enforcing their own privacy preferences.

Hence, we work towards a *Visual Privacy Advisor* (Figure 3.1) that helps users enforce their privacy preferences and prevents leakage of private information. We approach this complex problem by first making personal information explicit by categorizing personal information into 68 image attributes. Based on such attribute predictions and user privacy preferences, we infer a privacy score that can be used to prevent unintentional sharing of information. Our model is trained to predict the user specific privacy risk and interestingly, it outperforms human judgment on the same images.

Our main contributions in this chapter are as follows: (i) To the best of our knowledge, we are the first to formulate the problem of identifying a diverse set of personal information in images and personalizing predictions to users based on their privacy preferences; (ii) We provide a sizable dataset of 22k images annotated with 68 privacy attributes; (iii) We conduct a user study and analyze the diversity of users' privacy preferences as well as the level to which they achieve to follow their privacy preferences on image data; (iv) We propose the first model for Privacy Attribute Prediction. We also extend it to directly estimate user-specific privacy risks; and (v) Finally, we show that our models outperform users in following their own privacy preferences on images

3.2 THE VISUAL PRIVACY (VISPR) DATASET

Mobile devices and social media platforms provide privacy settings, so that users can communicate their privacy preferences on the disclosure of different type of textual information. How does this concept transfer to image data? We need to establish a similar concept of privacy relevant information types – but now for *images*. This will allow us to query users about their privacy preferences on the disclosure of various information types, as we will do in the next section.

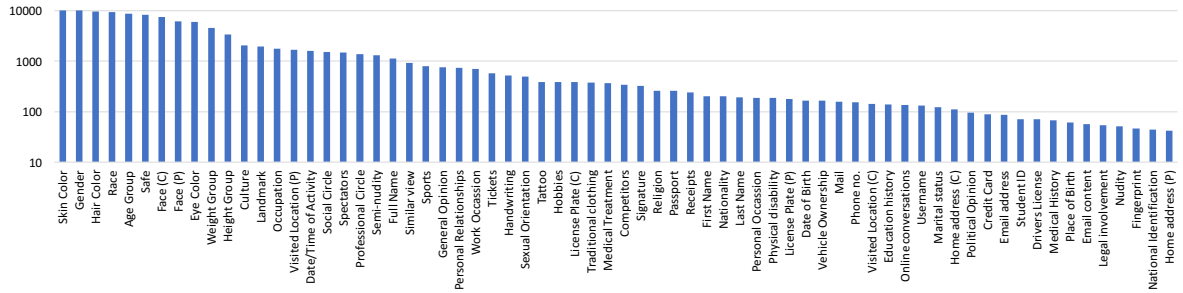


Figure 3.2: Label distribution in the VISPR dataset. Y-axis indicates the number of images.

Therefore, we propose in this section a categorization of personal information into 68 privacy attributes such as gender, tattoo, email address or fingerprint. We collect a dataset of 22k images that allows the study of privacy relevant attributes in images and the training of automatic recognizers.

3.2.1 *Privacy Attributes*

As motivated before, we need to categorize different types of personal content in images – akin to the privacy settings deployed in today's devices and services. Therefore, we define a list of *privacy attributes* an image can disclose.

The primary challenge here is the lack of a standard list of privacy attributes. We thus compile attributes from multiple sources. First, we consolidate relevant attributes from the guidelines for handling *Personally Identifiable Information* (McCallister, 2010) provided in the EU Data Protection Directive 95/46/EC (Directive, 1995) and the US Privacy Act of 1974. Second, we add relevant attributes from the rules on prohibiting sharing personal information on various social networking websites (e.g., Twitter, Reddit, Flickr). Finally, we manually examine images that are shared on these websites and identify additional attributes. As a result, we draft an initial set of 104 potential privacy attributes. As discussed in the next section, these are reduced to 68 attributes (see Table 3.1) after pruning.

3.2.2 *Annotation Setup*

The annotation was set up as a multi-label task to three annotators annotating independent sets of images. A web-based tool was provided to select multiple options corresponding to the 104 privacy attributes per image. Additionally, annotators could mark if they were unsure about their annotation. In case none of the provided privacy labels applied, they were instructed to label the image as *safe*, which we use as one of our privacy attributes. Images were discarded if annotators were unsure, or if the image contained a copyright watermark, was a historic photograph, contained primarily non-English text, or was of poor quality.

3.2.3 *Data Collection and Annotation Procedure*

In this section, we discuss the steps taken to obtain the final set of 22k images annotated with 68 privacy attributes.

Seed sample. We first gather 100k random images from the OpenImages dataset (Krasin et al., 2017), a collection of ∼9 million Flickr images. Using the definition and examples of the privacy attributes, the annotators annotate 10,000 images randomly selected from the downloaded images.

Handling imbalance. Based on the label statistics from these 10,000 images, we add images to balance attributes with fewer than 100 occurrences. These additional images are added by querying relevant OpenImages labels possibly representative of insufficient privacy attributes.

Extended search for rare classes. In spite of using the above strategy, 37 attributes contain under 40 images. We manually add images for these attributes by querying relevant keywords on Flickr. We do not add multiple images from the same album. For credit cards, we manually obtain 50 high-quality images from Twitter, which are the only non-Flickr images in our dataset.

Selected attributes. After annotating the dataset with the initial 104 labels, we discard 19 labels because either (i) images were difficult to obtain manually (e.g. iris/retinal scan, insurance details) or (ii) the set of images did not clearly represent the attribute. We additionally merge groups of attributes which capture similar concepts (e.g. work and home phone number). In the end, we obtain a dataset of 22,167 images, each annotated with one or more of 68 privacy attributes.

Curation. To reduce labeling mistakes, we organize the dataset into batches of images with each batch corresponding to a privacy attribute. We curate attribute batches which either contain fewer than 500 images or are considered sensitive by users.

Splits. We perform a random 45-20-35 split with 10,000 training, 4,167 validation and 8,000 test images. The final statistics of our dataset is presented in Table 3.1. The labels and its distribution in our dataset is shown in Figure 3.2.

3.3 UNDERSTANDING PRIVACY RISKS

In this section, we explore how users' personal privacy preferences relate to the attributes in Section 3.3.1. Furthermore, we study how good users are at enforcing

Split	All	Train	Val	Test
Images	22,167	10,000	4,167	8,000
Labels	115,742	51,799	22,026	41,917
Avg Labels/Image	5.22	5.18	5.29	5.24
Max Images/Label	10,460	4,710	1,969	3,781
Min Images/Label	44	20	7	12

Table 3.1: VISPR dataset statistics

their own privacy preferences on visual data when making judgments based on image data in Section 3.3.2.

3.3.1 *Understanding Users' Privacy Preferences*

In this section, we study the degree to which various users are sensitive to the privacy attributes discussed in Section 3.2.

User study. We present each user with a series of 72 questions in a randomized order. Each of these questions corresponds to either exactly one of 67 privacy attributes (excluding the safe attribute) or a control question. In each question, the users are asked how much they would find their privacy violated if they accidentally shared details of a particular attribute publicly online. For instance: "How much would you find your privacy violated if you accidentally shared details on personal occasions you have attended (like a birthday party or friend's wedding)." Responses for the question are collected on a scale of 1 to 5, where: (1) Privacy is not violated (2) Privacy is slightly violated (3) Privacy is somewhat violated (4) Privacy is violated (5) Privacy is extremely violated. We treat these responses as users privacy preference for this particular privacy attribute.

Participants. We collect responses of 305 unique AMT workers in this survey. Out of the 305 respondents, 59% were male, 78% were under 40 years of age with 57% from USA and 38% from India. Additionally, 75% were regular Facebook users, 80% and 44% reported to be aware of and have used Twitter and Flickr at least once.

Analysis. In order to understand the diversity in users' privacy preferences, we first cluster the users based on their preferences into *user privacy profiles*. We cluster using K-means and choose K based on silhouette score (Rousseeuw, 1987), which considers distance between points within the cluster and additionally distance between points and their neighbouring cluster. We choose $K = 30$ as this yields the lowest silhouette score. This enables visualizing the preferences over the attributes, as seen in Figure 3.3, where each row represents the preferences for one of the 30

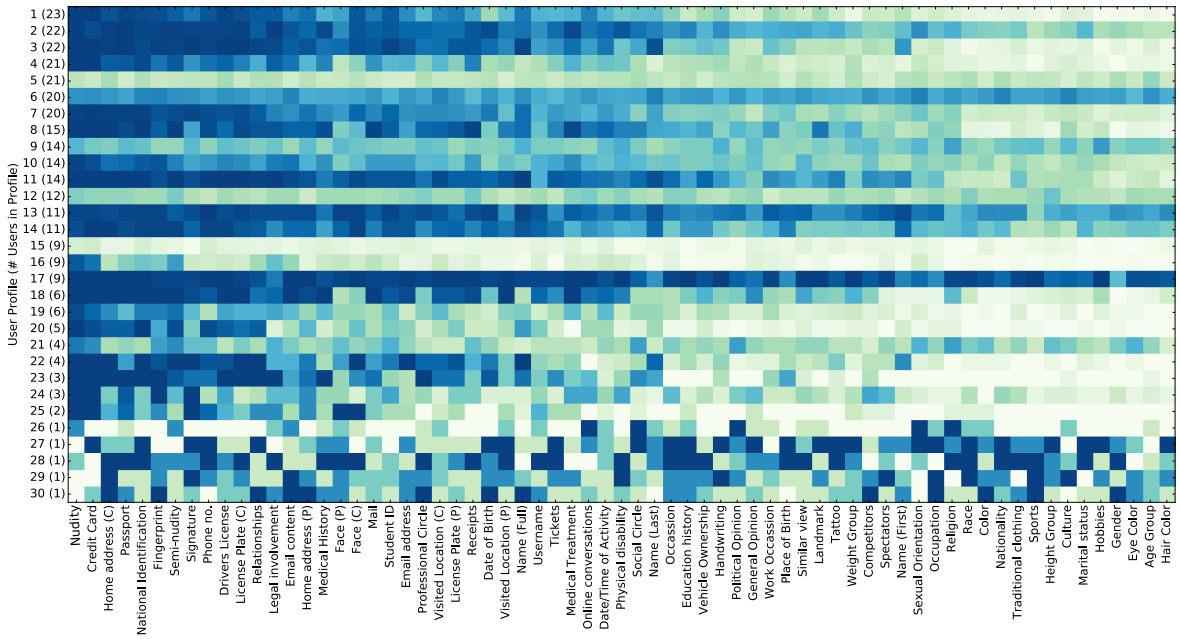


Figure 3.3: Privacy preferences of user profiles for the privacy attributes. Darker colors represent higher privacy-sensitivity to attributes. Each row corresponds to one of the 30 profiles and the number in brackets on the Y-axis represents the number of users mapped to the profile. Rows are ordered based on number of users linked to the profile.

user profiles (ordered based on number of users associated with the profile). We observe from this study: (i) Users show a wide variety of preferences. This supports requiring user-specific privacy risk predictions; (ii) The *majority* (Profiles 1-4, 7-11, 13-14, 18-20 in Figure 3.3) display a similar order of sensitivity to the attributes; (iii) A *minority* (Profiles 21-30) of users are particularly sensitive to some attributes such as their political view, sexual orientation or religion; and (iv) The *uniformly-sensitive users* (Profiles 5, 6, 12, 15, 17) are uniformly sensitive to all attributes even though to different degrees.

3.3.2 *Users and Visual Privacy Judgment*

In this study, we first ask participants to judge their personal privacy risk based on images representing an attribute (providing a visual privacy risk score) and afterwards asking the actual user's privacy preferences for the same attribute (providing a desired or explicit privacy risk score). Hence, we study how good users are at assessing their personal privacy risks based on images.

User study. In this study, we split the survey into two parts. In the first part, the users are shown a group of 3-6 images. Given the sensitive nature of attributes, we cannot obtain or ask users to rate their personal images and hence use images from the dataset. They are asked how comfortable they are sharing such images publicly, considering they are the subject in these images. Responses are collected on a scale of 1 to 5, where: (1) Extremely comfortable (2) Slightly comfortable (3) Somewhat comfortable (4) Not comfortable (5) Extremely uncomfortable. Each group of images represents one of the 68 privacy attributes. In most cases, the attributes occur isolated and are the most prominent visual cue in the image. We refer to these responses as *human visual privacy score*. The second part is identical to questions and the setting in the previous user-study on privacy preferences. Each question is designed to obtain the privacy preference of the user for each attribute. As before, the user rates on a scale of 1 (Not Violated) to 5 (Extremely Violated). We refer to these responses as *privacy preference score*.

Participants. We split the study into two parts to prevent user fatigue. Each part contains only half of the attributes. We obtain 50 unique responses for this survey from AMT. In each of these parts, roughly: 70% of the respondents were under 40 years, 57% were male and 87% were from USA. Additionally, 80% responded that they use Facebook, 84% Twitter and 46% Flickr.

Analysis. We compute for each attribute average privacy preference score and human visual scores, and visualized them as a scatter plot in Figure 3.4. From the results, we observe: (i) The off-diagonal data points show a clear inconsistency in the users between the required privacy preference and their judgment of privacy risk in images; (ii) For cases close to the diagonal, like credit cards, passport and national identification documents, users display consistent behaviour on images and attributes; (iii) However, when photographs are natural scenes containing people or vehicles, users underestimate (below diagonal) the privacy score, such as in the case of family photographs or cars displaying license plate numbers. We speculate this is indicative of personal photographs commonly shared online; and (iv) They overestimate (above diagonal) the privacy risk of some photographs showing birth place or their name. We speculate this is because the photographs are often official documents, making users more cautious.

3.4 PREDICTING PRIVACY RISKS

In this section, we make a step towards our overall goal of a *Visual Privacy Advisor*. As illustrated in Figure 3.5, we follow a similar paradigm e.g. on social networks that defines privacy risk based on both the content type and user-specific privacy settings. In our case, the content type is described by (user-independent) attributes

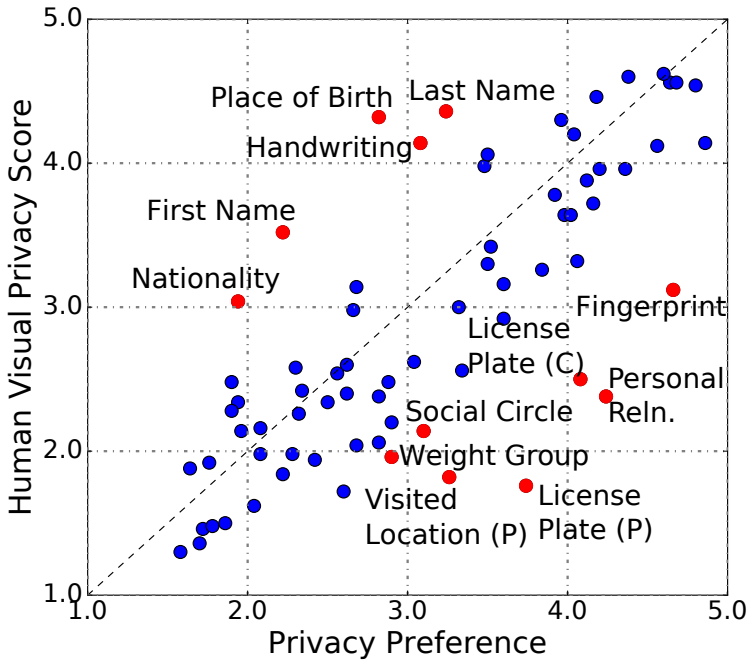


Figure 3.4: Users are asked to rate on a scale of 1 (Not violated) to 5 (Extremely violated) how much an attribute affects their privacy. *X*-axis denotes their desired privacy preference and *Y*-axis denotes their evaluation of risk on images. The red markers indicate privacy attributes with highly underestimated or overestimated user ratings

in the previous section. We combine these with the user-specific privacy preferences to determine if the image contains a privacy violation.

We describe our model for privacy attribute prediction in Section 3.4.1, followed by our approaches to personalized privacy risk prediction in Section 3.4.2. We conclude with a comparison of human judgment of privacy risks in images against the prediction of our proposed models in Section 3.4.3.

3.4.1 *Privacy Attribute Prediction*

In this section, we define the *user-independent* task of predicting privacy attributes from images. Then, we present and evaluate different methods on our new VISPR dataset.

Task. We propose the task of *Privacy Attribute Prediction*, which is to predict one or more of 68 privacy attributes based on an image. This can be seen as a multilabel classification problem that recognizes different type of personal information visual

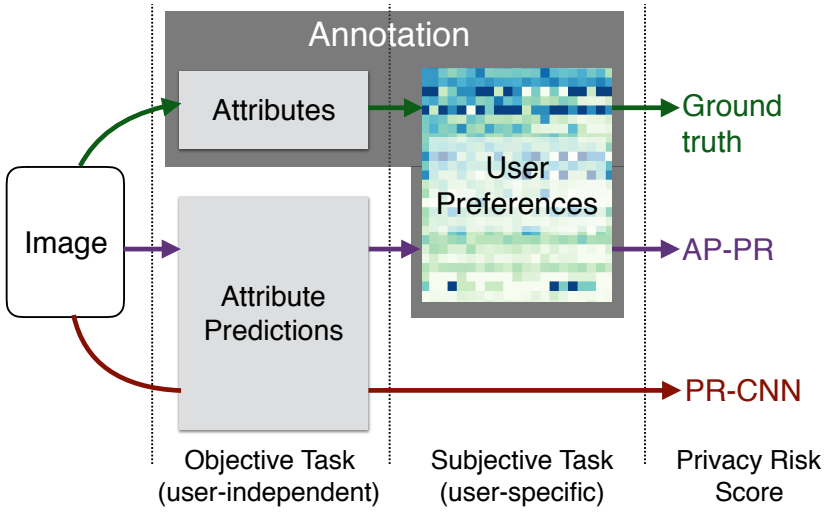


Figure 3.5: We learn an end-to-end model for user-specific privacy risk estimation.

Figure 3.6: Qualitative Results of our Privacy Attribute Prediction method

data and therefore has the potential to make this information explicit. Figure 3.1 shows multiple examples for this task. The task is challenging due to image diversity, subtle cues and high level semantics.

Metric. To assess performance of methods for this task, we compute the Average Precision (AP) per class, which is the area under Precision-Recall curve for the attribute. Additionally, the overall performance of a method is given by Class-based Mean Average Precision (C-MAP), the average of the AP score across all 68 attributes.

Methods. We experiment with three types of visual features extracted from CNNs – CaffeNet (Jia et al., 2014), GoogleNet (Szegedy et al., 2015) and ResNet-50 (He et al., 2016a). First, we train a linear SVM model using features from the layer preceding the last fully-connected layer of these CNNs. In a pilot study, we found that the

Training	Features	C-MAP
SVM	CaffeNet	37.93
	GoogleNet	39.88
	Resnet-50	40.50
End-to-End	CaffeNet	42.99
	GoogleNet	43.29
	Resnet-50	47.45

Table 3.2: Accuracy of our methods given by Class-based Mean Average Precision, evaluated on test

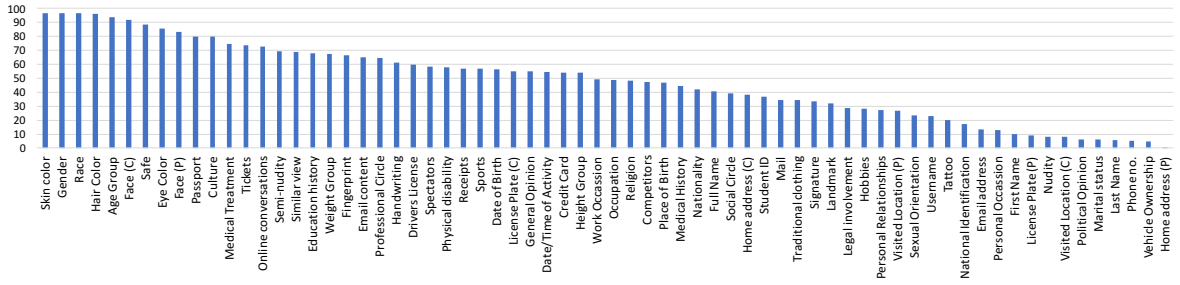


Figure 3.7: Average Precision (AP) Scores for the privacy attributes from our method

multilabel SVM with smoothed hinge loss (Lapin et al., 2016) yields better results than SVM multi-label prediction (Crammer and Singer, 2003) and cross-entropy loss. Second, we fine-tune the CNNs initialized with pretrained ImageNet models, based on a multi-label classification loss with sigmoid activations.

Results. Quantitative results of our method are shown in Table 3.2 and qualitative results in Figure 3.6. We additionally present the Average Precision scores per class in Figure 3.7. We make the following observations: (i) The CNN performs well in attributes such as tickets, passports, medical treatment that correlated well with scenes (e.g. airport, hospital). It also performs well in recognizing attributes which are human-centric, such as faces, gender and age; (ii) Fine-grained differences cause confusions such as predicting student IDs for drivers licenses or differentiating between street and other signboards; (iii) We observe failure modes due to small details in the image, such as tattoos, marriage rings or a credit card in the hands of a child; and (iv) A shortcoming of being unable to recognize relationship-based attributes (e.g., personal or social relationships, vehicle ownership) which requires reasoning based on interaction of multiple visual cues in an image rather than just their presence.

3.4.2 *Personalizing Privacy Risk Prediction*

In the previous section, we discussed predicting privacy attributes in images, a task independent of user privacy preferences. In this section, we investigate *user-specific* visual privacy feedback. The goal is to compute a *privacy risk score* per image, representing the risk of privacy leakage for the particular user.

Task. As illustrated in Figure 3.5, we combine privacy attributes (user independent) together with the privacy preferences based on these attributes (user specific) to arrive at the privacy risk score. We consider the *privacy risk score* of an image $\boldsymbol{x}$ as $\max_a y_a u_a$, where $\boldsymbol{y} \in [0,1]^A$ indicates presence of privacy attributes in the image and $\boldsymbol{u} \in [0,5]^A$ are the user preferences over the attributes. This represents the user-specific score of the most sensitive attribute, most likely to be present in an image. As a result, the privacy-risk score is comparable to the preference-score: 1 (Not Sensitive) to 5 (Extremely Sensitive). As illustrated in Figure 3.5, we compute the ground-truth privacy risk score based on ground-truth attribute annotation for an image (represented as a k-hot vector $\boldsymbol{y} \in \{0,1\}^A$) and privacy preferences of users.

Method: Attribute Prediction-based Privacy Risk (AP-PR). Our first method performs Attributed-Based Privacy Risk (*AP-PR*) prediction. As illustrated in Figure 3.5, we combine the privacy attribute prediction and the profile's privacy preferences (that we can assume as provided by users at test time) to compute the privacy risk score as defined above.

Method: Privacy Risk CNN (PR-CNN). We propose a Privacy Risk CNN (*PR-CNN*) that does not directly use the user profile's privacy preferences – but only indirectly via the ground-truth. The key observation is that AP-PR scores suffer from erroneous attribute predictions (see Figure 3.7). Therefore, we extend the the privacy attribute prediction network by additional fully-connected layers to directly predict the privacy risk score. A parameter search yielded best results using additional two fully-connected hidden layers of 128 neurons, each followed by sigmoid activations. We finetune this network from our Googlenet Privacy Attribute Prediction network for 30 user profiles described in Section 3.3 and a Euclidean loss.

Evaluation. We use two metrics for evaluation. First, the $L1$ error averaged over all images and profiles; it represents the mean absolute difference between the ratings. Secondly, we calculate the Precision-Recall curves for varying thresholds of sensitivity which indicates how well our models detect images above a certain true privacy risk. By calculating the area under the Precision-Recall curves over all user profiles, we additionally report the Mean Average Precision (MAP).

	L1-Error	MAP			
		1+	2+	3+	4+
AP-PR	0.656	**94.94**	**94.27**	87.97	77.89
PR-CNN	**0.637**	94.35	93.65	**88.14**	**78.38**

Table 3.3: Evaluation of Personalized Privacy Risk

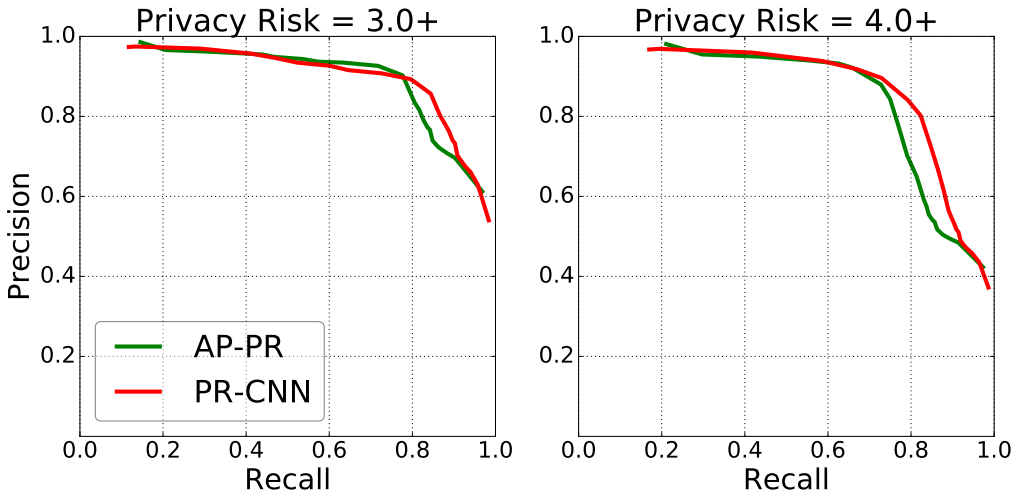


Figure 3.8: Performance of our approach in predicting Privacy Risks of images. Our approach performs better on high privacy-risk images.

In our experiments, we use the previously introduced user-profiles instead of individual users in order to cater to all the diverse privacy preferences equally that we have seen in the previous section. We assign a privacy risk score of 0.5 for the *safe* attribute for all profiles.

The evaluation of our approach on these metrics is presented in Table 3.3. Each graph in Figure 3.8 represents PR curves over the ground-truth thresholded to obtain a particular risk interval, such that any score above this threshold is considered private. This allows us to estimate performance of methods at various levels of sensitivity. We then obtain the PR-curves for each sensitivity interval by thresholding scores estimated by AP-PR and PR-CNN.

From these results, we observe: (i) PR-CNN performs better in predicting risk compared to using the intermediate attributes predictions. Notably, the prediction is on average less than one step on the scale from 1 to 5 away from the true privacy risk. (ii) Moreover, it is better at detecting high-risk images, as shown in Figure 3.8. In particular, we notice better recall for high-risk images.

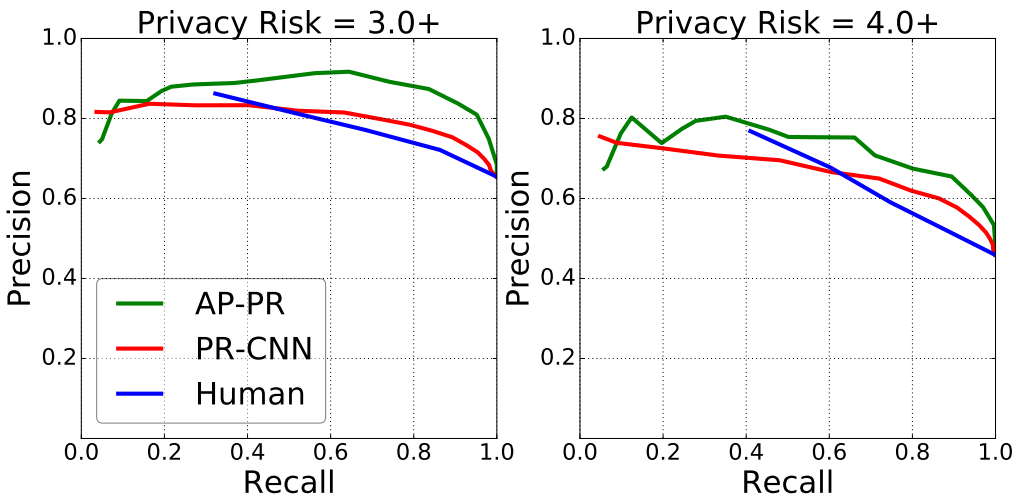


Figure 3.9: The Precision-Recall curves of three risk estimations are displayed – users implicitly evaluating risk from images and our two methods AP-PR and PR-CNN.

3.4.3 *Humans vs. Machine*

In Section 3.3, we have shown inconsistency in users' privacy preferences and their assessment of privacy risks in images. In this section, we compare our proposed approach for evaluating privacy risk against human judgments.

In our second user study (Section 3.3.2), for each attribute, users first assessed their personal privacy risk on images (providing a visual privacy risk score) and later rated their privacy preference (providing a desired privacy risk score). We have computed scores with our privacy risk models AP-PR and PR-CNN on those very same images.

As a result, for each image, we have (a) users' privacy preference (b) users' privacy risk judgment from images (c) our AP-PR privacy risk score from images (d) our PR-CNN privacy risk score from images. All these scores are on a scale of 1 (Not Sensitive) to 5 (Extremely Sensitive). Using the users desired preference as the ground-truth, we now ask: *who is better at reproducing the user's desired privacy preference on images?* As from the previous section, we use precision-recall and L_1-error as metrics to compare the desired preference score (a) and predicted privacy risk score for evaluation (b, c, d).

The precision-recall-curves for the three candidates are presented in Figure 3.9. We observe: (i) AP-PR achieves better precision-recall for the task than PR-CNN and – remarkably – is even *consistently better than the users' image-based judgment*. (ii) On average, the PR-CNN estimates privacy risks (L_1 error = 1.03) slightly better than the user's image-based judgment (L_1 error = 1.1) and AP-PR (L_1 error = 1.27).

3.5 CONCLUSION

In this chapter, we extended the concept of privacy settings to visual content and have presented work towards a *Visual Privacy Advisor* that can provide feedback to the users based on their privacy preferences. The significance of this research direction is highlighted by our user study which shows users often fail to enforce their own privacy preferences when judging image content. Our survey also captures typical privacy preference profiles that show a surprising level of diversity. Our new VISPR dataset allowed us to train visual models that recognize privacy attributes, predict privacy risk scores and detect images that conflict with user's privacy. In particular, a final comparison of human vs. machine prediction of privacy risks on images, shows an improvement by our model over human judgment. This highlights the feasibility and future opportunities of the overarching goal – a *Visual Privacy Advisor*.

In the next chapter, we extend our work to additionally identify privacy risks in images on a pixel-level. We additionally leverage the localized information to obfuscate corresponding pixels.

4

AUTOMATIC REDACTIONS

In the previous chapter, we were motivated to identify and control disclosure of a wide spectrum of private information in images. Consequently, we proposed a taxonomy of visual privacy attributes, a novel dataset, and methods to estimate *image-level* privacy leakage. In this chapter, we extend this line of work towards the goal of controlling leakage on a *pixel-level* by obfuscating relevant regions. By conducting a user study we find that obfuscating the image regions related to the private information leads to privacy while retaining utility of the images. Moreover, by varying the size of the regions different privacy-utility trade-offs can be achieved. Our findings argue for a "redaction by segmentation" paradigm.

Hence, we propose the first sizable dataset of private images "in the wild" annotated with pixel and instance level labels across a broad range of privacy classes. We present the first model for automatic redaction of diverse private information. It is effective at achieving various privacy-utility trade-offs within 83% of the performance of redactions based on ground-truth annotation.

The content of this chapter is based on Orekondy et al. (2018). As a first author, Tribhuvanesh Orekondy conducted all the experiments and was the main writer for the conference paper.

4.1 INTRODUCTION

More and more visual data is captured and shared on the Internet. Images and video contain a wide range of private information that may be shared unintentionally such as e.g. email-address, picture-id or finger-print (see Figure 4.1). Consequently, there is a growing interest within the computer vision community (Brkic et al., 2017; Hassan et al., 2017; Oh et al., 2016; Oh et al., 2017; Orekondy et al., 2017; Raval et al., 2017) to assess the amount of leaked information, understand implications on privacy and ultimately control and enforce privacy again. Yet, we are missing an understanding how image content relates to private information and how automated redaction can be approached.

Therefore, we address two important questions in this context. First, how can private information be redacted while maintaining an intelligible image? We investigate this question in a user study with highly encouraging results: we can redact private information in images while preserving its utility. Furthermore, varying the amount of pixels redacted results in different privacy vs. utility trade-offs. We conclude that redaction by segmentation is a valid approach to perform visual redactions.

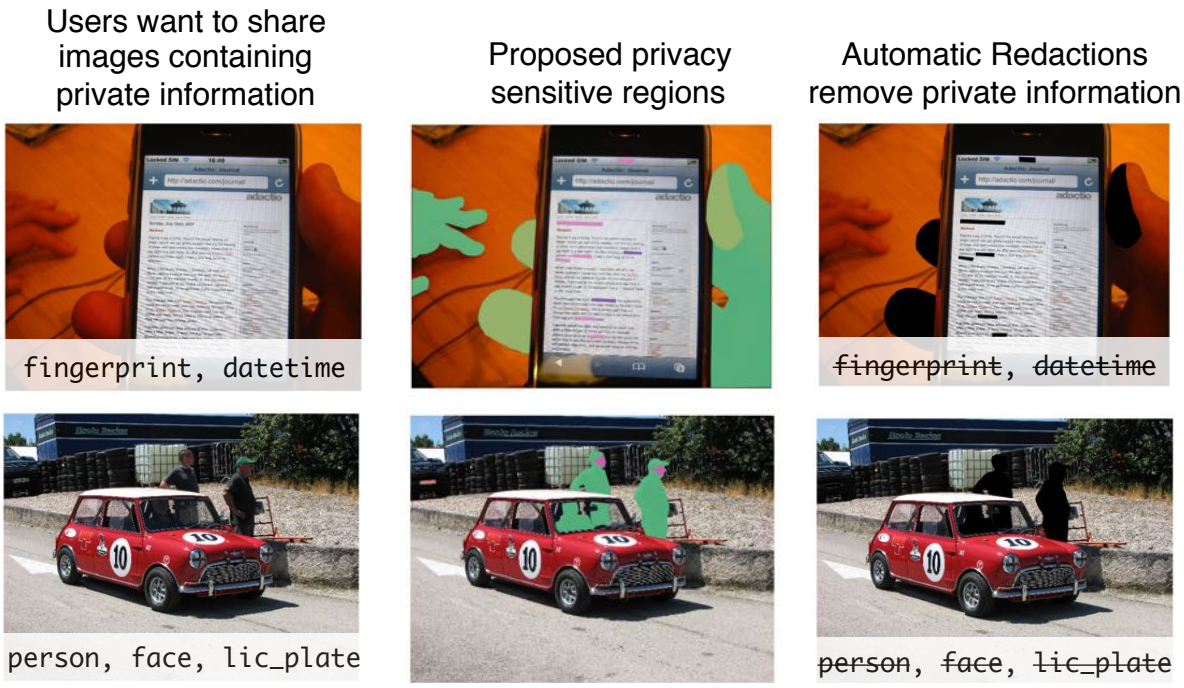


Figure 4.1: Users often share images containing private information, which poses a privacy risk. For example, in the top row, user might unintentionally leak their fingerprint. We present methods to aid users automatically redact such content by proposing privacy sensitive regions in images.

We ask a second question in this chapter: What kind of privacy-utility trade-offs can be achieved by automatic redaction schemes? Based on our first finding, we approach this as a pixel labeling task on multiple privacy classes (which we refer to as *privacy attributes*). Segmenting privacy attributes in images presents a new challenge of reasoning about regions including multiple modalities. For instance, in Figure 4.1, identifying the `name` and `datetime` requires mapping the relevant pixels to the text domain for understanding, while identifying the `student_id` requires reasoning over both visual and text domains. Our automated methods address these challenges and localize these privacy attributes for redaction via segmentation. By performing both quantitative and human evaluation, we find these automated methods to be effective in segmentation as well as privacy-utility metrics.

Our model and evaluation for automatic redaction is facilitated by a new dataset that extends the Visual Privacy (VISPR) dataset (Section 3.2) to include high-quality pixel and instance-level annotations. To this end, we propose a dataset containing 8.5k images annotated with 47.6k instances over 24 privacy attributes.

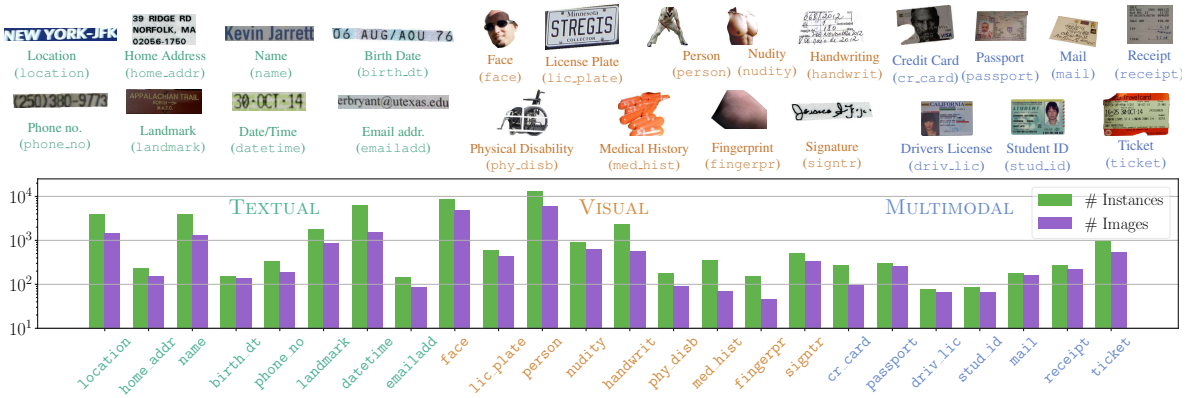


Figure 4.2: Examples and distribution of privacy attributes in the dataset.

4.2 THE VISUAL REDACTIONS DATASET

In this section we present our pixel-label visual privacy dataset as an extension to the VISPR dataset (Orekondy et al., 2017). We begin with a discussion on how images (Section 4.2.1) and attributes (Section 4.2.2) were selected for the task. This is followed by the annotation procedure (Section 4.2.3) and a brief analysis (Section 4.2.4) of the dataset.

4.2.1 *Selecting Images for Pixel-level Annotation*

The VISPR dataset contains 22k real-world user-uploaded publicly available Flickr images which makes this a great starting point for addressing the visual redaction problem "in the wild". 10k of these images are annotated as safe. From the remaining 12k images we pixel-annotate the subset of 8,473 images that contain at most 5 people. The main reason to focus on this subset was to reduce the annotation cost while maximizing the amount of non-person pixels. We preserve the identical 45-20-35 train-val-test split of these images as in the VISPR dataset.

4.2.2 *Shortlisting Privacy Attributes*

The 22k images in the multilabel VISPR dataset are annotated using 68 image-level privacy attributes (∼5.2 attributes per image). These privacy attributes are compiled from multiple privacy-relevant sources – the US Privacy Act of 1974, EU Data Protection Directive 95/46/EC and various social network website rules. Additionally, they cover a diverse range of private information that can be leaked in images (e.g. face, tattoo, physical disability, personal relationships, passport, occupation). Therefore, we use these as a starting point for redactions in images. We select 42 out of 67 privacy attributes (excluding attribute 'safe', which indicates

none of the other 67 attributes are present) for three reasons. First, for 11 attributes (e.g. religion, occupation, sports) typically the entire image is linked to the attribute (e.g. scene with church or sport stadium). In such cases, the solution to keeping the information private is to not share such images (as proposed in Orekondy et al. (2017)). We instead focus on attributes which can be localized for redaction, such that the image might still be useful. Second, 8 attributes were extremely tedious to annotate, because of their strong co-occurrence with crowd-scenes (e.g. political and general opinion, occupation) or the effort required to outline them (e.g. hair color). Third, 6 attributes (e.g. place of birth, email content, national id) contained under 30 examples for training. In spite of filtering such attributes, we still cover a broad spectrum of information to help de-identify people in images (such as by obfuscating faces or names). We further merge few groups among these 42 attributes: (i) when they occur as a complete and partial version (e.g. {complete face, partial face} merged into `face`) (ii) when they localize to the same region (e.g. {race, skin color, gender, relationships} merged into `person`). As a result, we work with 24 localizable privacy attributes in our dataset representative of 42 of the original 67 VISPR privacy attributes (see Figure 4.2 for the complete list).

4.2.3 *Dataset Annotation*

In this section, we discuss the annotation procedure.

Annotation tool and instructions. We use VGG Image Annotator tool (Dutta et al., 2016) for annotation. Five expert annotators draw polygons around instances based on an instruction manual. A summary of instructions, definitions of attributes and examples are provided in the supplementary material.

Consensus and agreement measure. Agreement is calculated w.r.t. images annotated by one of the authors. We measure agreement using Mean Intersection Over Union (mIoU): $\sum \frac{tp}{tp+fp+fn}$ averaged over images.

Consensus experiment and annotating `person`. We observed 93.8% agreement in consensus task of annotating instances of `person` in 272 images. Annotators separately annotated `person` in remaining images. We obtain 13,171 `person` instances annotated over 5,920 images.

Annotating `face`. We observed an agreement of 86.2% (lower due to small sizes of instances) in the consensus task for annotating `face` in 100 images. Using the 5,920 images of people as a starting point, annotators annotated 8,996 instances of faces in separate sets of images.

Annotating remaining attributes. Images for each of the remaining attributes are annotated successively by at most a single annotator. 8 of the text-based attributes (e.g. `name`, `phone_no`) are annotated using 4-sided polygons or bounding boxes. We gather annotation of 26,676 instances.

Auxiliary detections. We augment all images in the dataset with text detections obtained using the Google Cloud Vision API to aid localization of text-based attributes. This is provided as OCR and bounding box annotation in structured hierarchy of text elements in the order: characters, words, paragraphs, blocks and pages. In addition, we also gather face and landmark bounding box detections using the same API. These detections are solely used as auxiliary input to methods discussed in Section 4.4 and not for evaluation.

Summary. With an annotation effort of ∼800 hours concentrated over four months with five annotators (excluding the authors), we propose the first sizable pixel-labeled privacy dataset of 8,473 images annotated with ∼47.6k instances using 24 privacy attributes.

4.2.4 *Dataset Analysis and Challenges*

We now present a brief analysis of the dataset and the new challenges it presents for segmentation tasks. Examples of the proposed attributes and their distribution among the 8k images in the dataset are presented in Figure 4.2.

Popular datasets (Cordts et al., 2016; Everingham et al., 2010a; Lin et al., 2014) provide pixel-level annotation of various common visual objects. These objects are common in visual scenes, such as vehicles (car, bicycle), animals (dog, sheep) or household items (chair, table). Common to all these objects are their distinctive visual cues. Looking at the examples of attributes in Figure 4.2, one can notice similar cues among the VISUAL attributes, but it is not evident in the others. Recognizing TEXTUAL attributes (such as names or phone numbers) in images instead require detecting and parsing text information and additionally associating it with prior knowledge. While some of the MULTIMODAL attributes can be associated with visual cues, often the text content greatly helps disambiguate instances (a card-like object could be a `student_id` or `driv_lic`). We also observe a strong correlation between modalities and sizes of instances. We find TEXTUAL instances to occupy on average less than 1% of pixels in images, while MULTIMODAL attributes predominantly occur as close-ups occupying 45% of the image area on average. Consequently, the privacy attributes pose challenges from multiple modalities and require specialized methods to individually address them. Moreover, they provide different insights due to the variance in sizes. Hence, going forward, we treat the modes TEXTUAL, VISUAL and MULTIMODAL as categories to aid analysis and addressing challenges presented by them.

Figure 4.3: Dilation/Erosion of attribute `fingerprint`

Applicability to other problems. We believe the proposed dataset could be beneficial to many other problems apart from visual redactions. In visual privacy, it complements datasets to perform tasks such as person de-identification (Brkic et al., 2017; Hassan et al., 2017). Outside of the privacy domain, we also provide a sizable face segmentation dataset with 9k `face` instances, compared to 2.9k in Labeled Faces in the Wild (Kae et al., 2013) and 200 in FASSEG (Khan et al., 2015).

4.3 UNDERSTANDING PRIVACY AND UTILITY W.R.T. REDACTED PIXELS

In this section, we study how redacting ground-truth pixels of attributes influences privacy and utility of the image by conducting a user study on Amazon Mechanical Turk (AMT). The results from this section motivates our approach in Section 4.4. We will also use the results from this study as a reference point for evaluating our proposed automated methods in Section 4.5.2.

4.3.1 *Generating Redactions*

Given an image I_a containing attribute a, we generate a ground-truth redacted version of the image $I_{\bar{a}}$ by simply blacking-out pixels corresponding to a in the ground-truth.

Spatially extending a. We now want to redact fewer or more pixels in image $I_{\bar{a}}$ to understand how this influences the image's privacy and utility. We generate multiple versions of the ground-truth redacted image $\{I^s_{\bar{a}} : s \in S\}$ at different scales of redaction, such that $I^{ns}_{\bar{a}}$ contains n times as many blacked-out pixels of $I^s_{\bar{a}}$. We achieve different scales of redactions by dilating/eroding the ground-truth binary mask of a, as shown in Figure 4.3. We use seven scales $S = \{0.0, 0.25, 0.5, 1.0, 2.0, 4.0, \text{inf}\}$, where $I^0_{\bar{a}}$ is the unredacted image, $I^1_{\bar{a}}$ $(= I_{\bar{a}})$ is the GT redacted image and $I^{\text{inf}}_{\bar{a}}$ is a completely blacked-out image.

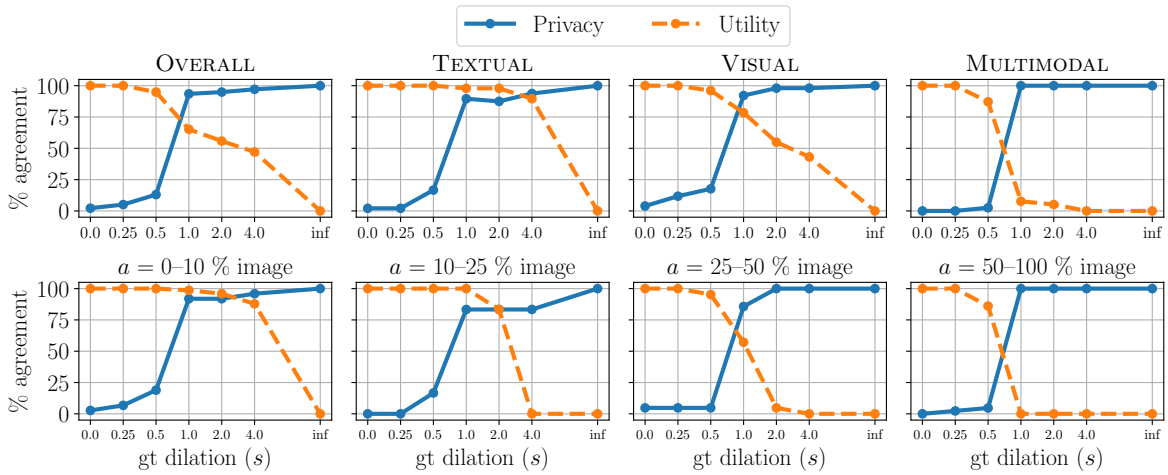


Figure 4.4: Privacy and utility using various scales of ground-truth redaction over (Top row) modes (Bottom row) sizes

4.3.2 *User Study*

We create an AMT project of 1,008 tasks (24 attributes $\times$ 6 images $\times$ 7 scales), each to be responded by 5 unique workers from a pool of 29 qualified workers. Each task contains 2 yes/no questions based on an image $I^s_{\tilde{a}}$, one each for Privacy and Utility. We consider *privacy* and *utility* w.r.t. (i) two versions of the same image: $(I_a, I^s_{\tilde{a}})$, and (ii) users (AMT workers in our case).

Defining privacy. To understand if attribute a has been successfully redacted in $I^s_{\tilde{a}}$, we pose the privacy question in the form: "Is a visible in the image?". We also provide a brief description of the attribute a along with examples. We consider $I^s_{\tilde{a}}$ to be *private*, if a majority of the users respond *no*.

Defining utility. To understand utility of an image, we pose the question: "Is the image intelligible, so that it can be shared on social networking websites? i. e.does this image convey the main content of the original image (i.e., the image without the black patch)". As a result, we define the utility of an image independent to its aesthetic value and instead associate it with the semantic information. We consider $I^s_{\tilde{a}}$ to have *utility*, if a majority of the users respond *yes*.

Measuring privacy and utility. We label each of the 1,008 images with varying redacted scales their privacy and utility as discussed above. For any given redaction scale s, we aggregate privacy/utility scores simply as the percentage of images considered private/useful. Consequently, an ideal visual redaction has both high privacy and utility.

4.3.3 *Analysis*

We now discuss results based on the privacy-utility scores obtained over modes and various sizes (i. e.relative size of a in I_a) based on Figure 4.4.

Privacy is a step function. We observe in Figure 4.4 across all plots, that a minimum number of pixels of attribute a need to be removed to effectively redact it from the image. This minimum number corresponds to exactly the ground-truth redaction ($s = 1$) – redacting fewer pixels than this makes the image non-private and redacting more pixels achieves marginal privacy gains. More specifically, we achieve 94% privacy with ground-truth redactions. The imperfect privacy score is predominantly (5/9 failure cases) due to turkers overlooking important details in the question. Apart from this, other cases involve contextual cues revealing the attribute (e.g. wheelchair shadow) and regions that were not annotated (e.g. outline of a `person` at a distance).

Gradual loss in utility. From Figure 4.4 OVERALL, we find utility to decrease gradually as the size of redacted region increases. Another interesting observation is that utility strongly depends on the size of a in the image. In the bottom row of Figure 4.4, we see that for smaller GT regions ($a = 0 - 10\%$), we still obtain high utility at larger dilations. However, as the area of the GT regions increases beyond 50% of the image, redaction entails blacking-out the majority of the image pixels and hence zero utility.

Privacy and utility. What can we take away from this while proposing automated methods to preserve privacy while retaining utility? Due to the correlation between modes and sizes, we can predict more pixels for smaller attributes with minimal loss to utility. For instance, for TEXTUAL attributes, we can predict 4x as many ground-truth pixels for redaction. However, for larger ground-truth regions (>50% of image) both privacy and utility are step functions and hence making redaction a choice between privacy and utility.

GT segmentations are a good proxy. In general, for images over all attributes and sizes (Figure 4.4 OVERALL), we see that we can already achieve high privacy *while* retaining considerable utility of the image. Moreover, we obtain near-perfect privacy with the highest utility in all cases at $s = 1$, the ground-truth redactions. This justifies to address privacy attribute redaction as a segmentation task.

4.4 PIXEL-LABELING OF PRIVATE REGIONS

In Section 4.2 we discussed the challenges of attributes occurring across multiple modalities (TEXTUAL, VISUAL, MULTIMODAL). In Section 4.3, we motivated how

ground-truth segmentations in our dataset make a good proxy for visual redactions. In this section we propose automated methods to perform pixel-level labeling (semantic segmentation) of privacy attributes in images, with an emphasis on methods tackling each modality.

We begin with a simple baseline **Nearest Neighbor (NN)**: A 2048-dim feature is extracted using ResNet-50 for each image. At test time, we predict the segmentation mask of the closest training image in terms of L_2 distance.

4.4.1 *Methods for* TEXTUAL*-centric attributes*

To facilitate segmenting textual attributes, for each image we first obtain an ordered sequence of bounding box detections of words and their OCR using the Google Cloud Vision API (as discussed in Section 4.2.3).

Proxy GT. We represent n words in an image as a sequence $[(w_i, b_i, y_i)]_{i=1}^{n}$, where w_i is the word text, b_i is the bounding box and y_i is the label. We use 9 labels (8 TEXTUAL attributes + safe). We assign each y_i in the sequence the ground-truth attribute that maximally overlaps with b_i, or a *safe* label in case of zero overlap. At test-time, we segment pixels in region b_i if a non-safe label is predicted for word w_i. For the test set, we refer to predictions from this proxy dataset as **PROXY** to obtain an upper-bound for our methods on these text detections.

Rule-based classification (RULES). We use the following rules to label words in the sequence: (i) `name`: if it exists in a set of 241k names obtained from the US Census Bureau website (ii) `location, landmark, home_address`: if it exists in a set of 2.8M locations consisting of countries, states, cities and villages from the GeoNames geographical database (*GeoNames Geographical Database*) (iii) `datetime, phone_no, birth_dt`: if the word contains a digit (iv) `emailadd`: if the word contains the symbol @, we predict this word and adjacent words assuming a format □@□.□.

Named entity recognition (NER). We use the popular Stanford NER CRFClassifier (Finkel et al., 2005) to label each word of the sequence as from a set of recognized entity classes (e.g. person, organiziation, etc.). We use the model which is trained on case-invariant text to predict one of seven entity classes.

Sequence labeling (SEQ). We train a sequence labeler similar to (Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016) as shown in Figure 4.5. We preprocess by replacing all digits with 0s and stem each word to reduce the size of the vocabulary. We tokenize the words in the training sequences using a vocabulary of size 4,149 (number of words with at least 4 occurrences). We embed the words using 100-d GloVe embeddings (Pennington et al., 2014). To capture the temporal nature, we use two-level Bidirectional LSTMs. At each time-step, we obtain a joint embedding by

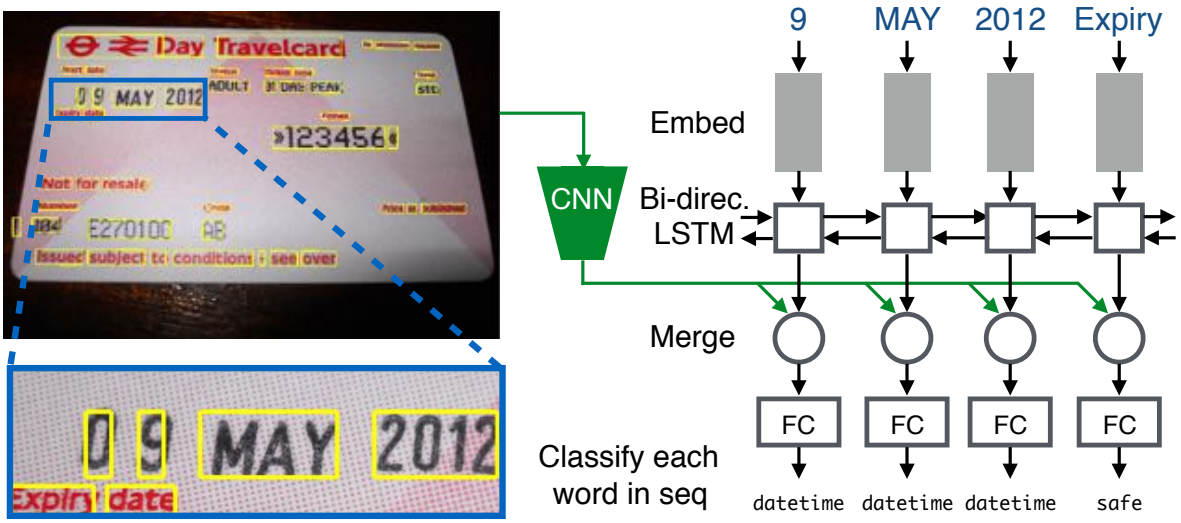


Figure 4.5: Architecture to perform sequence labeling

element-wise multiplication of: the *text* embedding (256-d output of the LSTM) and the *image* embedding (2048-d ResNet-50 (He et al., 2016b) feature reduced to 256-d using an FC layer). We classify this joint embedding into 9 labels using an FC layer followed by softmax activation.

4.4.2 *Methods for* Visual*-centric attributes*

Recent deep-learning segmentation methods have proven to be effective in localizing objects based on their visual cues. We propose using a state of the art method in addition to few pretrained methods for Visual attributes.

Pretrained models (PTM). We use pretrained methods to classify three classes typically encountered in popular visual scene datasets. (i) `face`: We use bounding box face detections obtained using the Google Cloud Vision API. (ii) `person`: We use the state-of-the-art segmentation method FCIS (Li et al., 2017b) to predict pixels of COCO class "person" (iii) `lic_plate`: We use OpenALPR (*OpenALPR*) to detect license plates in images.

FCIS. We retrain all layers of the FCIS model (Li et al., 2017b) for our task and dataset. We train it for 30 epochs with learning rate 0.0005 over trainval examples and their horizontally mirrored versions. We fine tune it from the model provided by the authors trained for segmentation on MS-COCO (Lin et al., 2014). We obtained best results using default hyper-parameters.

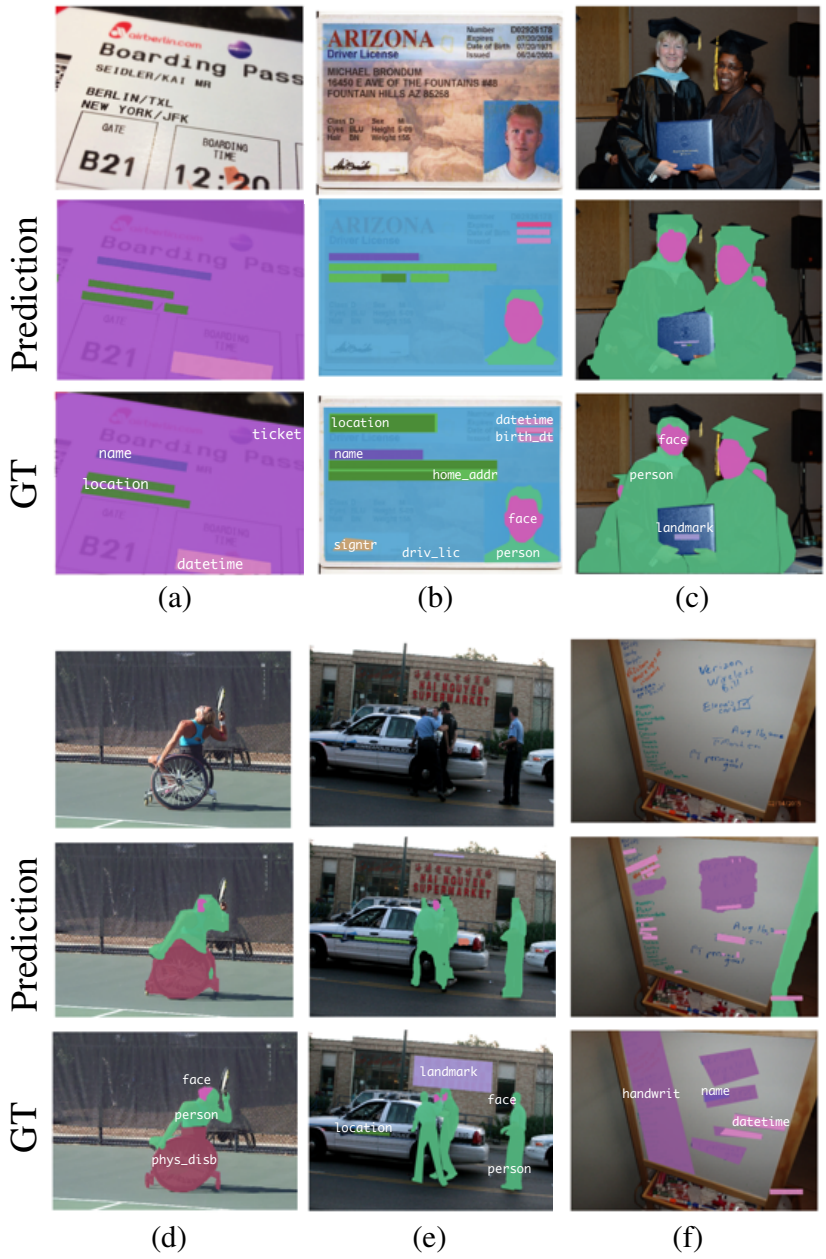


Figure 4.6: Qualitative examples from our method

4.4.3 *Methods for* MULTIMODAL*-centric attributes*

Recognizing Multimodal attributes (e.g. `driv_lic`, `receipt`) require reasoning over both visual and textual domains. We treat this as a classification problem due to: (i) limited training examples ($\sim$125 per multimodal attribute) (ii) large region of these attributes ($\sim$45% image area), which provides only $\sim$10% utility even after GT-based redaction (Section 4.3.2).

Weakly supervised labeling (WSL). We propose learning a multilabel classifier based on visual-only (**WSL:I**) and visual+text content (**WSL:I+T**). If the class probability of an attribute is beyond a certain threshold, we predict all pixels in the image for the attribute. WSL:I is the same approach used in (Orekondy et al., 2017) – a multilabel ResNet-50 (He et al., 2016b) classifier. In the case of WSL:I+T, we obtain a multimodal embedding by concatenating visual and text representations. We obtain visual representation (identical to WSL:I) with a ResNet-50 architecture. We obtain text representation by encoding all words in the image. We tried three such variants: (i) *Bag-of-Words (BOW) encoding*: Words in the image are represented as a one-hot vector with vocabulary of size 1,751. (ii) *LSTM encoding*: Identical to SEQ, we encode the word sequence using an LSTM with 128-hidden units. We use output from the last cell as the text representation. (iii) *Conv1D encoding*: We use 1D convolutions to encode the word sequence (typically used for sentence classification tasks (Kim, 2014)) followed by max pooling to obtain a fixed-size text representation In all three cases, we reduce the text-representation to 512-d using an FC+ReLU layer. We report BOW encoding results for **WSL:I+T** in the rest of the chapter since this provided the best results.

Salient object prediction (SAL). Using WSL:I+T as the base classifier, we use the salient object as an approximation of the attribute's location. We obtain class-agnostic saliency obtained using DeepLab-v2 ResNet (Chen et al., 2015; Joon Oh et al., 2017).

Weakly supervised iterative refinement (IR). For document-like objects, the text regions tend to be densely clustered in images. Hence, after classification using WSL:I+T, we refine the convex hull of the text regions using DenseCRF (Krähenbühl and Koltun, 2011) to "spill into" the document region.

4.5 EXPERIMENTS AND DISCUSSION

In this section, we discuss segmentation performance (Section 4.5.1) and privacy-vs-utility performance (Section 4.5.2) of our proposed methods.

4.5.1 *Evaluating Segmentation Performance*

We now evaluate methods proposed in Section 4.4 in terms of its segmentation performance using Mean Average Precision, suggested in Pascal VOC (Everingham et al., 2010a). This is calculated by averaging area under precision-recall curves over the privacy attributes. We use 50 thresholds uniformly spaced between 0 and 1 to obtain this curve. At each threshold t, we: (i) binarize the prediction score masks per image by thresholding pixel-level scores at t (ii) aggregate pixel-level TP, FP, FN counts (normalized by image size) per attribute over all images to obtain attribute-level precision and recall. We ignore GT masks containing under 25^2 pixels during evaluation ($<$1% GT masks).

Table 4.1 presents the quantitative results of the proposed methods on the test set. Qualitative results in Figure 4.6 are based on an **ENSEMBLE**, using predictions of SEQ for TEXTUAL, FCIS for VISUAL, WCS:I+T for MULTIMODAL attributes. Auxiliary results and analysis are available in the supplementary material. We generally observe that NN underperforms simple baselines across all modalities, highlighting the difficulty and diversity presented by the dataset.

Textual. We observe: (i) *Patterns, frequency and context*: SEQ achieves the best overall score, justifying the need for special methods to tackle text attributes. It is reasonably effective in detecting `datetime` (timestamps, Fig. 4.6a), `email` (email addresses) and `phone_no` (phone numbers) due to patterns they often display. We additionally find SEQ detect attributes which often require prior knowledge (e.g. `name`, `location`). The common success modes in such cases are when the words are popular entities (e.g. "Berlin" in Fig. 4.6a) or have discriminative visual/textual context (e.g. detecting `homeadd` in Fig. 4.6b). (ii) *Challenges imposed by text detections*: PROXY represents an upper bound to our textual methods. The low scores highlights the difficulty of text detection and this is especially severe for scene and handwritten text detection, a frequent case in our dataset (e.g. Fig. 4.6e,f). Moreover, our text detections do not perfectly overlap with ground-truth annotations. Since text regions are small, we additionally pay a high performance penalty even for correct detections (e.g. IoU=0.42 for `homeadd` (home addresses) in Fig. 4.6b). Moreover, even in the case of correct text detections, we observe failures in OCR which affects the quality of input for dependent methods. This can be observed by the under-performance of NER, which is typically very effective on clean sanitized text.

Visual. We observe: (i) *The unreasonable effectiveness of FCIS*: We obtain the highest score in the VISUAL category using FCIS. We find FCIS to be highly effective localizing visual objects commonly encountered in other datasets (e.g. `person`, `face`). Moreover, we find it achieves reasonable performance even when there is a lack of training data e.g. only $<$60 examples of `fprint` (fingerprints), `phys_disb` (physical disability); see Fig. 4.6d. The common failure modes are difficult examples (e.g. `face` in Fig. 4.6e)

TEXTUAL

Method	mAP	location	homeaddr	name	birthdt	phoneno	landmark	datetime	email
PROXY	45.0	31.7	37.8	48.7	52.5	52.6	33.6	52.4	50.8
NN	0.9	0.3	1.9	0.4	*0.7*	0.0	3.1	0.6	0.0
NER	3.0	*6.0*	1.7	4.4	0.5	0.0	0.5	10.9	0.0
RULES	4.2	3.1	0.5	2.8	0.6	1.4	1.2	6.4	*17.5*
FCIS	*7.2*	4.3	0.2	*9.8*	0.1	*2.5*	**27.6**	*12.9*	0.0
SEQ	**26.8**	**18.4**	**19.4**	**19.1**	**25.1**	**45.8**	*13.9*	**33.4**	**38.9**

VISUAL

Method	mAP	face	lplate	person	nudity	hwrit	phydisb	medhist	fprint	sign
NN	16.6	9.0	16.0	33.6	6.2	37.5	11.4	18.9	16.9	0.1
WSL:I	*20.8*	5.0	4.3	30.3	*16.4*	*49.9*	*13.7*	*37.7*	*28.8*	*1.3*
PTM	20.0	*47.6*	*44.5*	**88.3**	0.0	0.0	0.0	0.0	0.0	0.0
FCIS	**68.3**	**83.8**	**77.9**	*87.0*	**69.7**	**80.7**	**59.0**	**45.8**	**68.1**	**42.6**

MULTIMODAL

Method	mAP	cr_card	passport	driv_lic	stud id	mail	receipt	ticket
NN	24.1	10.5	49.5	19.9	14.5	20.6	17.1	36.7
WSL:I+T	*55.6*	27.7	*68.8*	**83.3**	**56.1**	**41.4**	54.2	58.0
SAL	36.2	**55.9**	37.2	23.8	30.4	8.1	42.5	55.1
IR	53.6	41.7	51.2	67.8	48.1	36.9	*57.2*	*72.5*
FCIS	**59.2**	*53.2*	**76.3**	66.5	*50.3*	33.1	**59.4**	**75.4**

Table 4.1: Quantitative results of our methods for segmenting privacy regions. **Bold** numbers denote highest and *italicized* numbers second highest scores in the columns.

and uncommon visual objects e.g. sign (signatures) in Fig. 4.6b. (ii) *Comparison with Baselines*: PTM achieves comparable results for person, due to Flickr images used to train both models. However, it underperforms for face (detections are not precise enough) and lplate (license plates; poor performance in the wild).

Multimodal. We observe: (i) *WSL:I is a good simple baseline*: WSL:I achieves reasonable performance (45.4) for multimodal attributes, compared to other modes (1.5 in text and 20.8 in visual) although the prediction spans the entire image. This is attributed to large size of MULTIMODAL instances found in images. (ii) *Multimodal reasoning helps*: We find WSL:I+T improves performance over WCS:I by 20%, justifying the need for methods to perform multimodal reasoning to detect these attributes. This is particularly necessary to disambiguate similar looking visual objects (e.g. card-like objects driv_lic (driver's license) and stud_id (student identity card), Fig. 4.6b). (iii) *Precision-Recall trade-off*: We find precision for WSL:I+T for this method can be improved for some attributes (e.g. cr_card (credit card), ticket) by IR, which

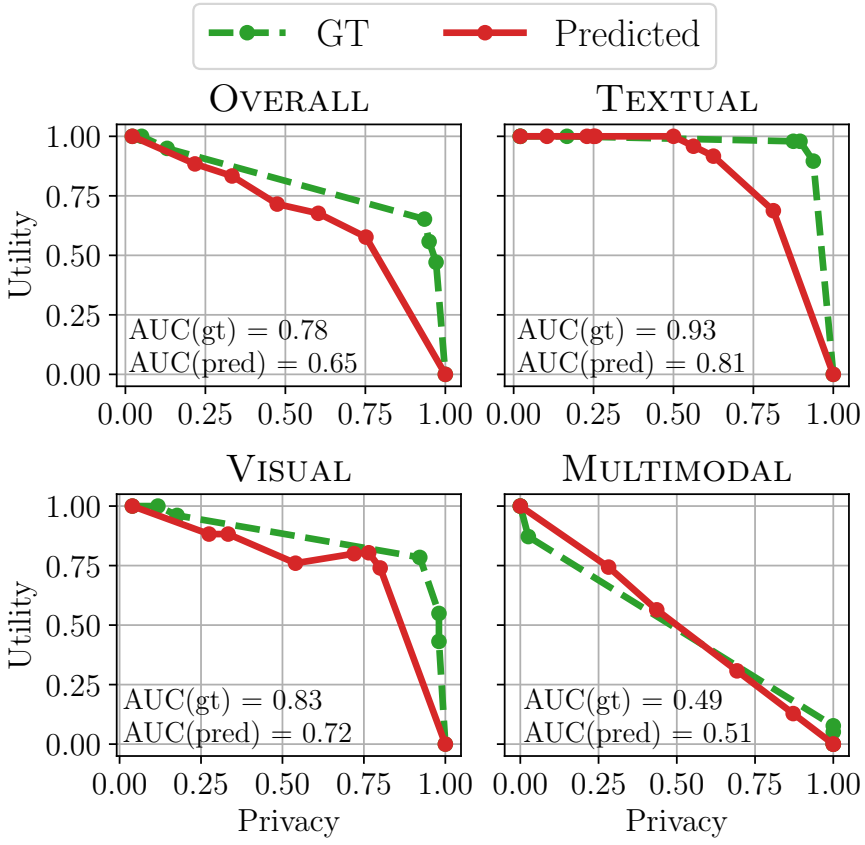


Figure 4.7: Comparing redactions using predicted and ground-truth segmentations

instead of the entire image, predicts only the smoothened hull of text regions. We observe FCIS achieve the best overall score due to higher precision.

4.5.2 *Privacy vs. Utility Trade-off by Automatic Redaction*

In the previous section, we evaluated our approaches w.r.t. segmentation quality. Now, we ask how effective are redactions based on our proposed methods in terms of privacy and utility?

To answer this, we once again run the user study in Section 4.3.2 on AMT, but now by redacting proposed pixels of our automated method over those exact images. To vary the number of predicted pixels, we vary the threshold to binarize the predicted score masks over attributes. As a result, we obtain 6-8 redacted versions for each of the 144 images (24 attributes $\times$ 6 images). Each image is labeled by 5 unique qualified AMT workers.

Results. We obtain privacy-utility scores for each threshold and plot it as a curve in Figure 4.7. We also plot the scores obtained for different dilations of redacted ground-

truth annotated region. It should be noted that perfect redactions are unavailable to us and we use these ground-truth based redactions (or manual redactions) only to serve as a reference. We evaluate performance by calculating area under the curve (AUC). We observe: (i) Overall, we find our method obtain a privacy-utility score of 65% – a relative performance of 83% compared to redactions using ground-truth annotation from the dataset. (ii) MULTIMODAL attributes present a hard choice between privacy and utility, as these regions are often large. We find the slightly lower AUC(gt) to be an artifact of sampling. (iii) Although we obtain a low mAP for TEXTUAL attributes, we observe an 81% privacy-utility score. This occurs as we can now over-predict regions, exhibiting low precision and high recall w.r.t. segmentation, but yet retaining high utility due to their small size. Consequently, we can predict more text pixels "for free".

Based on these observations, we find the automatic redactions of our models trained on the proposed dataset show highly promising results – they closely mimic performance achieved by redacting ground-truth regions across a broad range of private information.

4.6 CONCLUSION

We proposed a redaction by segmentation approach to aid users selectively sanitize images of private content. To learn automated approaches for this task, we proposed the first sizable visual redactions dataset containing images with pixel-level annotations of 24 privacy attributes. By conducting a user study, we showed that redacting ground-truth regions in this dataset provides near-perfect privacy while preserving the image's utility. We then presented automated approaches to segment privacy attributes in images and observed that we can already reasonably segment these attributes. By performing a privacy-vs-utility evaluation of our automated approach, we achieved a highly encouraging 83% performance w.r.t. GT-based redactions.

Part II

LEAKAGE DURING TRAINING

The previous part addressed leakage of information in raw data. Specifically, techniques to identify and control (e.g., obfuscating) a wide range of private information in image content prior to sharing on social media. We now consider the case where the intent of sharing is not social media, but rather to train an ML model collaboratively with other individuals. In such a case, the individual can instead intermittently share minimal information (model parameter updates) during the training process. Consequently, we switch focus from analyzing information leakage in raw data to analyzing leakage in training artifacts.

In Chapter 5, we study leakage of unintentional information in model parameter updates communicated during Federated Learning. We find that the updates encode user-identifiable signals leading to deanonymization risks. Additionally, the chapter presents techniques to mitigate leakage of user-identifiable information.

5

UNDERSTANDING AND CONTROLLING DEANONYMIZATION IN FEDERATED LEARNING

Until now, we addressed leakage in raw data and specifically, visual content representative of personal photos. Here, the goal of the user (the data owner) was to share the raw sensor data (e.g., images captured on smartphones) on the internet (e.g., social networks). Now, we switch focus to an alternate goal of sharing data towards training of a ML model. Since a single user's data might be insufficient to train a powerful and complex model, we specifically consider *collaboratively* trained ML models, where multiple users contribute training data to an aggregator (or a server). However, as we saw in the previous part, raw data contains many pieces of orthogonal and private information (e.g., identities, race) that might be irrelevant for training a general ML model (e.g., car detector).

In such cases, Federated Learning (FL) systems are gaining popularity as a solution to training Machine Learning (ML) models from large-scale user data collected on personal devices (e.g., smartphones) without their raw data leaving the device. At the core of FL is a network of anonymous user devices sharing training information (model parameter updates) computed locally on personal data. However, the type and degree to which user-specific information is leaked in the model updates is poorly understood. In this chapter, we identify model updates encode subtle variations in which users capture and generate data. The variations provide a strong statistical signal, allowing an adversary to effectively deanonymize participating devices using a limited set of auxiliary data. We analyze resulting deanonymization attacks on diverse tasks on real-world (anonymized) user-generated data across a range of closed- and open-world scenarios. We study various strategies to mitigate the risks of deanonymization. As random perturbation methods do not offer convincing operating points, we propose data-augmentation strategies which introduces adversarial biases in device data and thereby, offer substantial protection against deanonymization threats with little effect on utility.

The content of this chapter is based on the technical report Orekondy et al. (2020a), which is currently under review. A short version of the report (Orekondy et al., 2019a) was presented as an oral presentation at the Workshop on Federated Learning for Data Privacy and Confidentiality in conjunction with NeurIPS 2019. As a first author, Tribhuvanesh Orekondy conducted all the experiments and was the main writer for the paper.

5.1 INTRODUCTION

Advances in machine learning (ML) is increasingly fueled by accessibility to data sources capturing rich representations of the world e.g., 9M photographs (Krasin et al., 2017), 1.6M tweets (Go et al., 2009), etc. While such large-scale data advances learning fundamental ML models (e.g., visual object recognition), the representations also encode a massive amount of unnecessary individual-specific information (e.g., person identities) (Orekondy et al., 2017; Gurari et al., 2019). For situations where the data is decentralized (e.g., user-generated photos on edge devices), Federated Learning (McMahan et al., 2017) provides a solution based on the principles of data minimization (House, 2012; European Union, 2016) towards training a ML model. The core idea is participants distill from raw private data residing on individuals' device the information necessary to train the model, and intermittently communicate them to a server. The information communicated by the participants take the form of model updates computed locally on-device.

To prevent privacy violations, it is crucial that model updates reveals information solely necessary for the training task (e.g., visual features to identify cats) and nothing about the participants (e.g., person identities). To ensure this, federated learning is combined with additional steps to restrict the amount of data- and participant-specific information revealed in the process. In the specific case of restricting participant-specific information encoded in model updates, typical steps include: stripping the data of PII information (Yang et al., 2018), de-identifying the updates and auxiliary metadata (Yang et al., 2018; Hard et al., 2018; McMahan et al., 2017), and avoiding authentication via user-identity prior to participation (Bonawitz et al., 2017). Hence, it is assumed that model updates received by the server contains minimal non-identifiable information to improve the model.

However, it is in the nature of many real-world federated settings, that the clients represent diverse users with different interests, preferences and habits. Hence, the underlying data distributions of the users are not identically distributed and as a consequence, is characteristic of the users. Therefore, we find that the model updates nonetheless encode individual-specific information and introduce *significant* deanonymization risks. Apart from constituting a privacy violation, deanonymization in federated learning undermines existing mechanisms to ensure the source of model updates are masked. Furthermore, deanonymization amplifies effectiveness of recent inference attacks (e.g., attribute inference (Melis et al., 2019)), as identities can be tied to sensitive attributes inferred from the participants' private training data.

We investigate deanonymization risks and consequences by following the popular Federated Averaging algorithm (McMahan et al., 2017; Bonawitz et al., 2019), where participating devices intermittently communicate de-identified model parameter updates to a server. Here, the high-dimensional updates are a product of multiple gradient steps on multiple batches of the local device data. We assume honest-but-

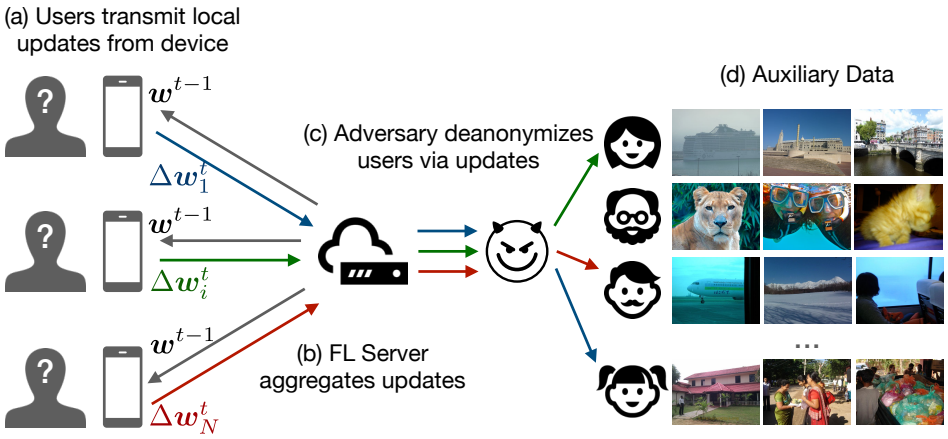


Figure 5.1: Deanonymization in federated learning. In this chapter, we study how subtle user-biases captured in model parameter updates leads to deanonymization of their devices.

curious server who intends to deanonymize participating devices (Fig. 5.1c) with limited access to prior information of users (Fig. 5.1d). Central to our deanonymization attack is exploiting subtle, but inherent, individual- specific biases introduced when participants collect data on personal devices. For instance, Alice capturing more photos of automobiles on her mobile device compared to Bob, who photographs food. Our approach learns a suitable representation where the biases (modeled from limited prior data) can be leveraged to re-identify individuals via their model updates.

We evaluate deanonymization risks in a federated learning setup when training complex models (e.g., MobileNet CNNs (Howard et al., 2017)) involving numerous participants (53-327 users). Furthermore, we use real-world (anonymized) user-generated datasets (e.g., PIPA, Blog) to closely emulate existing federated learning applications (McMahan et al., 2017; McMahan et al., 2018). Our evaluation indicates that participants can be consistently deanonymized across a range of scenarios. For instance, individuals transmitting model updates for an image classifier (with output classes e.g., chair, umbrella) on PIPA dataset are re-identified with high accuracy (19-175$\times$ chance-level). Furthermore, we find the attacks surprisingly possible in spite of a range of data-limited scenarios, such as when the adversary has only a single prior example of the targeted individual.

Moreover, we propose a novel cross-modal attack which tackles a challenging scenario when the attacker's prior information varies in modality from the private data used during training by the participants. For instance, the attacker leverages text information, while the participants are training using image data. Our experiments indicate that in spite of the cross-modal challenge, attacks are quite effective (0.76 AUC).

It is worth noting that our deanonymization attack can also amplify the performance of recent attacks that infer sensitive properties of the training data. For

example, we show that learning an attack model to jointly perform deanonymization and attribute inference (Melis et al., 2019) are synergistic, with a consistent improvement of up to 4% accuracy on both tasks. These results are further concerning, as sensitive attributes can be linked to identities of participants in federated learning.

After demonstrating the the risks of deanonymization in federated learning, we explore countermeasures to mitigate the threat. We propose augmenting users' data distribution with an adversarial bias to decouple users' subtle variations from their prior information. As a result, we propose the first mitigation strategy that directly operates on the user data itself, while maintaining utility of the task. We find our strategy mitigate attacks with up to 95% effectiveness and incurs only negligible cost on the underlying task performance. In contrast, we find perturbation- and DP-based training approaches (e.g., DP-FedAvg (McMahan et al., 2018)) incur large privacy and utility costs in our setup as they are typically effective only when training with a massive number of users (in the order of thousands).

5.2 BACKGROUND, NOTATION AND TERMINOLOGY

In this section, we provide the preliminaries to Federated Learning, within which we explore our threat model in the next section. At this point, we remark that research towards a Federated Learning system encompasses among many other things, architecture (Bonawitz et al., 2019), optimization techniques (Konečnỳ et al., 2016a; McMahan et al., 2017), strategies to improve communication (Konečnỳ et al., 2016b), aggregation (Bonawitz et al., 2017), implementation (Abadi et al., 2016a), and applications (Chen et al., 2019; Yang et al., 2018; Hard et al., 2018). To keep the background in this section concise, we present key concepts to understand: (i) how devices generate model parameter updates using the `FederatedAveraging` (McMahan et al., 2018) algorithm; and (ii) how users anonymously communicate the parameter updates to the server in FL (Bonawitz et al., 2019; Melis et al., 2019; Nasr et al., 2019).

Notation and learning objective. In supervised learning, the overall objective is to learn a mapping $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ of a model f parameterized by $\boldsymbol{w} \in \mathbb{R}$. The idea is to learn the parameters which minimizes the empirical risk represented by a loss function L on a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} H(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i} L(f_{\boldsymbol{w}}(\boldsymbol{x}_i),\ \boldsymbol{y}_i) \tag{5.1}$$

In FL, data is partitioned across multiple devices $k \in \mathbb{K}$: $\mathcal{D} = \bigcup_k \mathcal{D}_k$. Using $H_k(\boldsymbol{w})$ to denote the objective solved locally on device k, the objective in Equation 5.1 can now be re-written as:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \sum_{k=1}^{K} \frac{n_k}{n} H_k(\boldsymbol{w}) \tag{5.2}$$

Algorithm 1: `FederatedAveraging` (McMahan et al., 2017) for training data on multiple devices

Server's algorithm:

Input: K devices; T number of rounds; C fraction of devices sampled each round; B device's batch size; E number of local epochs

Randomly initialize $w^{t=0}$

for *round* $t \leftarrow 1$ **to** T **do**

  $M \leftarrow \max(1,\ C \cdot K)$

  $\mathbb{K}_t \leftarrow$ sample M devices from $\mathbb{K}$

  for *client* $k \in \mathbb{K}_t$ **do**

    $\Delta \boldsymbol{w}_k^{t+1} \leftarrow \texttt{DeviceUpdate}(k, \boldsymbol{w}^t)$

  end

  $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t + \sum_{k \in \mathbb{K}_t} \frac{n_k}{n} \Delta \boldsymbol{w}_k^{t+1}$

end

$\texttt{DeviceUpdate}(k, \boldsymbol{w}^t)$:

$\mathcal{B} \leftarrow$ split local data $\mathcal{D}_k^{\text{private}}$ into batches of size B

$\boldsymbol{w} \leftarrow \boldsymbol{w}^t$

for *local epoch* $i \leftarrow 1$ **to** E **do**

  for *batch* $b \in \mathcal{B}$ **do**

    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla L(f_{\boldsymbol{w}}; b)$

  end

end

$\Delta \boldsymbol{w} \leftarrow \boldsymbol{w}^t - \boldsymbol{w}$

return $\Delta \boldsymbol{w}$

Federated averaging algorithm. Given the data $\mathcal{D}_k$ partitioned among devices $k \in \mathbb{K}$, the objective is to learn parameters $\boldsymbol{w}$ of the model $f_{\boldsymbol{w}}$, in the presence of a server S. We use the popular `FederatedAveraging` algorithm (McMahan et al., 2017; McMahan and Ramage, 2017) (Algorithm 1) proposed specifically to perform training on non-IID and imbalanced decentralized data; this has also served as the footing for multiple prior works (Geyer et al., 2017; McMahan et al., 2018; Bonawitz et al., 2017; Smith et al., 2017). The idea here is that training occurs over multiple rounds, where in each round t, a fraction of devices $k \in \mathbb{K}_t$ train models $f_{\boldsymbol{w}}$ using the local data $\mathcal{D}_k^{\text{private}}$ and only communicate incremental model update $\Delta \boldsymbol{w}_k^t$ towards the server's global model $\boldsymbol{w}^t$. The server aggregates (such as by averaging) parameter updates from multiple devices and shares back an updated improved model after each round. Over multiple rounds of communications, the devices converge to model parameters $\boldsymbol{w}^T$ that has been effectively learnt from all the data $\mathcal{D}$, without their raw data ever being communicated to the server or another device. It should be noted that although we consider the simple `FederatedAveraging` algorithm, we expect

our results to generalize to a broad class of decentralized algorithms which involve periodically exchanging model parameter updates.

De-identification in federated learning. A number of precautions are employed to ensure any identifiable information is stripped away from per-device update reports (which includes parameter updates $\Delta \boldsymbol{w}_k^t$ and additional metadata). We first iterate over de-identification strategies employed on-device. The client is initially registered into the FL process by being assigned population identifier (Yang et al., 2018) and thereby bypassing the need to authenticate with a device or user identity (Bonawitz et al., 2019). When possible, PII information is stripped away from the training data (Hard et al., 2018) prior to training on-device. After a number of local training steps, the parameter updates $\Delta \boldsymbol{w}_k^t$ along with anonymized operational metrics (Yang et al., 2018) is transmitted by the device. A (trusted) shuffler (Bittau et al., 2017) can be additionally employed to ensure the transmitted per-device update reports are further sanitized before reaching the server. The shuffler typically strips away a range of user-specific metadata (e.g., IP addresses, routing details) and batches the reports (reordering updates to disassociate timing ordering information). On the whole, multiple mechanism are in-place to ensure that only the essence of the update-reports (i.e., the parameter updates $\Delta \boldsymbol{w}_k^t$) are received by the server to aggregate updates. Consequently, for the rest of the chapter, we assume access to *only* the parameter updates to perform deanonymization.

5.3 DEANONYMIZATION ATTACKS IN FEDERATED LEARNING

In this section, we begin by presenting our threat model to deanonymize devices. We then discuss an insight to why this threat arises and work towards our attack models.

5.3.1 *Threat Model*

To highlight deanonymization risks in Federated Learning (McMahan et al., 2017), we analyze a scenario with K honest users ($K \geq 2$) who collaboratively train an ML model $f_w : \mathcal{X} \to \mathcal{Y}$ over multiple rounds. A server S co-ordinates the training, by periodically collecting model updates from a random subset of users. The model update communicated by each user is a result of performing multiple gradient steps over multiple batches on their local private data (see `DeviceUpdate(.)` in Algo. 1). Furthermore, the model updates are stripped of identifiable metadata (Hard et al., 2018; Bonawitz et al., 2017; Yang et al., 2018) (e.g., device identifiers) and are optionally shuffled (Bittau et al., 2017) to obscure the source of each individual update. Prior to summarizing information from multiple updates, we assume the

server observes only the essence of per-user model update (i.e., parameter updates $\Delta \boldsymbol{w}^t_{\text{anon}}$) to improve $f_{\boldsymbol{w}}$.

We investigate deanonymization through the lens of an honest-but-curious server (the 'adversary') during the training process who uses the model update as an attack surface. The inference-time objective of the adversary is to deanonymize the model update i.e., re-identify the user u who generated $\Delta \boldsymbol{w}^t_{\text{anon}}$. Such a deanonymization objective undermines sanitization mechanisms which de-identify model updates, such as decoupling the update from user identity (Bonawitz et al., 2019), stripping away identifiable metadata (Hard et al., 2018; Yang et al., 2018), and blind-shuffling mechanisms (Bittau et al., 2017). Furthermore, deanonymization also serves as a stepping stone for amplifying information recovered from other inference attacks. For instance, as we show later in §5.5.1.3, deanonymization can be coupled with attribute inference attacks to improve attack performances and further associate recovered attributes with identities.

To deanonymize, the adversary leverages limited prior knowledge of users. Formally, our threat model performs:

$$f^{\text{adv}} : \Delta \boldsymbol{w}^t_{\text{anon}} \times \mathcal{D}^{\text{prior}}_u \rightarrow u \stackrel{?}{=} \text{anon} \tag{5.3}$$

Here, $\Delta \boldsymbol{w}^t_{\text{anon}}$ is the deanonymization target, which is a result of an anonymous user taking multiple gradient steps on her local data $\mathcal{D}^{\text{private}}_{\text{anon}}$. The adversary's auxiliary knowledge of users is denoted by $\{\mathcal{D}^{\text{prior}}_u : u \in \mathbb{U}\}$. We assume $\mathcal{D}^{\text{prior}}_u$ represents a limited set of data generated by user u and is distinct from their private data i.e., $\mathcal{D}^{\text{prior}} \cap \mathcal{D}^{\text{private}}_u = \emptyset \; \forall u \in \mathbb{U}$. For instance, historical data collected by the service, or content publicly shared by the users. In Section 5.4.2, we further elaborate on how we model the adversary's prior knowledge, as it plays a significant role in deanonymization attacks.

5.3.2 *Selection Bias and Biased Estimators*

The core idea of our threat model is to use users' selection bias as an identification cue, which we hypothesize (and shortly verify) is consistent among both the users' prior data (known to adversary) and private device data (unknown to adversary). This implicit user selection biases arise from behavioral factors (Fadem, 2012; Hernán et al., 2004; Berk, 1983) that results in subtle variations of how humans capture data. For instance, Alice's interest in automobiles might result in more variations of cars captured in her text/photos, compared to Bob whose interest lies in sports. At this point, we remark that this results in a non-IID data distribution among data on users and devices, which is well-known in FL literature (McMahan et al., 2017; Bonawitz et al., 2019). However, we do identify and exploit the property that although the data is non-IID among users (large inter-user distances), the data displays lesser variation *within* data generated by the same user (small intra-user distances).

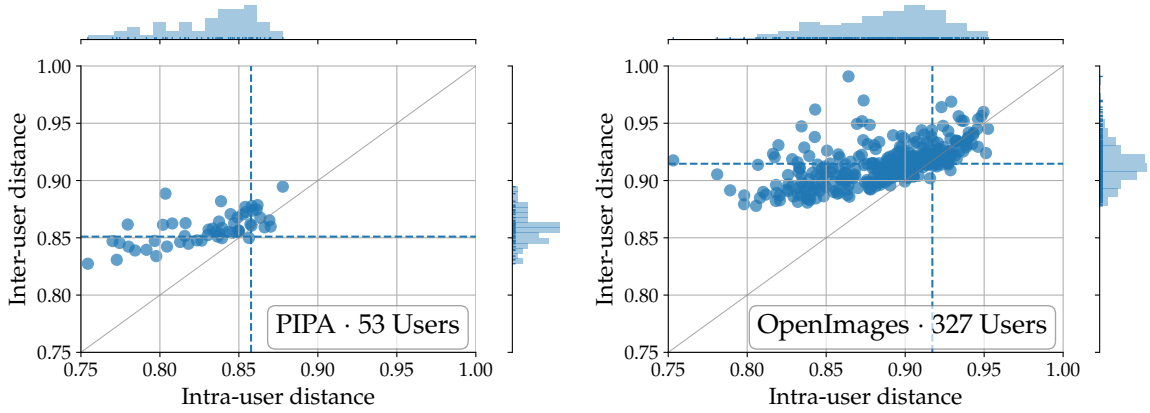


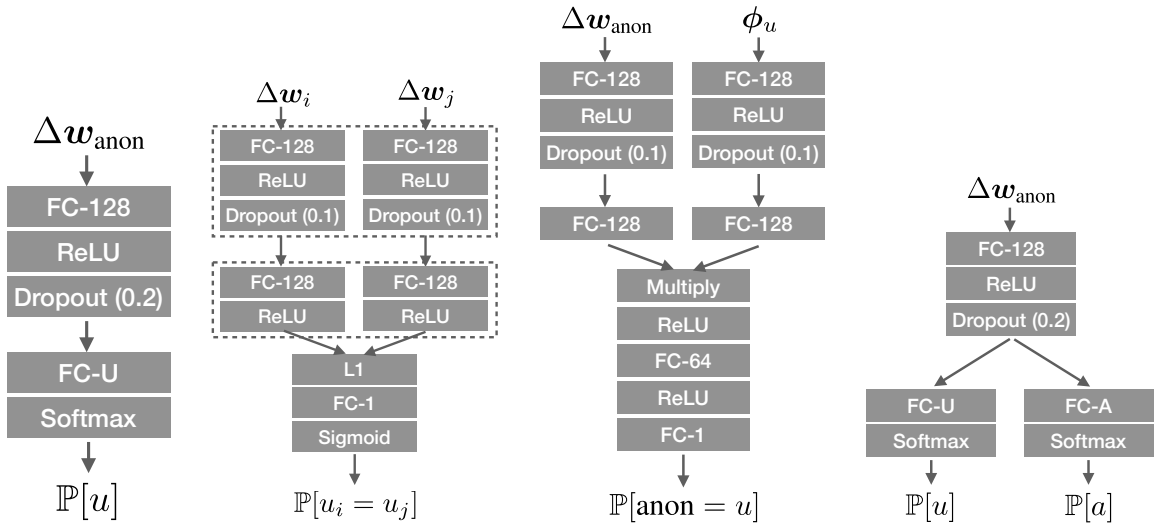
Figure 5.2: Variations in user data. Each point represents distances computed over the image set of a single user.

To validate the assumption, we now present an experiment to quantify user variations on two public image datasets (PIPA (Zhang et al., 2015) and OpenImages (Krasin et al., 2017)). In both cases, we (i) group the images based on the real-world user who captured them using the corresponding `author` fields; and (ii) vectorize images by extracting the 1024-dim `avgpool` features from MobileNet CNN (Howard et al., 2017) and L_2-normalize them. We obtain statistics for each user by computing two L_2 distances: (a) intra-user distance: median image feature distance between images within each user; and (b) inter-user distance: median image distance between user images and a set of random images. We plot these distances per user on a scatter plot in Figure 5.2, each point indicating a distinct user. If images captured by the users were unbiased, we would have found their corresponding points at the intersection of blue dashed lines. However, points predominantly being above the diagonal indicates that examples within each users' collection are similar (low intra-user distances), but are greater (high inter-user distances) when compared to other user collections. In Section 5.5.2.4, we further analyze how similar user-specific variations also arise in the parameter delta space.

The resulting non-IID distribution of user data $\mathcal{D}_u$ among devices leads to each device fitting a *biased* estimator during the `DeviceUpdate` step (Algo. 1) with a bias error: $\text{Bias}[\boldsymbol{w}_u] = \mathbb{E}[\boldsymbol{w}_u] - \boldsymbol{w}^*$, where the expectation term is over the user's training data $\mathcal{D}_u$ and $\boldsymbol{w}^*$ is the optimal estimator. We conjecture (validated in §5.5.2.4) that the bias error signal is consistently encoded in both: (i) the parameter updates transmitted by user's device $\Delta\boldsymbol{w}_u^t$; and (ii) when estimating on prior data of the user $\boldsymbol{w}_u^{\text{prior}} = \text{SGD}(\mathcal{D}_u^{\text{prior}})$. Hence, we reformulate the threat model (Eq. 5.3) in the parameter update space:

$$f^{\text{adv}} : \Delta\boldsymbol{w}_u^{\text{prior}} \times \Delta\boldsymbol{w}_{\text{anon}}^t \rightarrow u \stackrel{?}{=} \text{anon} \tag{5.4}$$

Next, we look at attack models to learn this mapping.



(a) MLP model for re-identification attack (b) Siamese model for matching attack (c) Model for cross-modal matching attack (d) MTL model to predict user u and attribute a

Figure 5.3: Architectures of attack models. Dotted lines indicate shared layers.

5.3.3 *Attacks*

In this section, we present attack models to deanonymize users based on their model updates (Eq. 5.4).

Re-identification attack. In the re-identification scenario, the adversary leverages prior data to learn before-hand (via attack model $f^{\text{re-id}}$) what updates from targeted users look like. The adversary then uses the attack model to re-identify users based on their anonymous update. Formally, the re-identification attack involves training an attack model $f^{\text{re-id}} : \Delta \boldsymbol{w}_u^{\text{prior}} \rightarrow u$ to capture user-specific bias signals in the high-dimensional parameter delta space. At test-time, users are re-identified using their model updates:

$$f^{\text{re-id}} : \Delta \boldsymbol{w}_{\text{anon}} \rightarrow u \tag{5.5}$$

For the re-identification attack model $f^{\text{re-id}}$, we adopt a Multilayer Perceptron (MLP) classifier (architecture in Fig. 5.3a) with a single hidden layer of 128 units and ReLU activation, trained using SGD with learning rate (LR) 0.01, 0.9 momentum and 10^{-6} LR decay.

Matching attack. Instead of learning an update-to-user mapping, the adversary in the matching scenario learns a metric space among model updates. Learning a metric space helps embed model updates close together if they are generated by the same

user, independent of whether the user is a part of the adversary's prior knowledge base. Formally, the adversary's objective is to predict the match probability of a pair of distinct parameter updates:

$$f^{\text{mat}} : (\Delta \boldsymbol{w}_i,\ \Delta \boldsymbol{w}_j) \rightarrow i \stackrel{?}{=} j \tag{5.6}$$

where one or both parameter updates are anonymous. The matching attack is particular helpful in scenarios where the adversary encounters novel users at test-time (§5.5.1.2), or extending to cross-modal situations (discussed in next paragraph). We adopt a Siamese network (Bromley et al., 1994) with metric learning (Weinberger et al., 2006) to perform the matching attack. A Siamese model is characterized by twin networks which accepts distinct inputs ($\Delta \boldsymbol{w}_i$ and $\Delta \boldsymbol{w}_j$ in our case) and is connected by another network to estimate similarity between the individual embeddings produced by the twin networks. In addition, the weights of the twin networks are shared to ensure extremely similar inputs are not mapped to distant embeddings. Our Siamese network (architecture in Fig. 5.3b) is constructed as : (a) two FC-128 layers with ReLU activations which individually encodes $\Delta \boldsymbol{w}_i, \Delta \boldsymbol{w}_j$ into a 128-dim embedding; (b) L_1 distance layer to represent distance between these embeddings; and (c) FC-1 layer with sigmoid activation to predict the match probability. We minimize the binary-cross entropy loss and perform optimization using RMSProp with learning rate 10^{-3}.

Cross-modal matching attack. We extend the matching attack to accommodate the situation where the modality of attacker's prior knowledge (e.g., text) differs from the private data (e.g., visual data) used by the users during training. In such a scenario, parameter updates can no longer be represented in the same space (as in Eq. 5.4,5.6). As a result, the cross-modal matching attack performs:

$$f^{\text{cm-mat}} : (\Delta \boldsymbol{w}_{\text{anon}},\ \boldsymbol{\phi}_u) \rightarrow \text{anon} \stackrel{?}{=} u \tag{5.7}$$

where $\boldsymbol{\phi}_u \in \mathbb{R}^D$ denotes an embedding of the user's prior data $\mathcal{D}_u^{\text{prior}}$. In §5.5.1.1, we discuss exactly how we obtain such an embedding. The attack model (architecture in Fig. 5.3c) to estimate the match probability closely resembles the Siamese network for the matching attack. The only modification is replacing the twin networks with two different networks (each with a single FC-128 layer) to map the inputs into a common 128-dim feature space.

5.4 EXPERIMENTAL SETUP: DATASETS, TASKS, AND MODELS

In this section, we discuss the experimental setup and datasets (summarized in Table 5.1) used to train and evaluate the collaboratively learnt ML model in an FL setup.

Dataset ($\mathcal{D}$)	Task	# Users	N	Input ($\mathcal{X}$)	Output ($\mathcal{Y}$)	Model (f_w)
PIPA	Multi-label class.	53	33K	Image	Labels	`CNN-PIPA-FL`
OpenImages	Multi-label class.	327	317K	Image	Labels	`CNN-OI-FL`
Blog	Language Modeling	55	454K	Text	Text	`NNLM-FL`
Yelp	Sentiment Analysis	118	85.6K	Text	Score	`NNSA-FL`

Table 5.1: Datasets $\mathcal{D}$ and models f_w. List of datasets used along with corresponding statistics, tasks, and models.

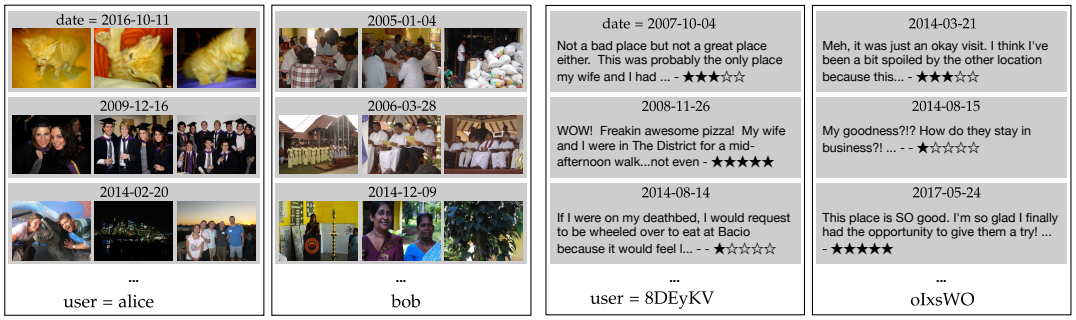


Figure 5.4: Examples of users and corresponding data. OpenImages (top) and Yelp (bottom). Images here are grouped by the anonymized userid and captured/review date. Qualitatively, we observe that the difference between users' data is typically subtle.

5.4.1 *Datasets*

We now present the datasets (Table 5.1, examples in Fig. 5.4) used to train and evaluate the collaboratively trained models f_w. We highlight that the datasets used are well-suited since: (a) they are publicly available; (b) samples are annotated with non-private labels (e.g., tv, flower); (c) examples are complex and realistic; and (d) each training example has a notion of "owner" or "user". Property (d) is particularly important in FL scenarios, as it allows us to partition and distribute data on devices based on user identities. Each of the following paragraphs discusses the (i) dataset $\mathcal{D}$; (ii) corresponding task $\mathcal{X} \to \mathcal{Y}$; and (iii) training model $f_w : \mathcal{X} \to \mathcal{Y}$ to perform the task.

(i) PIPA. PIPA (Zhang et al., 2015) is a dataset consisting of ∼37k personal photos uploaded by actual Flickr users (indicated in the `author` field in Flickr photo metadata). To assure certain minimal amount of per-user data, we only use users with at least 100 images, resulting in 33K images over 53 users. We obtain labels for each image by running a state-of-the-art object detector (Huang et al., 2017b) that detects 80 COCO (Lin et al., 2014) classes, such as umbrella, backpack, and bicycle. To perform reasonable training and evaluation of the multilabel classification task, we use 19

classes (e.g., chair, cup, tv) that occur in approximately >1% of images with high precision. We train a multi-label image classifier `CNN-PIPA-FL` $f_w : \mathbb{R}^{224\times224\times3} \rightarrow \mathbb{R}^{19}$, for this dataset in an FL setup. We use the MobileNet (Howard et al., 2017) architecture designed specifically to be run on mobile devices, as it is a lightweight architecture that strikes a good balance between latency, accuracy and size.

(ii) OpenImages. OpenImages (Krasin et al., 2017) is a large-scale public dataset from Google, consisting of 9M Flickr image URLs and weakly labeled image-level annotations across 19.8k classes. To make training feasible, we prune out users with less than 500 images, resulting in 317k images from 327 users annotated with 18 classes (e.g., food, building). Furthermore, images of the same user can cover a wide time span (typically >5 years). Similar to PIPA, we formulate the training of a multi-label image classifier `CNN-OI-FL` based on the MobileNet architecture.

(iii) Blog Authorship. The Blog Authorship Corpus (Schler et al., 2006) contains ∼681K posts collected from 19K bloggers from `blogger.com`. We work with a subset of 55 users with at least 1000 corresponding posts. Since these blog posts are lengthy (13.5 sentences, 209 words per post), we further split each post into corresponding sentences. As a result, we obtain 454K text sequences over 55 users. We train a language model (`NNLM-FL`): $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-i}, \cdots, \boldsymbol{x}_{t-1};\ \boldsymbol{w})$ i.e., predicting probability distribution of the next word $\boldsymbol{x}_t$ in a sequence given contextual information. Language models trained in an FL architecture are currently deployed to enable smart compose keyboards (Yang et al., 2018). We train a Neural Network Language Model (Bengio et al., 2003) using an embedding layer (with E=100 dims), LSTM layer (Hochreiter and Schmidhuber, 1997) (with L=64 hidden units), and a fully-connected layer (with vocabulary size V=5000).

(iv) Yelp. The Yelp Dataset (Challenge, 2013) contains ∼6M user-reviews of 188K businesses. To allow for each user contributing meaningful parameter deltas, we filter users with at least 500 total reviews. This results in 85K user reviews over 118 users. Each user review contains text (mean length = 180 words) and a 1-5 star rating. We train a sentiment analyzer, modeled as a neural network regressor: $y = f_w([\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots])$, where $y \in [1, 5]$ is the rating and $\boldsymbol{x}_i$ is a representation of i-th word in the review. We use a standard recurrent neural network architecture with an embedding size of E=50, L=128 hidden LSTM units, and a vocabulary size of V=1000.

5.4.2 *Data Setup for Adversarial Knowledge*

The datasets collected (Table 5.1) contain sets of user-specific data $\mathcal{D}_u = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_u}$ over users $u \in \mathbb{U}$. A limited subset of this data is strategically held-out to model the adversary's prior knowledge $\mathcal{D}_u^{\text{prior}}$, and the remaining used as the users' private

training data $\mathcal{D}_u^{\text{private}}$. We consider multiple prior-data limitation strategies to systematically study their influence on deanonymization attacks: (i) limiting the subset of users the adversary has prior knowledge on (§5.4.2.1); and (ii) limiting the amount and quality of prior knowledge (§5.4.2.2).

5.4.2.1 *User Scenarios*

To tackle the case where a subset of participating users in FL may or may not be a part of adversary's prior knowledge database, we set-up two scenarios:

Closed-world. The adversary has some prior information on all users participating anonymously in FL. Consequently, deanonymization of a particular device always maps to a closed-set of 'seen' users. This scenario captures instances of silo-based federated learning scenarios, which typically involve a small number of organizations (the users).

Open-world. We extend the above world to additionally include 'unseen' users during FL, for which the adversary does not have prior information. Hence, a parameter delta $\Delta \boldsymbol{w}_{\text{anon}}$ could map either to a seen or an unseen user. This presents a challenging scenario, as it leads to 'finding a needle in a haystack' i.e, the adversary wants to re-identify a particular target user in spite of background noise generated by many unseen users.

5.4.2.2 *Type of Prior Knowledge*

To understand the role of prior information in a systematic manner, we consider both the *amount* and *distribution* of adversary's prior information w.r.t private data on the FL device. Specifically for the distribution, we model both $\mathcal{D}_u^{\text{prior}}$ and $\mathcal{D}_u^{\text{private}}$ to be sampled (without replacement) from user u's universal data distribution $\mathcal{D}_u$ in one of the four following manners.

(i) `random` prior. Both the prior and private data are IID samples from $\mathcal{D}_u$ i.e., $\mathcal{D}_u^{\text{prior}}, \mathcal{D}_u^{\text{private}} \overset{\text{iid}}{\sim} \mathcal{D}_u$ This scenario captures the adversary scraping information on target user u randomly from various social media sources.

(ii) `chrono` prior. We also consider both prior and private data to be sampled non-IID from $\mathcal{D}_u$ by factoring in timestamps of data (e.g., from image EXIF metadata). Here, data in $\mathcal{D}_u^{\text{prior}}$ chronologically precedes data in $\mathcal{D}_u^{\text{private}}$ For instance, this could occur when an adversary has historical data on the targeted user, such as from a previously de-identified account. In the specific case of the PIPA dataset, where the exact timestamp per example is unavailable, we sample prior and private data non-IID using album information (`photoset` field).

	PIPA			OpenImages	
split	`random`	`chrono`	split	`random`	`chrono`
`CNN-PIPA-FL`	45.1	37.7	`CNN-OI-FL`	62.9	62.2
`CNN-PIPA-SGD`	49.7	40.7	`CNN-OI-SGD`	68.0	67.8
K-NN	14.9	15.8	K-NN	9.7	13.6
Chance	9.5	9.7	Chance	6.3	6.3

	Blog			Yelp	
split	`random`	`chrono`	split	`random`	`chrono`
`NNLM-FL`	28.02	27.83	`NNSA-FL`	0.716	0.708
`NNLM-SGD`	28.62	28.22	`NNSA-SGD`	0.576	0.602
Chance	0.09	0.09	Chance	1.472	1.514

Table 5.2: Evaluation of f_w. Datasets from Table 5.1. Metrics used are: (a) PIPA: Average Precision (AP) (b) OpenImages: Average Precision (AP) (c) Blog: Top5 accuracy (d) Yelp: Mean Absolute Error (MAE). For (a-c), higher is better and for (d), lower is better.

(iii) `profile` prior. We briefly address a scenario where the adversary uses a set of curated 'profile' data as a proxy to users' data. For instance, by curating targeted prior data $\mathcal{D}_u^{\text{prior}}$ to specifically contain weapons to identify participating users who fit that profile.

(iv) `cross-model` prior. We consider the case where adversary's prior data of the user $\mathcal{D}_u^{\text{prior}}$ is gathered from a different modality compared to the private data. For instance, where the prior data is text-based, but the users train on visual data.

5.4.3 *Collaborative Models: Training and Performance*

In Section 5.4.1, we discussed details on the datasets and corresponding model architectures f_w. Section 5.4.2 presented how we strategically hold-out a subset of the data to serve as adversary's prior knowledge. Now we discuss setup and performances of collaborative models in our FL setting.

Training models f_w. For each dataset, we train models f_w using `FederatedAveraging` (Algorithm 1) (McMahan et al., 2017). For all models, crucial hyper-parameters (e.g., size of vocabulary or embedding) were selected carefully after rigorous evaluation over a set of standard choices. In `FederatedAveraging` algorithm, we use C=0.1 and E=1, which we empirically find results in a good trade-off between convergence and communications required. We train the models for 200 epochs with learning rate η=0.01, resulting in 1-4 GPU days to train a single model for a particular architecture,

dataset and scenario. All models are written in Python using the Keras (Chollet et al., 2015) library with a TensorFlow (Abadi et al., 2016a) back-end.

Each user u in our datasets is associated with a variable number of examples $\mathcal{D}_u$ sampled according to some distribution (e.g., `chrono`; see §5.4.2.1). By default, we place half of the users' data $\mathcal{D}_u$ on their anonymous device and reserve the remaining to be used as adversary's prior knowledge. In Section 5.5.2.1, we vary the size of the adversary's prior knowledge and find attacks possible even in severely data-limited settings (e.g., 1-50 prior samples).

Evaluation of f_w. We evaluate performance of the collaboratively-trained models on a 20% held-out test set. For reference, we similarly evaluate models trained in a centralized manner i.e., standard training from a single pool of training data. The performances of FL-trained models (represented as '`X-FL`') and SGD-trained models ('`X-SGD`') are presented in Table 5.2. When possible, we also present the K-Nearest Neighbours (KNN, with K=10) baseline. We observe strong performances of the FL-trained models f_w across all datasets, where they consistency recover $80-98\%$ performance of models trained using centralized SGD.

5.5 EVALUATION

In the previous section, we discussed training ML models in an FL setup for four different datasets covering various tasks such as image classification and language modeling. Within this FL scenario, we now detail the training of *deanonymization attack* models (§5.3.3), evaluate their effectiveness, and work towards understanding how the parameter updates leak user-identifiable information.

Evaluation metrics. We use the following metrics (computed using scikit-learn (Pedregosa et al., 2011)) to evaluate the adversary's attack performance: (i) **Mean Average Precision (AP)**: Adversary's precision-recall curves for held-out user data is computed. We then compute the per-user Average Precision (area under the precision-recall curves). We report the mean of Average Precisions across users in percentages (i.e., AP$\times$100); (ii) **Increase over Chance**: In order to analyze adversary's information gain, we compute this as (predicted AP)/(chance AP). We display this alongside AP scores in the form: $\square\times$; and (iii) **Top-1 accuracy**: We compute the classification success rates over all parameter updates in the test set. These metrics are common among classification tasks e.g., (Everingham et al., 2010b; Lin et al., 2014; Wang et al., 2016) for AP and (Krizhevsky et al., 2012; He et al., 2016a; Deng et al., 2009) for Top-1 accuracy. We use the AP as the primary metric, since it also takes into account ranking among predicted classes.

Training and evaluation data for attacker $f^{\mathbf{adv}}$. We train the ML models (f_w in Table 5.1) in an FL system simultaneously using two disjoint sets of devices per

	PIPA (#Users $U = 53$)					
	`random`			`chrono`		
	AP	Top-1	Top-5	AP	Top-1	Top-5
MLP	91.0 (48×)	84.7	96.3	42.2 (22×)	40.0	68.8
SVM	81.3 (43×)	89.3	91.9	27.7 (15×)	43.7	49.6
kNN	85.4 (45×)	82.6	92.6	31.5 (17×)	38.4	54.8
Chance	1.9 (1×)	2.0	9.9	1.9 (1×)	2.0	9.9

	OpenImages ($U = 327$)					
	`random`			`chrono`		
	AP	Top-1	Top-5	AP	Top-1	Top-5
MLP	53.7 (175×)	51.9	77.9	32.5 (106×)	31.9	57.1
SVM	49.0 (159×)	66.5	67.0	24.6 (80×)	41.7	42.5
kNN	46.0 (150×)	49.2	63.9	25.1 (82×)	30.3	43.1
Chance	0.3 (1×)	0.3	1.5	0.3 (1×)	0.3	1.5

	Blog ($U = 55$)					
	`random`			`chrono`		
	AP	Top-1	Top-5	AP	Top-1	Top-5
MLP	52.9 (29×)	50.1	89.9	44.8 (25×)	47.6	81.3
SVM	35.7 (20×)	46.3	49.2	27.0 (15×)	42.1	46.0
kNN	35.6 (20×)	39.8	64.9	29.5 (16×)	35.6	58.3
Chance	1.8 (1×)	1.7	8.8	1.8 (1×)	1.6	8.8

	Yelp ($U = 118$)					
	`random`			`chrono`		
	AP	Top-1	Top-5	AP	Top-1	Top-5
MLP	23.5 (28×)	25.2	50.1	16.0 (19×)	18.9	38.9
SVM	25.9 (31×)	43.2	44.9	17.1 (20×)	33.3	36.7
kNN	21.6 (25×)	25.3	41.1	15.4 (18×)	21.0	32.9
Chance	0.9 (1×)	0.8	4.1	0.9 (1×)	0.9	4.3

Table 5.3: Re-identification Attack Evaluation ($\Delta \boldsymbol{w}_{\text{anon}} \rightarrow u$). Performed in a closed-world. Chance-level AP $\approx 1/U$.

user: (a) $\mathbb{K}_{\text{anon}}$: anonymous user devices (that adversary wants to deanonymize); and (b) $\mathbb{K}_{\text{prior}}$: adversary's shadow devices containing target users' prior information (that we use to generate training data for attack models in §5.3.3). For simplicity, we restrict each of these sets to contain a single user. During training of f_w over multiple rounds, we accumulate the parameter updates $\Delta \boldsymbol{w}_k^t$ communicated by all devices in FL. To train the attack models f^{adv}, we use the set of parameter updates $\{(\Delta \boldsymbol{w}_k^t, u) : k \in \mathbb{K}_{\text{prior}}\}$, where we know a priori the device k to user u mapping. We discuss in detail training data-limited adversaries in Section 5.5.2.1. We evaluate attacks on the disjoint set of parameter updates $\{\Delta(\boldsymbol{w}_k^t, u) : k \in \mathbb{K}_{\text{anon}}\}$.

Representing $\Delta \boldsymbol{w}_k^t$ for attacks. The parameter updates contain hundred thousands to millions of parameters. To enable faster training and evaluation of attack models, we choose a subset of parameters by representing $\Delta \boldsymbol{w}_k^t$ using weights of layers which achieves best attack performance: (i) `CNN-PIPA-FL, CNN-OI-FL`: Fully Connected Layer (19K parameters); (ii) `NNLM-FL`: LSTM layer (10K parameters); and (iii) `NNSA-FL`: Embedding layer (50K parameters). This has little impact to our attack; influence of each layer is discussed in Section 5.5.2.2. Furthermore, we flatten $\Delta \boldsymbol{w}_k^t$ into a vector and L_2 normalize it.

5.5.1 *Effectiveness of Deanonymization Attacks*

In this section, we validate effectiveness of the deanonymization attacks. We begin by understanding the effectiveness in relation to adversary's prior knowledge (§5.5.1.1 and §5.5.1.2) and discuss how it can be coupled with attribute inference attacks (§5.5.1.3).

5.5.1.1 *Impact of Adversary's Prior Distributions*

In this section, we focus on how *types* of adversary's prior knowledge (§5.4.2.2) influences effectiveness of deanonymization. Consequently, we address a range of scenarios, such as when the adversary has similar (`random`) or historical prior data (`chrono`) of the targeted users to perform deanonymization. We also evaluate the novel challenge where the prior data is from a different modality (`cross-modal`).

Leveraging `random` and `chrono` prior to deanonymize. We present key results of the re-identification attack model 'MLP' (§5.3.3): $f^{\text{re-id}} : \Delta \boldsymbol{w}_{\text{anon}}^t \rightarrow u$ (in a closed-world setting) in Table 5.3. In addition, as baseline attack methods, we also demonstrate performances of 'SVM' (a linear support vector machine) and 'kNN' (a k-nearest neighour classifier using k=10).

From the results presented in Table 5.3, we observe: (i) All deanonymization attacks greatly outperform chance-level performances, with as much as 175× boost for MLP on the OpenImages dataset under the `random` prior, highlighting the effectiveness

of the proposed deanonymization attack; (ii) Even the most simple K-NN attack is reasonably effective and already presents a significant threat (150× over random chance on OpenImages, `random` prior); (iii) MLP is highly effective across all datasets and splits (175× over random chance on OpenImages, `random` prior); (iv) Although the absolute AP scores are lower for the more challenging and larger OpenImages dataset (53.7% AP on `random` prior), the increase over chance level performance is significantly higher (48× on PIPA vs. 175× on OpenImages under the same `random` prior); (v) The attack is effective (19-106×) even on `chrono` priors, where the adversary uses historical prior information to deanonymize users.

The above experiments were performed in a non-IID data-distribution among devices, which is natural in FL since users participate with personal data exhibiting unique biases (§5.3.2). We also perform attack evaluation in a contrasting IID setup, where we manually unbias data on devices by replacing each user example with an example drawn IID from $\mathcal{D} = \bigcup_k \mathcal{D}_k$. We observed near-chance-level adversary performance (e.g., 1.5× chance-level for PIPA) since user data is no longer characteristic. There is strong evidence that anonymous model parameter updates contain ample user information in an FL setup that allows for effective deanonymization.

Cross-modal attacks. We now evaluate the effectiveness of deanonymization attacks with a `cross-modal` prior (Section 5.4.2.2). Here, the adversary is limited to prior knowledge from a *different* modality from the data used during training by the users. In particular, we consider the case where the prior data consists of text samples and the private data consists of images. As we are not aware of any dataset which provides cross-modal user-generated data to evaluate the attack, we substitute PIPA prior image samples with corresponding text-representations obtained using a Neural Image Caption generator (Vinyals et al., 2015). Using this setup, we train the cross-modal matching network $f^{\text{cm-mat}} : (\Delta \boldsymbol{w}_{\text{anon}}, \boldsymbol{\phi}_u) \rightarrow \text{anon} \stackrel{?}{=} u$ (Eq. 5.7). To obtain a compact text representation $\boldsymbol{\phi}_u$ over the prior knowledge (set of text sentences for a particular user), we: (i) obtain the 4096-dim sentence-level embedding using InferSent (Conneau et al., 2017); and (ii) compute the mean over the sentence embeddings for the user. We evaluate $f^{\text{cm-mat}}$ on a balanced set of 10K pairs $\{((\Delta \boldsymbol{w}_{\text{anon}}, \boldsymbol{\phi}_u), \mathbb{1}_{\text{anon}=u})\}$. We observe an attack performance of 76.3 AP (chance = 50.0 AP), indicating that model updates can be interestingly deanonymized even using data from another modality.

Attacking using `profile` prior. In the previous attacks we looked at the task of deanonymizing devices by associating the parameter updates to prior data of users. We now look at a slightly different task of linking devices that fit a certain `profile` prior. We achieve this by manually constructing $\mathcal{D}^{\text{profile}}$ to comprise of examples of interest e.g., weapons. In Figure 5.5 we display the top users (in the OpenImages dataset) found using the re-identification attack who fit the corresponding profiles. We observe: (i) devices can be remarkably singled out using various proxy

Figure 5.5: `profile` prior. Devices can be isolated using proxy distributions of certain profiles e.g., guitars. Rows denote private data $\mathcal{D}_u^{\text{private}}$ of users on devices.

distributions (of e.g., handgun, guitar) circumventing the need for real user data; (ii) however, valid correlations in data can sometimes lead to false positives. For instance, 'dumbbells' which often co-occur in images along with other physical equipment devices leads to bicycle images of user 128 (which also displays similar correlations) being falsely identified.

5.5.1.2 *Impact of Number of Seen and Unseen Users*

In the previous section, we evaluated attacks in a closed-world scenario (§5.4.2.1), where the adversary was aware of every users' existence (i.e., included in prior knowledge). We now consider the open-world scenario, where at test-time the adversary additionally encounters model updates generated by *unseen* users (i.e., not in the prior knowledge). This introduces the challenge of differentiating between seen and unseen identities when deanonymizing.

User split. In our experimental setup, we split the users $\mathbb{U}$ into three variably-sized disjoint sets: (a) $\mathbb{U}_{\text{unseen}}$: prior data is unavailable and should be classified as unseen at test-time; (b) $\mathbb{U}_{\text{seen}}$: prior data is available and should be deanonymized at test-time; and (c) $\mathbb{U}_{\text{holdout}}$: these users are reserved purely for training purposes.

Re-identification setup. Previously in the closed-world scenario, we trained the MLP (§5.3.3) classifier $f^{\text{re-id}} : \Delta \boldsymbol{w}_k \rightarrow u$ with $|\mathbb{U}|$ classes representing all users at test time. Now we train a similar classifier over $|\mathbb{U}_{\text{seen}}| + 1$ output classes with the additional class `unseen` collectively denoting unseen users. During training, we use users $\mathbb{U}_{\text{holdout}}$ and their parameter updates to train the `unseen` class.

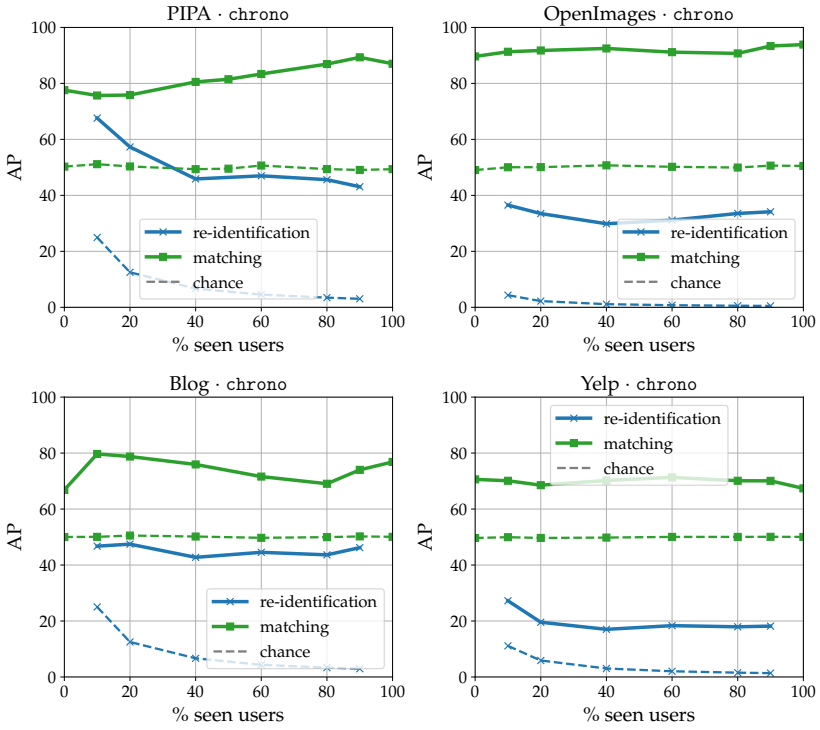


Figure 5.6: Open-world evaluation. Across re-identification (MLP) and matching (Siamese) attack models.

Matching setup. We train a Siamese network (§5.3.3) using parameter updates from held-out and seen set of users. Given a pair $(\Delta \boldsymbol{w}_i, \Delta \boldsymbol{w}_j)$, the network predicts the probability $\mathbb{P}[i = j]$ of being generated by the same user.

Evaluation. The performances are evaluated at different ratios of seen and unseen users at test time. We keep the size of the hold-out set constant to one-third of the total number of users. Evaluation for both re-identification and matching tasks on the challenging `chrono` prior distributions per dataset are presented in Figure 5.6. We observe: (i) even in the open-world scenario, we perform much higher than chance-level for both the tasks consistently across a wide range of seen vs. unseen scenarios; (ii) for the re-identification attack, as % seen users increase, the complexity of the task increases as well (due to larger output-space). Hence, we notice a drop in AP performance (67%→43% in PIPA). However, performance compared to chance-level significantly increases (3× →14×); (iii) in the matching task, the Siamese model performs much higher than chance-level even in a purely open-world setting, with no seen users (1.5× for PIPA and 1.8× for OpenImages). We find both the re-identification and matching attacks generalize well in the presence of unseen users at test time.

		STL		MTL	
Attributes	# Attrs	AttrInf	Deanon	AttrInf	Deanon
Age	5	89.1	-	90.8	90.9
Gender	2	93.1	-	94.4	91.6
Glasses	3	98.5	-	98.9	91.3
Hair Color	3	85.2	-	88.7	90.1
Hair Length	5	91.3	-	91.3	90.1
-	-	-	87.6	-	-

Table 5.4: Attribute inference and deanonymization attack performances. Results are reported in top-1 accuracies. Columns indicate when the inference tasks are trained individually (STL) and jointly (MTL).

5.5.1.3 *Amplification with Attribute Inference Attacks*

We now discuss how deanonymization attacks can be coupled with related inference attacks on model updates. Specifically, we consider the recent attribute inference attack (Melis et al., 2019), which recovers sensitive properties (e.g., race) that holds for subsets of training data. In this particular case, our attack objective involves jointly inferring both identity (via our deanonymization attacks) and sensitive attributes (via attribute inference attacks) via transmitted model updates.

To evaluate the attacks, we closely follow the data setup on Melis et al. (Melis et al., 2019) on the PIPA dataset. Attribute inference in this setting involves inferring sensitive attributes (e.g., age) from the model updates. To this end, we first train individual attribute classification models for each of the five attributes, and an additional re-identification model. All the classification models are MLPs following the architecture of the re-identification model. Table 5.4 (column STL) presents results over the five attribute inference (column AttrInf) tasks and deanonymization (column Deanon). Here, we observe that an attacker can consistently achieve 85.2-98.5% accuracy in inferring various attributes from model updates and 87.6% accuracy in inferring identities of participants. These results suggest that model updates indeed leak details unrelated to the trained task (recognizing chair, couch, etc.) and allows an attacker to recover sensitive attributes of the users' training data (via attribute inference) and further link them to an identity (via deanonymization).

We now recast the problem of inference on attributes and identities as a multi-task learning (MTL) (Caruana, 1997) problem. The core idea is to exploit commonalities between the two related tasks to learn a better representation jointly benefiting the tasks. To achieve this, we extend our re-identification model (§5.3.3, which performs user classification) with a secondary classification head (which performs attribute inference; see Fig. 5.3d). Consequently, the model is simultaneously trained for both attribute inference and deanonymization using their corresponding losses. The

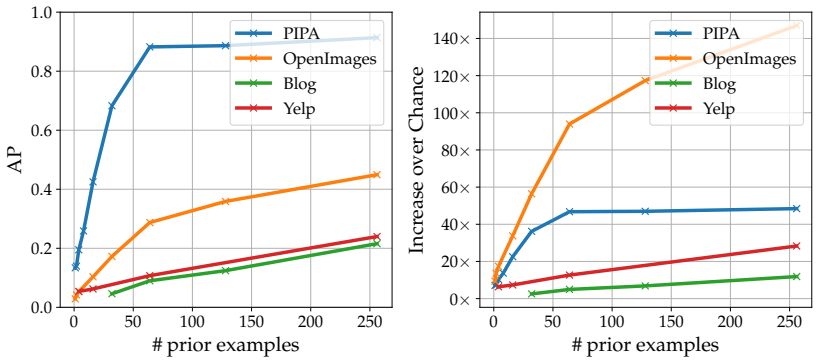


Figure 5.7: Number of prior examples per user. Evaluated on closed-world re-identification.

results for the model is presented under the MTL column in Table 5.4. We observe by learning the two tasks jointly can improve attribute inference performances consistently by 0-3.5% and deanonymization by 2.5-4%. Our results suggest that apart from jointly inferring sensitive attributes and recovering identities, the two related attacks surprisingly amplify each other's performances.

5.5.2 *Analysis*

In this section, we take a closer look at various factors that influence (e.g., amount of training data) the effectiveness of attacks. For simplicity, we study the factors using the re-identification attack in a closed-world setup. We conclude the section by reasoning why model updates lend themselves to deanonymization risks.

5.5.2.1 *Amount of Training Data*

We study the influence of data-limitation in deanonymization attacks in a closed-world re-identification scenario. We previously used the entire reserve set of prior information to perform the deanonymization attacks. We first address the influence in the amount of this prior information available per target user. From Figure 5.7, we observe: (i) even a single prior example of the user leads to non-chance-level re-identification, with as much as 13.4% AP (7 $\times$) performance on PIPA; (ii) performance of the attack increases significantly with the size of prior knowledge across all datasets e.g., 67% increase in performance on OpenImages by using 16$\rightarrow$32 prior examples; (iii) some tasks require more prior information than others. For instance, although Blog and PIPA contain similar number of users, an adversary requires approximately 5$\times$ as many prior Blog examples to achieve 20% AP. We attribute this to a weaker signal generated from sparse text content in Blog, as compared to dense pixel content in PIPA.

We also address the impact of size of training set ($\{\Delta \boldsymbol{w}_k^t : k \in \mathbb{K}_{\text{anon}}\}$) for attack models. We train multiple re-identification MLP adversary models, each trained on

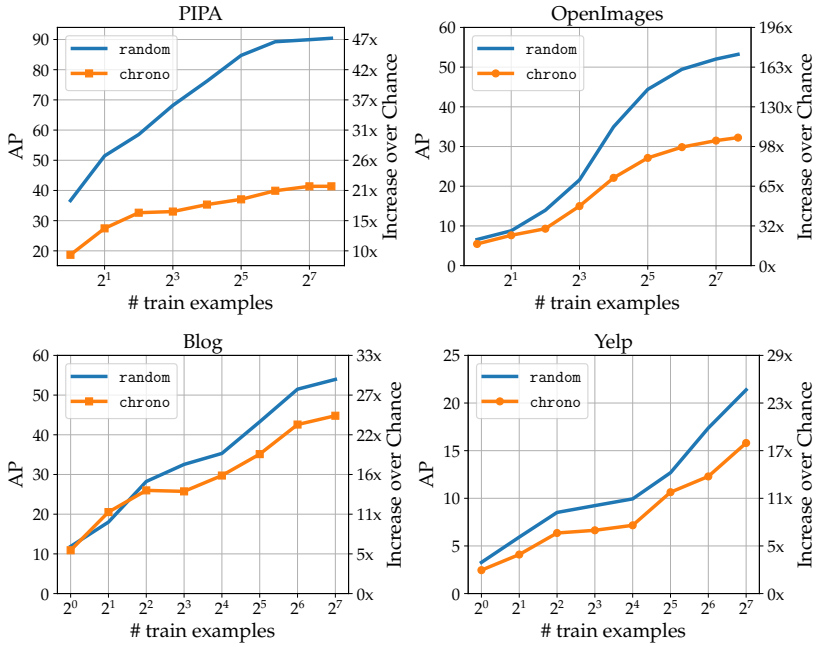


Figure 5.8: Number of training examples per user. Evaluated on closed-world re-identification.

a random subset of training data with increasing sizes. In Figure 5.8, we observe an adversary can train reasonably effective attack models, even with extremely limited labeled data. In particular, attack performances of 3-22× can be obtained with a single labeled example per user. While the amount of data (either training or prior) does strongly influence the attack performance, we nonetheless find deanonymization is possible in strongly data-limited situations.

5.5.2.2 *Impact of Parameter Layers*

The deanonymization targets (i.e., model updates Δw) comprise of parameters from multiple layers of a deep neural network. We now analyze how the layer type and depth affect attacker performance, since they influence the type of task-specific information learnt by the model. For instance, in CNNs, layers at various depths of the network are known to learn various concepts (Zeiler and Fergus, 2014) – lower level features (e.g., corners, edges) in the initial layers and higher level features (e.g., wheel, bird's feet) in the final layers. For parameters updates contributed by each individual layer, we train a total of 27 attack models for CNN-based models and 3 attack models for LSTM-based models. We were limited by storage capacity to evaluate on OpenImages as it would require $>$ 3TB.

From layer-wise performances in Figure 5.9 and Table 5.5, we observe: (i) *all* layers provide above-chance level information to perform re-identification attacks; (ii) in the CNN model, higher level layers contain more identifiable information with the final fully connected (FC) layer being the most informative; (iii) in the RNN-based

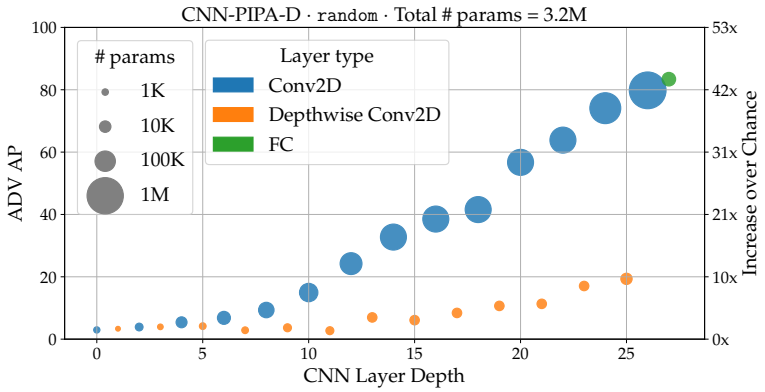


Figure 5.9: Re-identification performance by depth. Bubble sizes indicate the number of parameters in each layer. Last two layers contains 1M and 19K parameters respectively.

		NNLM-D (92K)		NNSM-D (141K)	
Depth	Layer type	AP	# params	AP	# params
1	Embedding	15.7 (9×)	50K	23.5 (28×)	50K
2	LSTM	46.0 (25×)	10K	19.2 (23×)	91K
3	FC	38.8 (21×)	32K	17.6 (21×)	128

Table 5.5: Re-identification performance by depth. For models trained on Blog and Yelp.

models, the LSTM parameters are more informative for language modeling, whereas it is the embedding layer for sentiment analysis.

5.5.2.3 *Impact of Optimization State*

We now analyze the influence of training progress of the ML model on deanonymization attacks. We group the parameter updates (separately for train and test attack sets), based on the epoch ranges during which they were generated. We split parameter updates collected during training of f_w over 200 epochs into 10 ranges, each with 20 epochs. We train and evaluate the MLP re-identification attack model over all 10×10 train-eval pairs. From Figure 5.10, we observe that the training progress at which the update was generated has little influence on the performance indicating an adversary can re-identify users at any stage of training.

5.5.2.4 *Reasoning About Effectiveness of Attacks*

In Section 5.3.2 (Fig. 5.2), we observed that users display a bias resulting in lower variations in data they capture. Consequently, we conjectured that the resulting bias is consistently encoded in the parameter updates, even when they are computed

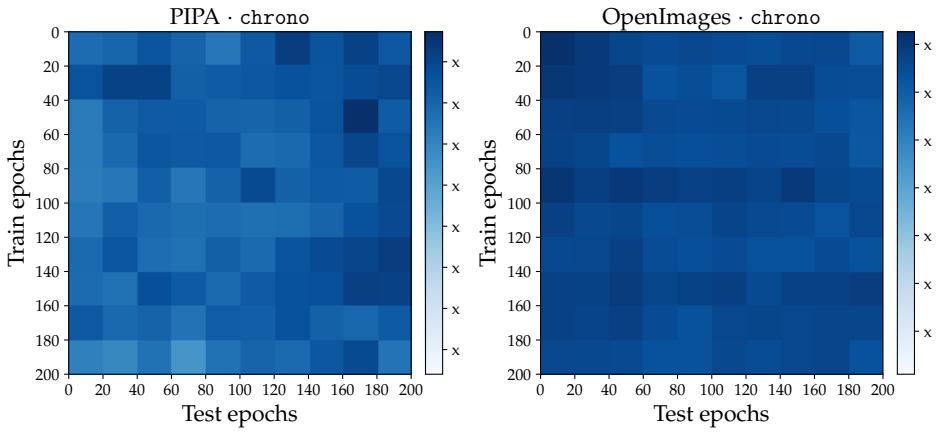


Figure 5.10: Effect of the epoch t. On the re-identification attack $\Delta \boldsymbol{w}^{t}_{\text{anon}} \rightarrow u$. As an example, the top-right cell denotes when the MLP was trained on $\Delta \boldsymbol{w}^{t}_{u}, t \in [0, 20]$ and evaluated on $\Delta \boldsymbol{w}^{t'}_{\text{anon}}, t' \in [180, 200]$

on different (prior and private) sets of users' data. To validate, we take a closer look at the parameter updates $\Delta \boldsymbol{w}^{\text{prior}}_{u}, \Delta \boldsymbol{w}^{\text{private}}_{u} \in \mathbb{R}^{D \times K}$ in the FC layer of eight users in the PIPA FL setup, where K (=19) is the number of classes and D(=1024) represents weights per class. In Figure 5.11, we illustrate bias per user (columns) in the parameter delta space by computing the L_2-norm of each of the K class weight vectors (column-dimension in $\Delta \boldsymbol{w}_u$). We observe: (i) for users who can be re-identified highly accurately (e.g., u=10), we find that the user is more biased towards images containing 'tie', 'tv', and 'laptop'. Furthermore, this bias is consistent in both the user's prior and private update signals; and (ii) surprisingly, even when biases are not entirely consistent (e.g., u=17), we find attacks to be reasonable effective (AP=95); and (iii) for users who cannot be re-identified easily (e.g., u=13), the biases are inconsistent between the prior (biased towards cars and cups) and private (biased towards chairs, ties, and umbrellas) update signals. We find our conjecture that the user bias signal translates to the parameter delta space, holds reasonably well, leading to highly effective deanonymization attacks we saw in the previous sections.

5.6 COUNTERMEASURES

In the previous section, we evaluated our threat models across a variety of challenging scenarios and consistently observed deanonymization risks. In this section, we present mitigation strategies to counter these attacks.

We attributed (§5.5.2.4) the effectiveness of the attacks to user bias, which is a powerful statistical signal in both the limited set of adversary's prior data and the users' private data. The focus of our mitigation strategies is to perturb the data bias on the anonymous device, to provide a false signal to the adversary. We spell out our requirements for the defense as: (a) maximally retain utility (performance of

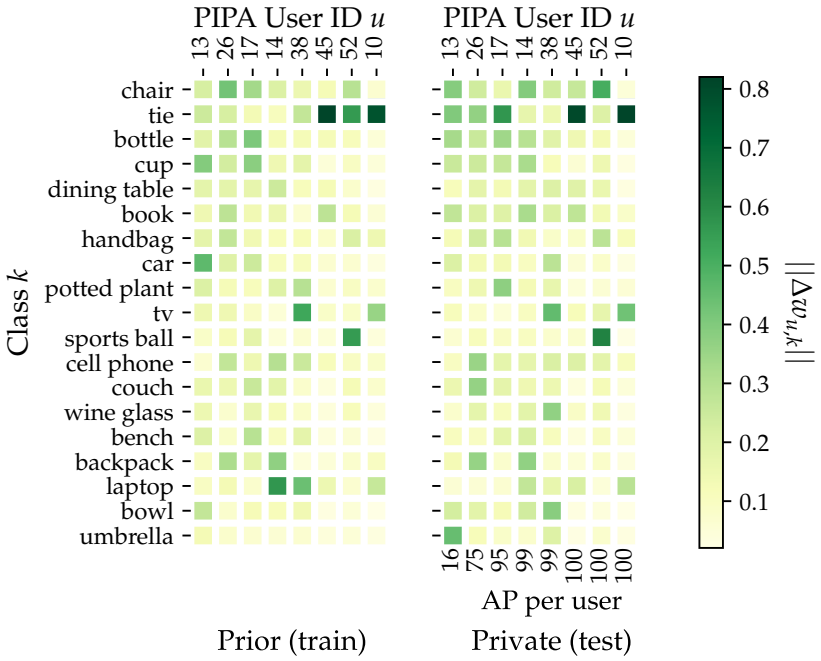


Figure 5.11: User bias visualized on parameter updates.

$\mathcal{D}$	Source ($\mathcal{D}$)	$\mathcal{D}^{\text{bkg}}$	Source ($\mathcal{D}^{\text{bkg}}$)	$\lvert\mathcal{D}^{\text{bkg}}\rvert$
PIPA (Zhang et al., 2015)	Flickr	OpenImages	Flickr	59K
OpenImages (Krasin et al., 2017)	Flickr	OpenImages	Flickr	490K
Blog (Schler et al., 2006)	Blogger	WikiReading (Hewlett et al., 2016)	Wikipedia	3M
Yelp (Challenge, 2013)	Yelp	Amazon Reviews (He and McAuley, 2016)	Amazon	1.7M

Table 5.6: Background datasets and sources. Used to mitigate deanonymization attacks.

f_w); (b) involve low computation overhead; (c) not rely on a trusted-third party; and (d) allow users to selectively employ the strategy to various extents depending on personal preferences.

5.6.1 *Methods*

Based on the requirements, we propose data-centric mitigation strategies: devices adversarially bias their data distribution on devices, rather than directly perturb model parameters. More specifically, users mix their original data $\mathcal{D}_u$ with certain "background" data $\mathcal{D}^{\text{bkg}}$ to "blend into the crowd", thereby rendering the parameters less user-specific. Here, the mixing takes place prior to participation in FL.

Collecting $\mathcal{D}^{\text{bkg}}$. The background dataset $\mathcal{D}^{\text{bkg}}$ can be any large (labeled) set of training examples for the same federated learning task (e.g. user-annotated dataset, scraped data from the Internet, a trusted open-source dataset). The background

datasets used in our experiments, their sources and sizes are listed in Table 5.6. We only select a random subset of the original background datasets (s.t. $|\mathcal{D}^{\text{bkg}}| \gg |\mathcal{D}_u|$) in each case, for experiments to complete within a feasible amount of time. The preprocessing of $\mathcal{D}^{\text{bkg}}$ and $\mathcal{D}$ are identical.

Now, we present three countermeasures which alter the characteristic data-distribution of the users.

Data replacement (`bkg-repl`). Each user replaces a fraction $\alpha \in [0,1]$ of his/her data $\mathcal{D}_u$ with ones from $\mathcal{D}^{\text{bkg}}$. At $\alpha = 0$, no mitigation strategy takes place; at $\alpha = 1$, every user has identical data composition. However, the strategy skews FL to learn from a noisy background data distribution displaying different statistics, instead of learning from interesting user data on which evaluation metrics need to be maximized.

Data augmentation (`rand-aug`). Instead of replacing, the user *augments* random data (since more data helps (Sun et al., 2017a; Halevy et al., 2009)) from $\mathcal{D}^{\text{bkg}}$:

$$\hat{\mathcal{D}_u} \leftarrow \mathcal{D}_u \cup \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \sim \mathcal{D}^{\text{bkg}}\}_{i=1}^{\alpha \cdot |\mathcal{D}_u|}, \tag{5.8}$$

where $\alpha \geq 0$ determines the size of augmentation. As $\alpha \to \infty$, devices' empirical data distributions converge to $\mathcal{D}^{\text{bkg}}$, making them indistinguishable from each other.

Mode-specific data augmentation (`mm-aug`). So far, the users' strategies were to mix their data with background data from a single source $\mathcal{D}^{\text{bkg}}$. We now consider the strategy where each device mixes data from *different* topics i.e., modes of the data distribution. For instance, Alice adversarially adds sports content to her data to mask her interest in automobiles before participating in FL. We perform this by first clustering $\mathcal{D}^{\text{bkg}}$ into M clusters $\bigcup_{m=1}^{M} \mathcal{D}_m^{\text{bkg}}$. We use the k-means clustering over the ImageNet pretrained Mobilenet features. Each user u picks a cluster m at random, and augments its data with ones from the cluster:

$$\hat{\mathcal{D}_u} \leftarrow \mathcal{D}_u \cup \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \sim \mathcal{D}_m^{\text{bkg}}\}_{i=1}^{\alpha \cdot |\mathcal{D}_u|} \tag{5.9}$$

where $\alpha \geq 0$ controls the degree of mix. We use M=100 for PIPA, M=500 for OpenImages, M=300 for Blog and Yelp.

We additionally consider two perturbation-based baselines to our data-augmentation strategies.

DP-FederatedAveraging (`dp-fedavg`). We implement a differentially private variant (McMahan et al., 2018) of the Federated Averaging algorithm. They key idea is to provide (ε, δ) participant-level differential privacy guarantees by bounding the contribution (the parameter update) provided by each participant. In practise, the

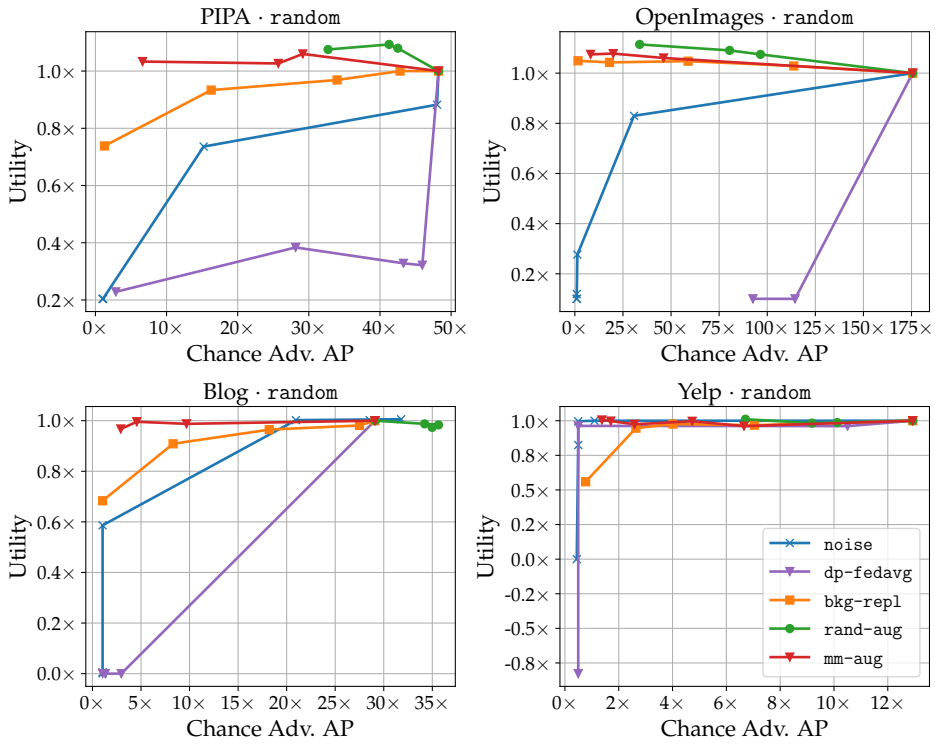


Figure 5.12: Mitigation strategies evaluation. Re-identification AP obtained by varying α and σ^2 in closed-world scenario. Top-left is the ideal region. Higher α and σ^2 values pushes operating points towards the left (i.e., lower deanonymization performance).

contributions are bounded by clipping the parameter updates and further adding random noise. In our experiments, we fix the clipping value to 50 and vary magnitude of gaussian noise added during training.

Random perturbations (`noise`). Although `dp-fedavg` has shown success in large-scale scenarios (with thousands of users), we found difficulty achieving reasonable results in our setup. Hence, we consider a relaxed version of introducing perturbations, where the user introduces zero-centered Gaussian noise to model updates before leaving the device.

5.6.2 *Evaluation*

We evaluate the proposed mitigation strategies by measuring the adversary's performance against our countermeasures. We analyze the effectiveness of the defense against the strongest adversary: closed-world re-identification attack on `random` prior (§5.5.1.1, Table 5.3).

We evaluate the strategies in terms of trade-off between privacy (reduction in adversary's performance) and utility (decentralized learning performance). As in

§5.5.1.1, we measure the adversary's performance as increase over chance-level AP. We measure utility by performance scores normalized to have utility=1.0 when no mitigation takes place.

The mitigation strategies are evaluated on a curve by varying hyperparameters. For `bkg-repl`, we use $\alpha \in \{0.0, 0.25, 0.50, 0.75, 1.0\}$. For `rand-aug` and `mm-aug`, we use $\alpha \in \{0.0, 0.5, 1.0, 2.0\}$. For `dp-fedavg`, we fix the clip value to 50 and vary the noise multiplier in the range $[10^{-5}, 10^{-1}]$. For `noise`, we consider Gaussian noise with $\mu = 0$ and $\sigma^2 \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$.

We present evaluation for our strategies in Figure 5.12. Better mitigation strategies have curves towards the top-left corners in each plot (high privacy, high utility). We observe: (i) the perturbation-based baselines (`noise` and `dp-fedavg`) in most cases severely decreases utility at a small gain in privacy; (ii) replacing data with background samples (`bkg-repl`) is a good alternative strategy: we have both higher privacy and utility than perturbation methods. However, due to a domain-shift between $\mathcal{D}^{\text{bkg}}$ and $\mathcal{D}$, utility is often impacted. This can be observed in PIPA, Blog and Yelp datasets, where it achieves $< 0.75\times$ utility since the user data is no longer used; (iii) the augmentation-based strategies `rand-aug` and `mm-aug` outperforms `noise` and `bkg-repl` in terms of utility and privacy; (iv) for the `mm-aug` strategy, already at $\alpha = 0.5$, we observe a good combination of privacy and utility (75% decrease in adversary's AP in OpenImages, compared to 45% for `rand-aug` and 67% for `bkg-replace`).

We find the strategy `mm-aug` offer the most effective and practical operating points, requiring the user to perform minimal augmentation to achieve reasonable privacy. We remark that the utility for `mm-aug` can be more than 1.0 even at higher privacy level, as can be seen in PIPA and OpenImages. This is due to the effect of additional data (Halevy et al., 2009; Sun et al., 2017a). This increased privacy and utility comes at the cost of preparing a labeled dataset and increased training time (training set becomes $(1+\alpha)\times$ bulky). However, this overhead will be less costly with increasingly powerful devices and energy-efficient ML models for mobile devices (Howard et al., 2017; Sandler et al., 2018).

5.7 CONCLUSION

In this chapter, we were motivated to understand privacy threats in Federated Learning, which is designed towards large-scale learning on user data on personal devices. We questioned whether devices can truly participate anonymously without compromising the identity of individuals. Our results indicate that the devices can be effectively deanonymized using the transmitted model parameter updates and a reasonable amount of prior data. We found this to be possible due to the inherent user bias in captured data acting as a fingerprint that is consistent across different sets of data captured by the user. To mitigate such attacks, we proposed cali-

brated domain-specific data augmentation, which shows strong results in preventing deanonymization with minimal impact to utility.

Part III

LEAKAGE DURING INFERENCE

Having discussed leakage of information in raw data (i.e., visual content) and during training (i.e., in model parameters), we now switch focus to inference time. Understanding leakage at inference time is particularly important as models are being increasingly deployed in many real-world environments (e.g., internet APIs, on edge devices). In this part, we specifically address leakage of model functionality.

In Chapter 6, we begin by presenting model functionality stealing attacks that pose a confidentiality threat to the model owner's intellectual property. While literature has been successful at executing these attacks on simple models (e.g., shallow neural networks), our approach highlights the threat on complex models (e.g., ResNets), despite making weaker assumptions. Our approach leverages advances in knowledge transfer and reinforcement learning to demonstrate successful model functionality stealing attacks.

In Chapter 7, we work towards the first effective defense to counteract threats posed by recent model stealing attacks, including our attack presented in Chapter 6. The key idea to our approach is to treat defense as optimization problem, where the perturbation is optimized to target the attacker's gradient signal during learning. Consequently, we present the first active defense that actively perturbs the predictions returned by the model owner (i.e., the defender).

6

KNOCKOFF NETS

Machine Learning (ML) models are increasingly deployed in the wild to perform a wide range of tasks. In this chapter, we ask to what extent can an adversary steal functionality of such "victim" models based solely on blackbox interactions: image in, predictions out. In contrast to prior work, we study complex victim blackbox models, and an adversary lacking knowledge of train/test data used by the model, its internals, and semantics over model outputs. We formulate model functionality stealing as a two-step approach: (i) querying a set of input images to the blackbox model to obtain predictions; and (ii) training a "knockoff" with queried image-prediction pairs. We make multiple remarkable observations: (a) querying random images from a different distribution than that of the blackbox training data results in a well-performing knockoff; (b) this is possible even when the knockoff is represented using a different architecture; and (c) our reinforcement learning approach additionally improves query sample efficiency in certain settings and provides performance gains. We validate model functionality stealing on a range of datasets and tasks, as well as show that a reasonable knockoff of an image analysis API could be created for as little as $30.

The content of this chapter is based on Orekondy et al. (2019b). As a first author, Tribhuvanesh Orekondy conducted all the experiments and was the main writer for the conference paper.

6.1 INTRODUCTION

Machine Learning (ML) models and especially deep neural networks are deployed to improve productivity or experience e.g., photo assistants in smartphones, image recognition APIs in cloud-based internet services, and for navigation and control in autonomous vehicles. Developing and engineering such models for commercial use is a product of intense time, money, and human effort – ranging from collecting a massive annotated dataset to tuning the right model for the task. The details of the dataset, exact model architecture, and hyperparameters are naturally kept confidential to protect the models' value. However, in order to be monetized or simply serve a purpose, they are deployed in various applications (e.g., home assistants) to function as blackboxes: input in, predictions out.

Large-scale deployments of deep learning models in the wild has motivated the community to ask: can someone abuse the model solely based on blackbox access? There has been a series of "inference attacks" (Shokri et al., 2017; Oh et al., 2018;

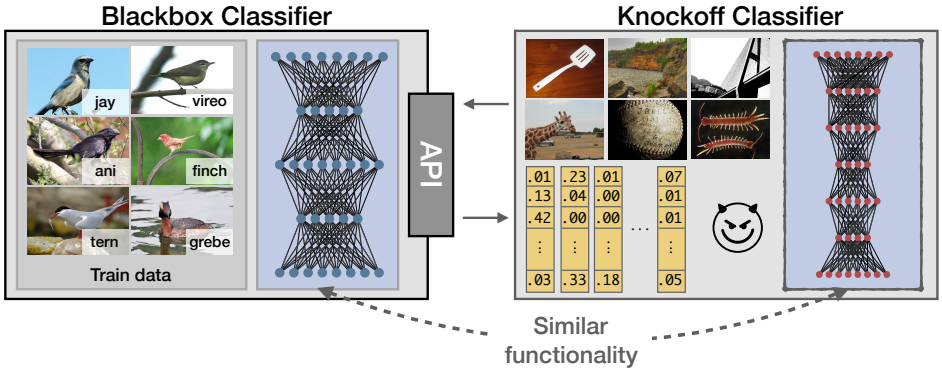


Figure 6.1: An adversary can create a "knockoff" of a blackbox model solely by interacting with its API: image in, prediction out. The knockoff bypasses the monetary costs and intellectual effort involved in creating the blackbox model.

Fredrikson et al., 2015; Salem et al., 2019) which try to infer properties (e.g., training data (Shokri et al., 2017), architecture (Oh et al., 2018)) about the model within the blackbox. In this chapter, we focus on model functionality stealing: can one create a "knockoff" of the blackbox model solely based on observed input-output pairs? In contrast to prior work (Lowd and Meek, 2005a; Tramèr et al., 2016; Papernot et al., 2017b; Juuti et al., 2019), we work towards purely stealing *functionality* of complex blackbox models by making fewer assumptions.

We formulate model functionality stealing as follows (shown in Figure 6.1). The adversary interacts with a blackbox "victim" CNN by providing it input images and obtaining respective predictions. The resulting image-prediction pairs are used to train a "knockoff" model. The adversary's intention is for the knockoff to compete with the victim model at the victim's task. Note that knowledge transfer (Hinton et al., 2015; Buciluǎ et al., 2006) approaches are a special case within our formulation, where the task, train/test data, and white-box teacher (victim) model are known to the adversary.

Within this formulation, we spell out questions answered in our chapter with an end-goal of model functionality stealing:

1. Can we train a knockoff on a random set of query images and corresponding blackbox predictions?
2. What makes for a good set of images to query?
3. How can we improve sample efficiency of queries?
4. What makes for a good knockoff architecture?

6.2 PROBLEM STATEMENT

We now formalize the task of functionality stealing (see also Figure 6.2).

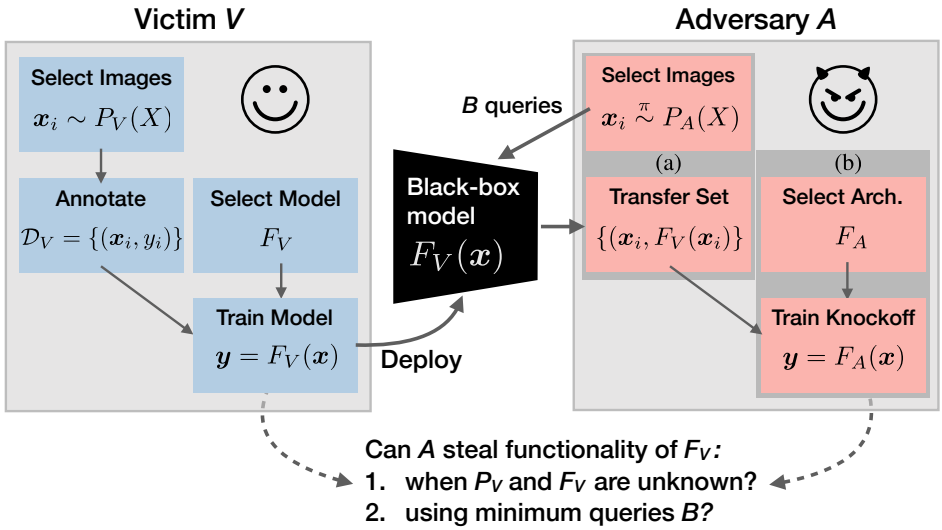


Figure 6.2: Problem Statement. Laying out the task of model functionality stealing in the view of two players - victim V and adversary A. We group adversary's moves into (a) Transfer Set Construction (b) Training Knockoff F_A.

Functionality stealing. In this chapter, we introduce the task as: given blackbox query access to a "victim" model $F_V : \mathcal{X} \rightarrow \mathcal{Y}$, to replicate its functionality using "knockoff" model F_A of the adversary. As shown in Figure 6.2, we set it up as a two-player game between a victim V and an adversary A. Now, we discuss the assumptions in which the players operate and their corresponding moves in this game.

Victim's move. The victim's end-goal is to deploy a trained CNN model F_V in the wild for a particular task (e.g., fine-grained bird classification). To train this particular model, the victim: (i) collects task-specific images $\boldsymbol{x} \sim P_V(X)$ and obtains expert annotations resulting in a dataset $\mathcal{D}_V = \{(\boldsymbol{x}_i, y_i)\}$; (ii) selects the model F_V that achieves best performance (accuracy) on a held-out test set of images $\mathcal{D}_V^{\text{test}}$. The resulting model is deployed as a blackbox which predicts output probabilities $\boldsymbol{y} = F_V(\boldsymbol{x})$ given an image $\boldsymbol{x}$. Furthermore, we assume each prediction incurs a cost (e.g., monetary, latency).

Adversary's unknowns. The adversary is presented with a blackbox CNN image classifier, which given *any* image $\boldsymbol{x} \in \mathcal{X}$ returns a K-dim posterior probability vector $\boldsymbol{y} \in [0,1]^K$, $\sum_k y_k = 1$. We relax this later by considering truncated versions of $\boldsymbol{y}$. We assume remaining aspects to be unknown: (i) the internals of F_V e.g., hyperparameters or architecture; (ii) the data used to train and evaluate the model; and (iii) semantics over the K classes.

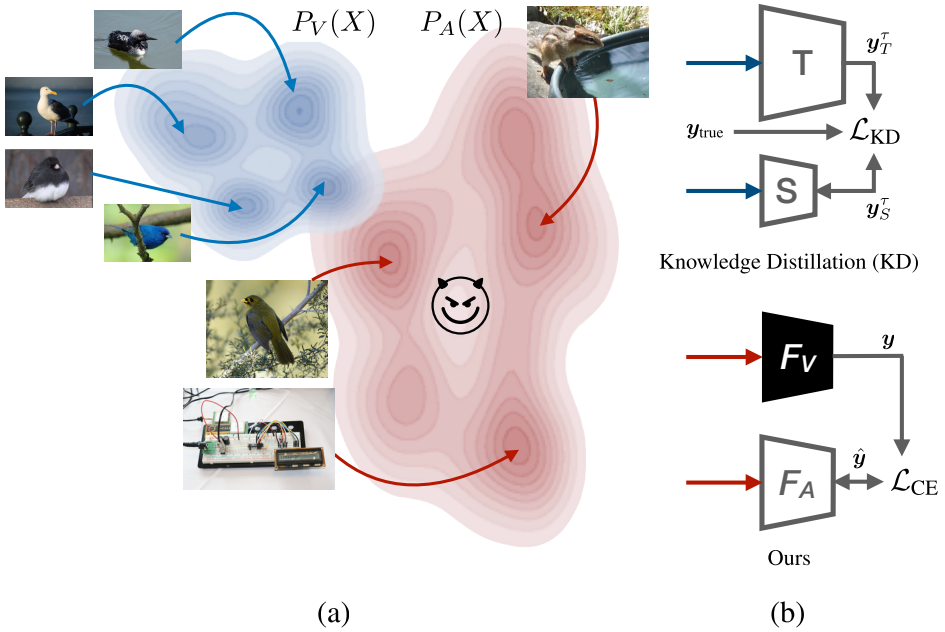


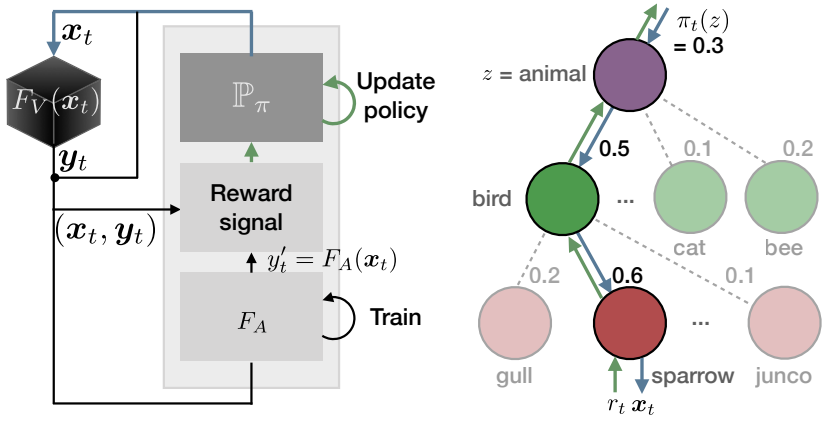
Figure 6.3: Comparison to KD. (a) Adversary has access only to image distribution $P_A(X)$ (b) Training in a KD-manner requires stronger knowledge of the victim. Both S and F_A are trained to classify images $\boldsymbol{x} \in P_V(X)$

Adversary's attack. To train a knockoff, the adversary: (i) interactively queries images $\{\boldsymbol{x}_i \overset{\pi}{\sim} P_A(X)\}$ using strategy π to obtain a "transfer set" of images and pseudo-labels $\{(\boldsymbol{x}_i, F_V(\boldsymbol{x}_i))\}_{i=1}^B$; and (ii) selects an architecture F_A for the knockoff and trains it to mimic the behaviour of F_V on the transfer set.

Objective. We focus on the adversary, whose primary objective is training a knockoff that performs well on the task for which F_V was designed i.e., on an unknown $\mathcal{D}_V^{\text{test}}$. In addition, we address two secondary objectives: (i) sample-efficiency: maximizing performance within a budget of B blackbox queries; and (ii) understanding what makes for good images to query the blackbox.

Victim's defense. Although we primarily address the adversary's strategy in the chapter, we briefly discuss victim's counter strategies (in Section 6.5) of reducing informativeness of predictions by truncation e.g., rounding-off.

Remarks: Comparison to knowledge distillation (KD). Training the knockoff model is reminiscent of KD approaches (Hinton et al., 2015; Romero et al., 2015), whose goal is to transfer the knowledge from a larger teacher network T (white-box) to a compact student network S (knockoff) via the transfer set. We illustrate key differences between KD and our setting in Figure 6.3: (a) **Independent distribution** P_A: F_A is trained on images $\boldsymbol{x} \sim P_A(X)$ *independent* to distribution P_V used for



(a) Overview (b) Hierarchical policy $\mathbb{P}_\pi$

Figure 6.4: Strategy `adaptive`.

training F_V; (b) **Data for supervision**: Student network S minimize variants of KD loss:

$$\mathcal{L}_{\text{KD}} = \lambda_1 \mathcal{L}_{\text{CE}}(\boldsymbol{y}_{\text{true}},\ \boldsymbol{y}_S) + \lambda_2 \mathcal{L}_{\text{CE}}(\boldsymbol{y}_S^\tau,\ \boldsymbol{y}_T^\tau) \tag{6.1}$$

where $\boldsymbol{y}_T^\tau = \text{softmax}(\boldsymbol{a}_T/\tau)$ is the softened posterior distribution of logits $\boldsymbol{a}$ controlled by temperature τ. In contrast, the knockoff (student) in our case lacks logits $\boldsymbol{a}_T$ and true labels $\boldsymbol{y}_{\text{true}}$ to supervise training.

6.3 GENERATING KNOCKOFFS

In this section, we elaborate on the adversary's approach in two steps: transfer set construction (Section 6.3.1) and training knockoff F_A (Section 6.3.2).

6.3.1 *Transfer Set Construction*

The goal is to obtain a transfer set i.e., image-prediction pairs, on which the knockoff will be trained to imitate the victim's blackbox model F_V.

Selecting $P_A(X)$. The adversary first selects an image distribution to sample images. We consider this to be a large discrete set of images. For instance, one of the distributions P_A we consider is the 1.2M images of ILSVRC dataset (Deng et al., 2009).

Sampling strategy π. Once the image distribution $P_A(X)$ is chosen, the adversary samples images $\boldsymbol{x} \overset{\pi}{\sim} P_A(X)$ using a strategy π. We consider two strategies.

6.3.1.1 *Random Strategy*

In this strategy, we randomly sample images (without replacement) $\boldsymbol{x} \overset{\text{iid}}{\sim} P_A(X)$ to query F_V. This is an extreme case where adversary performs pure exploration. However, there is a risk that the adversary samples images irrelevant to learning the task (e.g., over-querying dog images to a birds classifier).

6.3.1.2 *Adaptive Strategy*

We now incorporate a feedback signal resulting from each image queried to the blackbox. A policy π to adaptively sample images ($\boldsymbol{x}_t \sim \mathbb{P}_\pi(\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{t-1})$) is learnt to achieve two goals: (i) improving sample-efficiency of queries; and (ii) aiding interpretability of blackbox F_V. The approach is outlined in Figure 6.4a. At each time-step t, the policy module $\mathbb{P}_\pi$ samples a set of query images. A reward signal r_t is shaped based on multiple criteria and is used to update the policy with an end-goal of maximizing the expected reward.

Supplementing P_A. To encourage relevant queries, we enrich images in the adversary's distribution by associating each image $\boldsymbol{x}_i$ with a label $z_i \in Z$. No semantic relation of these labels with the blackbox's output classes is assumed or exploited. As an example, when P_A corresponds to 1.2M images of the ILSVRC (Deng et al., 2009) dataset, we use labels defined over 1000 classes. These labels can be alternatively obtained by unsupervised measures e.g., clustering or estimating graph-density (Ebert et al., 2012; Beluch et al., 2018). We find using labels aids understanding blackbox functionality. Furthermore, since we expect labels $\{z_i \in Z\}$ to be correlated or inter-dependent, we represent them within a coarse-to-fine hierarchy, as nodes of a tree as shown in Figure 6.4b.

Actions. At each time-step t, we sample actions from a discrete action space $z_t \in Z$ i.e., adversary's independent label space. Drawing an action is a forward-pass (denoted by a blue line in Figure 6.4b) through the tree: at each node, we sample a child node with probability $\pi_t(z)$ (which sums to 1 over siblings). The probabilities are determined by a softmax distribution over the node potentials: $\pi_t(z) = \frac{e^{H_t(z)}}{\sum_{z'} e^{H_t(z')}}$. Upon reaching a leaf-node, a sample of images is returned corresponding to label z_t.

Learning the policy. We use the received reward r_t for an action z_t to update the policy π using the gradient bandit algorithm (Sutton and Barto, 1998). This update is equivalent to a backward-pass through the tree (denoted by a green line in Figure 6.4b), where the node potentials are updated as:

$$H_{t+1}(z_t) = H_t(z_t) + \alpha(r_t - \bar{r}_t)(1 - \pi_t(z_t)) \quad \text{and} \tag{6.2}$$

$$H_{t+1}(z') = H_t(z') + \alpha(r_t - \bar{r}_t)\pi_t(z') \qquad \forall z' \neq z_t \tag{6.3}$$

where $\alpha = 1/N(z)$ is the learning rate, $N(z)$ is the number of times action z has been drawn, and $\bar{r}_t$ is the mean-reward over past Δ time-steps. $\pi_0(z)$ and $H_0(z)$ are intialized such that reaching all leaf nodes in the hierarchy are equally probable.

Rewards. To evaluate the quality of sampled images $\boldsymbol{x}_t$, we study three rewards. We use a margin-based **certainty** measure (Settles and Craven, 2008; Joshi et al., 2009) to encourage images where the victim is confident (hence indicating the domain F_V was trained on):

$$R^{\text{cert}}(\boldsymbol{y}_t) = P(\boldsymbol{y}_{t,k_1}|\boldsymbol{x}_t) - P(\boldsymbol{y}_{t,k_2}|\boldsymbol{x}_t) \tag{6.4}$$

where k_i is the ith-most confident class. To prevent the degenerate case of image exploitation over a single label, we introduce a **diversity** reward:

$$R^{\text{div}}(\boldsymbol{y}_{1:t}) = \sum_k \max(0, y_{t,k} - \bar{y}_{t:t-\Delta,k}) \tag{6.5}$$

To encourage images where the knockoff prediction $\hat{\boldsymbol{y}}_t = F_A(\boldsymbol{x}_t)$ does not imitate F_V, we reward high CE **loss**:

$$R^{\mathcal{L}}(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) = \mathcal{L}(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) \tag{6.6}$$

We sum up individual rewards when multiple measures are used. To maintain an equal weighting, each reward is individually rescaled to [0, 1] and subtracted with a baseline computed over past Δ time-steps.

6.3.2 *Training Knockoff F_A*

As a product of the previous step of interactively querying the blackbox model, we have a transfer set $\{(\boldsymbol{x}_t, F_V(\boldsymbol{x}_t)\}_{t=1}^B$, $\boldsymbol{x}_t \overset{\pi}{\sim} P_A(X)$. Now we address how this is used to train a knockoff F_A.

Selecting architecture F_A. Few works (Oh et al., 2018; Wang and Gong, 2018) have recently explored reverse-engineering the blackbox i.e., identifying the architecture, hyperparameters, etc. We however argue this is orthogonal to our requirement of simply stealing the functionality. Instead, we represent F_A with a reasonably complex architecture e.g., VGG (Simonyan and Zisserman, 2014) or ResNet (He et al., 2016a). Existing findings in KD (Hinton et al., 2015; Furlanello et al., 2018) and model compression (Buciluǎ et al., 2006; Han et al., 2016a; Iandola et al., 2016) indicate robustness to choice of reasonably complex student models. We investigate the choice under weaker knowledge of the teacher (F_V) e.g., training data and architecture is unknown.

Blackbox (F_V)	$\lvert\mathcal{D}_V^{\text{train}}\rvert + \lvert\mathcal{D}_V^{\text{test}}\rvert$	Output classes K
`Caltech256` (Griffin et al., 2007)	23.3k + 6.4k	256 general object categories
`CUBS200` (Wah et al., 2011)	6k + 5.8k	200 bird species
`Indoor67` (Quattoni and Torralba, 2009)	14.3k + 1.3k	67 indoor scenes
`Diabetic5` (*Eyepacs*)	34.1k + 1k	5 diabetic retinopathy scales

Table 6.1: Four victim blackboxes F_V. Each blackbox is named in the format: [dataset][# output classes].

Training to imitate. To bootstrap learning, we begin with a pretrained Imagenet network F_A (see § D.1 in supplementary for discussion on other initializations). We train the knockoff F_A to imitate F_V on the transfer set by minimizing the cross-entropy (CE) loss: $\mathcal{L}_{\text{CE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_k p(y_k) \cdot \log p(\hat{y}_k)$. This is a standard CE loss, albeit weighed with the confidence $p(y_k)$ of the victim's label.

6.4 EXPERIMENTAL SETUP

We now discuss the experimental setup of multiple victim blackboxes (Section 6.4.1), followed by details on the adversary's approach (Section 6.4.2).

6.4.1 *Black-box Victim Models F_V*

We choose four diverse image classification CNNs, addressing multiple challenges in image classification e.g., fine-grained recognition. Each CNN performs a task specific to a dataset. A summary of the blackboxes is presented in Table 6.1 (extended descriptions in appendix).

Training the black-boxes. All models are trained using a ResNet-34 architecture (with ImageNet (Deng et al., 2009) pretrained weights) on the training split of the respective datasets. We find this architecture choice achieve strong performance on all datasets at a reasonable computational cost. Models are trained using SGD with momentum (of 0.5) optimizer for 200 epochs with a base learning rate of 0.1 decayed by a factor of 0.1 every 60 epochs. We follow the train-test splits suggested by the respective authors for **Caltech-256** (Griffin et al., 2007), **CUBS-200-2011** (Wah et al., 2011), and **Indoor-Scenes** (Quattoni and Torralba, 2009). Since GT annotations for **Diabetic-Retinopathy** (*Eyepacs*) test images are not provided, we reserve 200 training images for each of the five classes for testing. The number of test images per class for all datasets are roughly balanced. The test images of these datasets $\mathcal{D}_V^{\text{test}}$ are used to evaluate both the victim and knockoff models.

After these four victim models are trained, we use them as a blackbox for the remainder of the chapter: images in, posterior probabilities out.

6.4.2 *Representing P_A*

In this section, we elaborate on the setup of two aspects relevant to transfer set construction (Section 6.3.1).

6.4.2.1 *Choice of P_A*

Our approach for transfer set construction involves the adversary querying images from a large discrete image distribution P_A. In this section, we present four choices considered in our experiments. Any information apart from the images from the respective datasets are unused in the `random` strategy. For the `adaptive` strategy, we use image-level labels (chosen independent of blackbox models) to guide sampling.

$P_A = P_V$. For reference, we sample from the exact set of images used to train the blackboxes. This is a special case of knowledge-distillation (Hinton et al., 2015) with unlabeled data at temperature $\tau = 1$.

$P_A =$ ILSVRC (Russakovsky et al., 2015; Deng et al., 2009). We use the collection of 1.2M images over 1000 categories presented in the ILSVRC-2012 (Russakovsky et al., 2015) challenge.

$P_A =$ OpenImages (Kuznetsova et al., 2018). OpenImages v4 is a large-scale dataset of 9.2M images gathered from Flickr. We use a subset of 550K unique images, gathered by sampling 2k images from each of 600 categories.

$P_A = D^2$. We construct a dataset wherein the adversary has access to all images in the universe. In our case, we create the dataset by pooling training data from: (i) all four datasets listed in Section 6.4.1; and (ii) both datasets presented in this section. This results in a "dataset of datasets" D^2 of 2.2M images and 2129 classes.

Overlap between P_A and P_V. We compute overlap between labels of the blackbox (K, e.g., 256 Caltech classes) and the adversary's dataset (Z, e.g., 1k ILSVRC classes) as: $100 \times |K \cap Z|/|K|$. Based on the overlap between the two image distributions, we categorize P_A as:

1. **$P_A = P_V$**: Images queried are identical to the ones used for training F_V. There is a 100% overlap.
2. **Closed-world ($P_A = D^2$)**: Blackbox train data P_V is a subset of the image universe P_A. There is a 100% overlap.
3. **Open-world ($P_A \in$ {ILSVRC, OpenImages})**: Any overlap between P_V and P_A is purely coincidental. Overlaps are: `Caltech256` (42% ILSVRC, 44% OpenImages), `CUBS200` (1%, 0.5%), `Indoor67` (15%, 6%), and `Diabetic5` (0%, 0%).

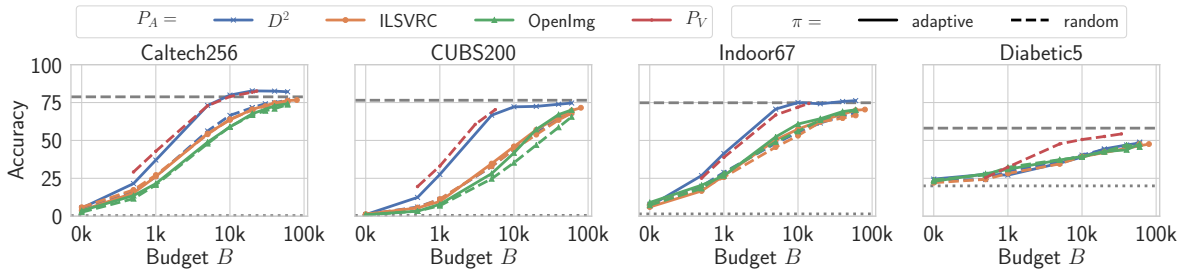


Figure 6.5: Performance of the knockoff at various budgets. Across choices of adversary's image distribution (P_A) and sampling strategy π. - represents accuracy of blackbox F_V and ····· represents chance-level performance. Enlarged version available in supplementary.

6.4.2.2 *Adaptive Strategy*

In the adaptive strategy (Section 6.3.1.2), we make use of auxiliary information (labels) in the adversary's data P_A to guide the construction of the transfer set. We represent these labels as the leaf nodes in the coarse-to-fine concept hierarchy tree. The root node in all cases is a single concept "entity". We obtain the rest of the hierarchy as follows: (i) D^2: we add as parents the dataset the images belong to; (ii) ILSVRC: for each of the 1K labels, we obtain 30 coarse labels by clustering the mean visual features of each label obtained using 2048-dim pool features of an ILSVRC pretrained Resnet model; (iii) OpenImages: We use the exact hierarchy provided by the authors.

6.5 RESULTS

We now discuss the experimental results.

Training phases. The knockoff models are trained in two phases: (a) *Online*: during transfer set construction (Section 6.3.1); followed by (b) *Offline*: the model is retrained using transfer set obtained thus far (Section 6.3.2). All results on knockoff are reported after step (b).

Evaluation metrics. We evaluate two aspects of the knockoff: (a) *Top-1 accuracy*: computed on victim's held-out test data $\mathcal{D}_V^{\text{test}}$ (b) *sample-efficiency*: best performance achieved after a budget of B queries. Accuracy is reported in two forms: absolute (x%) or relative to blackbox F_V ($x\times$).

In each of the following experiments, we evaluate our approach with identical hyperparameters across all blackboxes, highlighting the generalizability of model functionality stealing.

		random			
	P_A	Caltech256	CUBS200	Indoor67	Diabetic5
	$P_V(F_V)$	78.8 (1×)	76.5 (1×)	74.9 (1×)	58.1 (1×)
	P_V (KD)	82.6 (1.05×)	70.3 (0.92×)	74.4 (0.99×)	54.3 (0.93×)
Closed	D^2	76.6 (0.97×)	68.3 (0.89×)	68.3 (0.91×)	48.9 (0.84×)
Open	ILSVRC	75.4 (0.96×)	68.0 (0.89×)	66.5 (0.89×)	47.7 (0.82×)
	OpenImg	73.6 (0.93×)	65.6 (0.86×)	69.9 (0.93×)	47.0 (0.81×)

		adaptive			
	P_A	Caltech256	CUBS200	Indoor67	Diabetic5
	$P_V(F_V)$	-	-	-	-
	P_V (KD)	-	-	-	-
Closed	D^2	82.7 (1.05×)	74.7 (0.98×)	76.3 (1.02×)	48.3 (0.83×)
Open	ILSVRC	76.2 (0.97×)	69.7 (0.91×)	69.9 (0.93×)	44.6 (0.77×)
	OpenImg	74.2 (0.94×)	70.1 (0.92×)	70.2 (0.94×)	47.7 (0.82×)

Table 6.2: Accuracy on test sets. Accuracy of blackbox F_V indicated in gray and knockoffs F_A in black. KD = Knowledge Distillation. Closed- and open-world accuracies reported at B=60k.

6.5.1 *Transfer Set Construction*

In this section, we analyze influence of transfer set $\{(\boldsymbol{x}_i, F_V(\boldsymbol{x}_i)\}$ on the knockoff. For simplicity, for the remainder of this section we fix the architecture of the victim and knockoff to a Resnet-34 (He et al., 2016a).

Reference: $P_A = P_V$ (KD). From Table 6.2 (second row), we observe: (i) all knockoff models recover 0.92-1.05× performance of F_V; (ii) a better performance than F_V itself (e.g., 3.8% improvement on Caltech256) due to regularizing effect of training on soft-labels (Hinton et al., 2015).

Can we learn by querying *randomly* from an independent distribution? Unlike KD, the knockoff is now trained and evaluated on different image distributions (P_A and P_V respectively). We first focus on the random strategy, which does not use any auxiliary information.

We make the following observations from Table 6.2 (random): (i) **closed-world**: the knockoff is able to reasonably imitate all the blackbox models, recovering 0.84-0.97× blackbox performance; (ii) **open-world**: in this challenging scenario, the knockoff

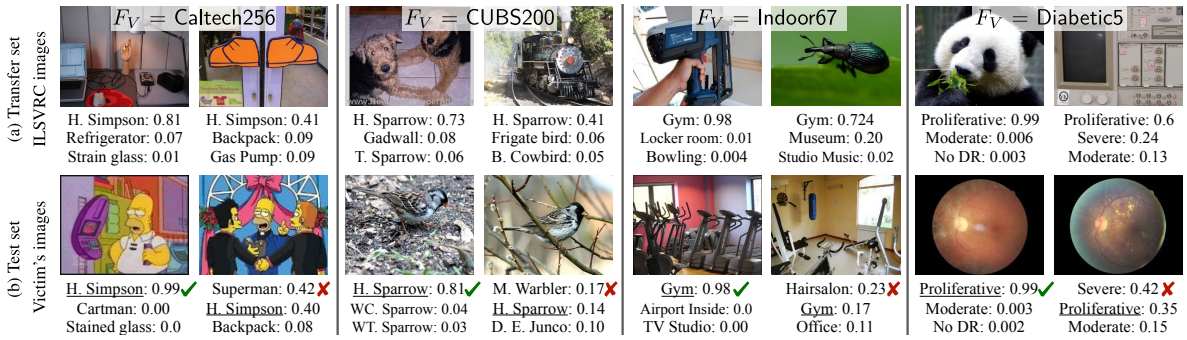


Figure 6.6: Qualitative results. (a) Samples from the transfer set ($\{(x_i, F_V(x_i))\}, x_i \sim P_A(X)$) displayed for four output classes (one from each blackbox): 'Homer Simpson', 'Harris Sparrow', 'Gym', and 'Proliferative DR'. (b) With the knockoff F_A trained on the transfer set, we visualize its predictions on victim's test set ($\{(x_i, F_A(x_i))\}, x_i \sim \mathcal{D}_V^{\text{test}}$). Ground truth labels are underlined. Objects from these classes, among numerous others, were never encountered while training F_A.

model has *never* encountered images of numerous classes at test-time e.g., >90% of the bird classes in `CUBS200`. Yet remarkably, the knockoff is able to obtain 0.81-0.96× performance of the blackbox. Moreover, results marginally vary (at most 0.04×) between ILSVRC and OpenImages, indicating any large diverse set of images makes for a good transfer set.

Upon qualitative analysis, we find the image and pseudo-label pairs in the transfer set are semantically incoherent (Fig. 6.6a) for output classes non-existent in training images P_A. However, when relevant images are presented at test-time (Fig. 6.6b), the adversary displays strong performance. Furthermore, we find the top predictions by knockoff relevant to the image e.g., predicting one comic character (superman) for another.

How *sample-efficient* can we get? Now we evaluate the `adaptive` strategy (discussed in Section 6.3.1.2). Note that we make use of auxiliary information of the images in these tasks (labels of images in P_A). We use the reward set which obtained the best performance in each scenario: {certainty} (Eq. 6.4) in closed-world and {certainty, diversity, loss} (Eq. 6.4-6.6) in open-world.

From Figure 6.5, we observe: (i) **closed-world**: `adaptive` is extremely sample-efficient in all but one case. Moreover, we also find the label hierarchy result in better performance (see supp. §D.3). Its performance is comparable to KD in spite of samples drawn from a 36-188× larger image distribution. We find significant sample-efficiency improvements e.g., while `CUBS200-random` reaches 68.3% at B=60k, `adaptive` achieves this 6× quicker at B=10k. We find comparably low performance in `Diabetic5` as the blackbox exhibits confident predictions for all images resulting in poor feedback signal to guide policy; (ii) **open-world**: although we find marginal improvements over `random` in this challenging scenario, they are pronounced in few

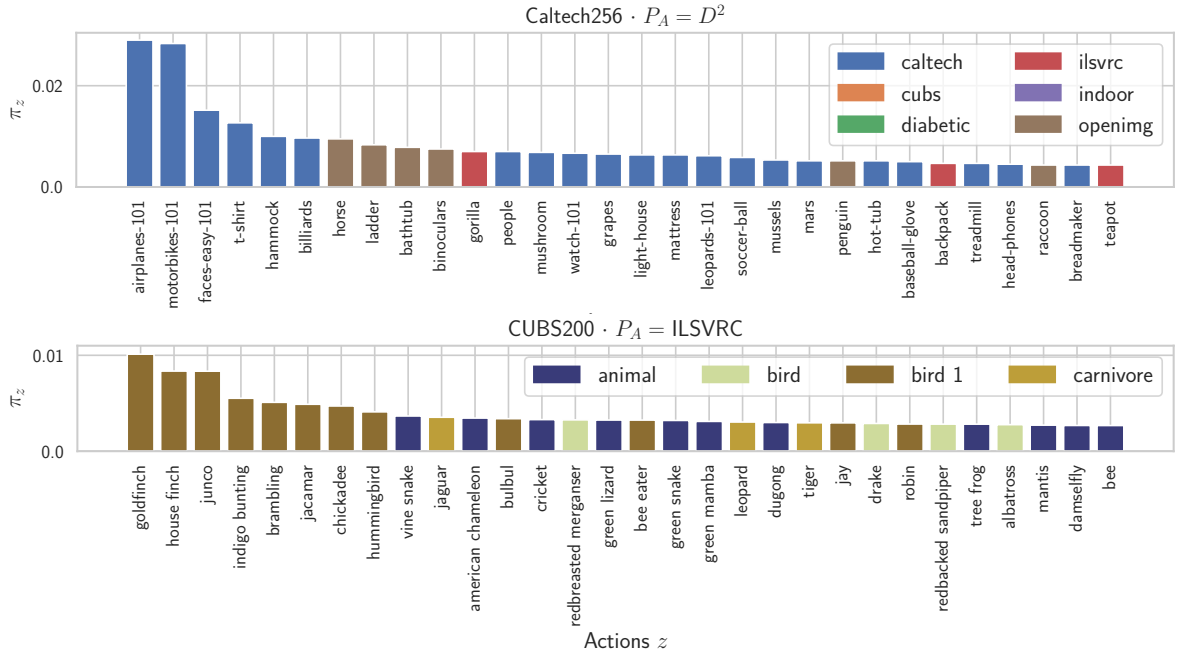


Figure 6.7: Policy π learnt by the `adaptive` approach. Each bar represents preference for action z. Top 30 actions (out of 2.1k and 1k) are displayed. Colors indicate parent of action in hierarchy.

cases e.g., 1.5× quicker to reach an accuracy 57% on `CUBS200` with OpenImages. (iii) as an added-benefit apart from sample-efficiency, from Table 6.2, we find `adaptive` display improved performance (up to 4.5%) consistently across all choices of F_V.

What can we learn by inspecting the *policy*? From previous experiments, we observed two benefits of the `adaptive` strategy: sample-efficiency (although more prominent in the closed-world) and improved performance. The policy π_t learnt by `adaptive` (Section 6.3.1.2) additionally allows us to understand what makes for good images to query. $\pi_t(z)$ is a discrete probability distribution indicating preference over action z. Each action z in our case corresponds to labels in the adversary's image distribution.

We visualize $\pi_t(z)$ in Figure 6.7, where each bar represents an action and its color, the parent in the hierarchy. We observe: (i) **closed-world** (Fig. 6.7 top): actions sampled with higher probabilities consistently correspond to output classes of F_V. Upon analyzing parents of these actions (the dataset source), the policy also learns to sample images for the output classes from an alternative richer image source e.g., "ladder" images in `Caltech256` sampled from OpenImages instead; (ii) **open-world** (Fig. 6.7 bottom): unlike closed-world, the optimal mapping between adversary's actions to blackbox's output classes is non-trivial and unclear. However, we find top actions typically correspond to output classes of F_V e.g., indigo bunting. The

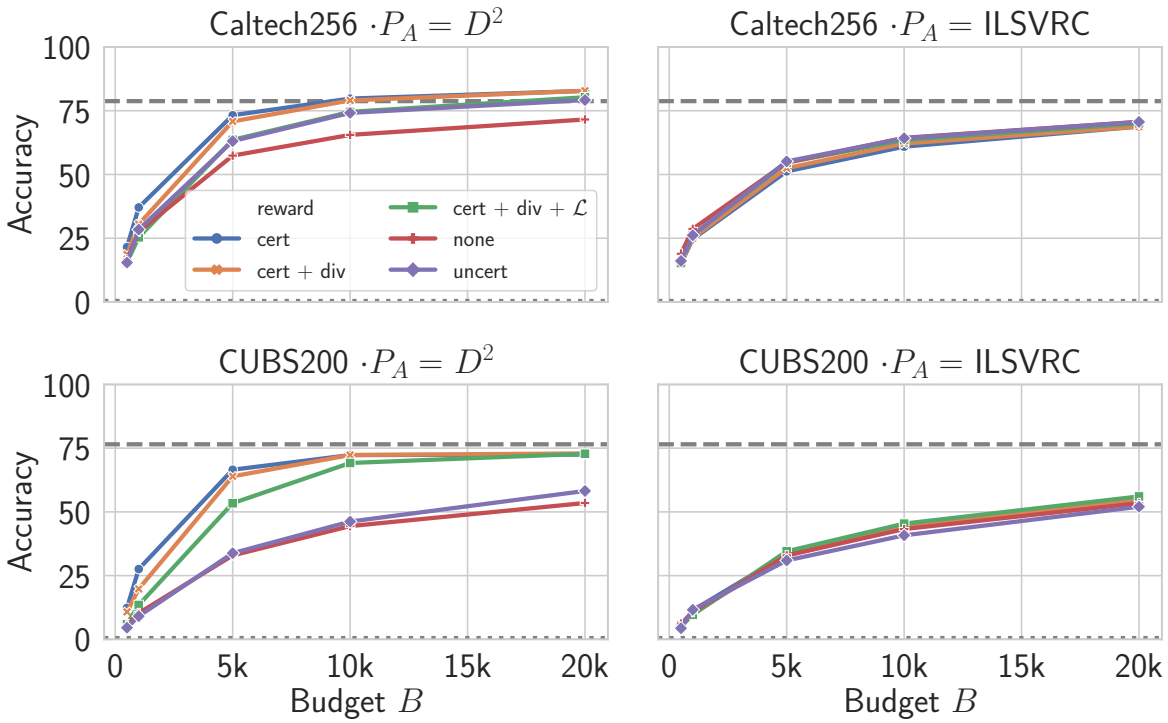


Figure 6.8: Reward ablation. cert: certainty, uncert: uncertainty, div: diversity, $\mathcal{L}$: loss, none: no reward (`random` strategy).

policy, in addition, learns to sample coarser actions related to the F_V's task e.g., predominantly drawing from birds and animals images to knockoff `CUBS200`.

What makes for a good *reward*? Using the `adaptive` sampling strategy, we now address influence of three rewards (discussed in Section 6.3.1.2). We observe: (i) **closed-world** (Fig. 6.8 left): All reward signals in `adaptive` helps with the sample efficiency over `random`. Reward `cert` (Eq. 6.4, which encourages exploitation) provides the best feedback signal. Including other rewards (Eq. 6.5-6.6) slightly deteriorates performance, as they encourage *exploration* over related or unseen actions – which is not ideal in a closed-world. Reward `uncert`, a popular measure used in AL literature (Ebert et al., 2012; Beluch et al., 2018; Settles and Craven, 2008) underperforms in our setting since it encourages uncertain (in our case, irrelevant) images. (ii) **open-world** (Fig. 6.8 right): Using all rewards (Eq. 6.4-6.6) display only none-to-marginal improvements for all choices of F_V, with the highest improvement in `CUBS200`. However, we notice an influence on learnt policies where adopting exploration (`div` + $\mathcal{L}$) with exploitation (`cert`) goals result in a softer probability distribution π over the action space and in turn, encouraging related images.

Can we train knockoffs with *truncated blackbox outputs*? So far, we found adversary's *attack* objective of knocking off blackbox models can be effectively carried

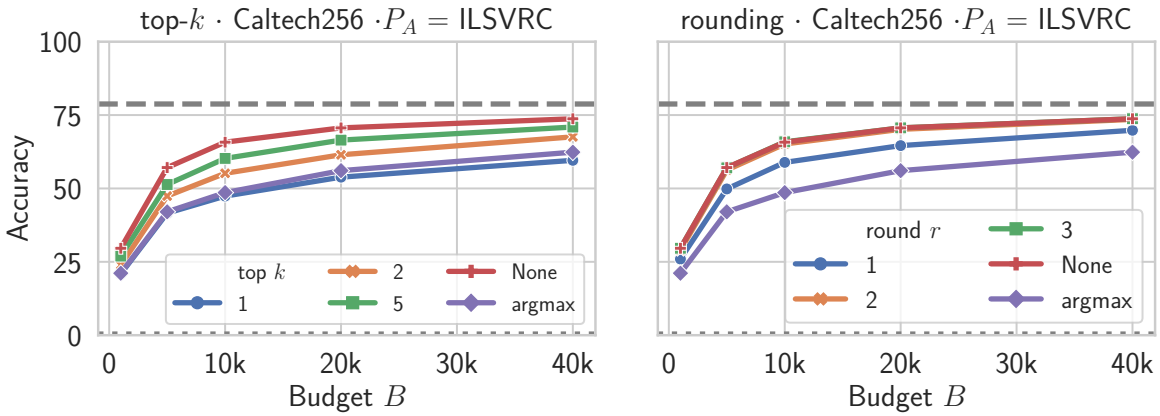


Figure 6.9: Truncated posteriors. Influence of training knockoff with truncated posteriors.

out with minimal assumptions. Now we explore the influence of victim's *defense* strategy of reducing informativeness of blackbox predictions to counter adversary's model stealing attack. We consider two truncation strategies: (a) top-k: top-k (out of K) unnormalized posterior probabilities are retained, while rest are zeroed-out; (b) rounding r: posteriors are rounded to r decimals e.g., round(0.127, r=2) = 0.13. In addition, we consider the extreme case "argmax", where only index $k = \arg\max_k y_k$ is returned.

From Figure 6.9 (with $K = 256$), we observe: (i) truncating $\boldsymbol{y}_i$ – either using top-k or rounding – slightly impacts the knockoff performance, with argmax achieving 0.76-0.84× accuracy of original performance for any budget B; (ii) top-k: even small increments of k significantly recovers the original performance – 0.91× at $k = 2$ and 0.96× at $k = 5$; (iii) rounding: recovery is more pronounced, with 0.99× original accuracy achieved at just $r = 2$. We find model functionality stealing minimally impacted by reducing informativeness of blackbox predictions.

6.5.2 *Architecture choice*

In the previous section, we found model functionality stealing to be consistently effective while keeping the architectures of the blackbox and knockoff fixed. Now we study the influence of the architectural choice F_A vs. F_V.

How does the *architecture* of F_A influence knockoff performance? We study the influence using two choices of the blackbox F_V architecture: Resnet-34 (He et al., 2016a) and VGG-16 (Simonyan and Zisserman, 2014). Keeping these fixed, we vary architecture of the knockoff F_A by choosing from: Alexnet (Krizhevsky et al., 2012), VGG-16 (Simonyan and Zisserman, 2014), Resnet-{18, 34, 50, 101} (He et al., 2016a), and Densenet-161 (Huang et al., 2017a).

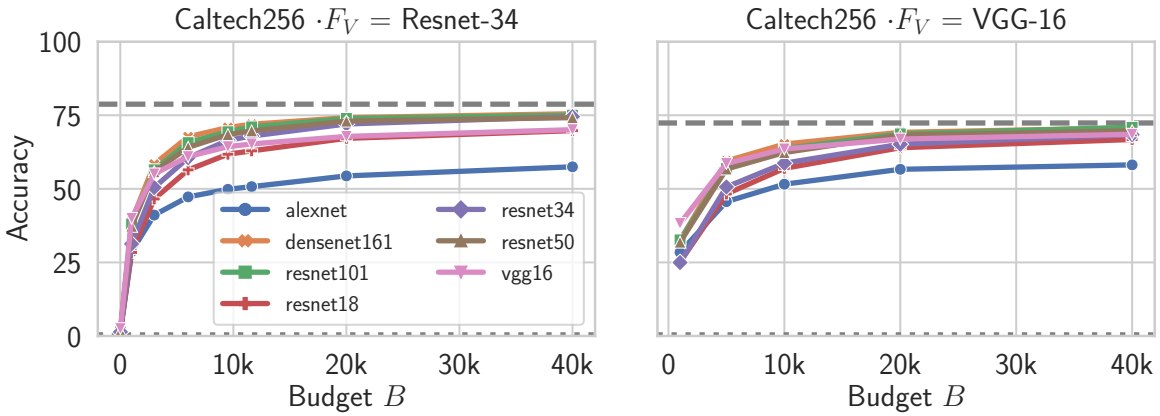


Figure 6.10: Architecture choices. F_V (left: Resnet-34 and right: VGG-16) and F_A (lines in each plot).

From Figure 6.10, we observe: (i) performance of the knockoff ordered by model complexity: Alexnet (lowest performance) is at one end of the spectrum while significantly more complex Resnet-101/Densenet-161 are at the other; (ii) performance transfers across model families: Resnet-34 achieves similar performance when stealing VGG-16 and vice versa; (iii) complexity helps: selecting a more complex model architecture of the knockoff is beneficial. This contrasts KD settings where the objective is to have a more compact student (knockoff) model.

6.5.3 *Stealing Functionality of a Real-world Black-box Model*

Now we validate how our model functionality stealing attack translates to a real-world scenario. Image recognition services are gaining popularity allowing users to obtain image-predictions for a variety of tasks at low costs ($1-2 per 1k queries). These image recognition APIs have also been used to evaluate other attacks e.g., adversarial examples (Liu et al., 2017; Bhagoji et al., 2017; Ilyas et al., 2018). We focus on a facial characteristics API which given an image, returns attributes and confidences per face. Note that in this experiment, we have semantic information of blackbox output classes.

Collecting P_A. The API returns probability vectors per face in the image and thus, querying irrelevant images leads to a wasted result with no output information. Hence, we use two face image sets P_A for this experiment: CelebA (220k images) (Liu et al., 2015) and OpenImages-Faces (98k images). We create the latter by cropping faces (plus margin) from images in the OpenImages dataset (Kuznetsova et al., 2018).

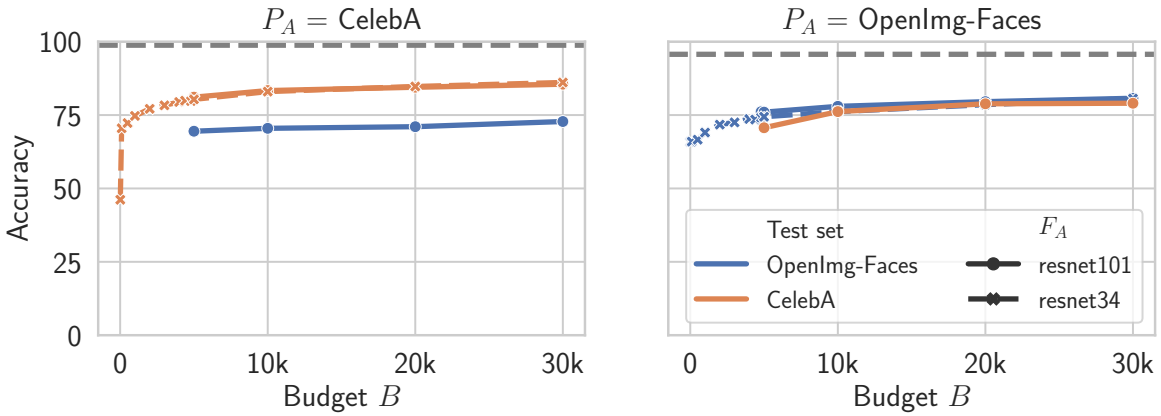


Figure 6.11: Knocking-off a real-world API. Performance of the knockoff achieved with two choices of P_A.

Evaluation. Unlike previous experiments, we cannot access victim's test data. Hence, we create test sets for each image set by collecting and manually screening seed annotations from the API on ∼5K images.

How does this translate to the *real-world*? We model two variants of the knockoff using the `random` strategy (`adaptive` is not used since no relevant auxiliary information of images are available). We present each variant using two choices of architecture F_A: a compact Resnet-34 and a complex Resnet-101. From Figure 6.11, we observe: (i) strong performance of the knockoffs achieving 0.76-0.82× performance as that of the API on the test sets; (ii) the diverse nature OpenImages-Faces helps improve generalization resulting in 0.82× accuracy of the API on both test-sets; (iii) the complexity of F_A does not play a significant role: both Resnet-34 and Resnet-101 show similar performance indicating a compact architecture is sufficient to capture discriminative features for this particular task.

We find model functionality stealing translates well to the real-world with knockoffs exhibiting a strong performance. The knockoff circumvents monetary and labour costs of: (a) collecting images; (b) obtaining expert annotations; and (c) tuning a model. As a result, an inexpensive knockoff can be trained which exhibits strong performance, using victim API queries amounting to only $30.

6.6 CONCLUSION

In this chapter, we investigated the problem of model functionality stealing where an adversary transfers the functionality of a victim model into a knockoff via blackbox access. In spite of minimal assumptions on the blackbox, we demonstrated the surprising effectiveness of our approach. Finally, we validated our approach on a real-world image recognition API and found strong performance of knockoffs. We

find functionality stealing poses a real-world threat that potentially undercuts an increasing number of deployed ML models.

In the next chapter, we work towards the first effective defense that mitigates the threat highlighted in this chapter.

7

PREDICTION POISONING

Having studied the effectiveness of model stealing attacks in Chapter 6, we now focus on defenses to mitigate such attacks. It is particularly important to work towards effective defenses, as advances in model functionality stealing attacks threaten the business model of ML applications, which require a lot of time, money, and effort to develop. Existing defenses take a passive role against stealing attacks, such as by truncating predicted information. We find such passive defenses ineffective against DNN stealing attacks. In this chapter, we propose the first defense which actively perturbs predictions targeted at poisoning the training objective of the attacker. We find our defense effective across a wide range of challenging datasets and DNN model stealing attacks, and additionally outperforms existing defenses. Our defense is the first that can withstand highly accurate model stealing attacks for tens of thousands of queries, amplifying the attacker's error rate up to a factor of $85\times$ with minimal impact on the utility for benign users.

The content of this chapter is based on Orekondy et al. (2020b). As a first author, Tribhuvanesh Orekondy conducted all the experiments and was the main writer for the conference paper.

7.1 INTRODUCTION

Effectiveness of state-of-the-art DNN models at a variety of predictive tasks has encouraged their usage in a variety of real-world applications e.g., home assistants, autonomous vehicles, commercial cloud APIs. Models in such applications are valuable intellectual property of their creators, as developing them for commercial use is a product of intense labour and monetary effort. Hence, it is vital to preemptively identify and control threats from an adversarial lens focused at such models. In this chapter we address model stealing, which involves an adversary attempting to counterfeit the functionality of a target victim ML model by exploiting black-box access (query inputs in, posterior predictions out).

Stealing attacks dates back to Lowd and Meek (2005a), who addressed reverse-engineering linear spam classification models. Recent literature predominantly focus on DNNs (specifically CNN image classifiers), and are shown to be highly effective (Tramèr et al., 2016) on complex models (Orekondy et al., 2019b), even without knowledge of the victim's architecture (Papernot et al., 2017b) nor the training data distribution. The attacks have also been shown to be highly effective at replicating pay-per-query image prediction APIs, for as little as $30 (Orekondy et al., 2019b).

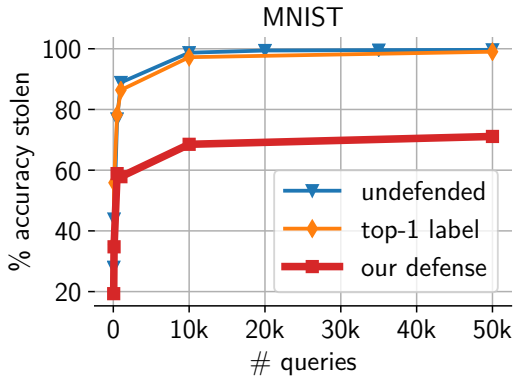
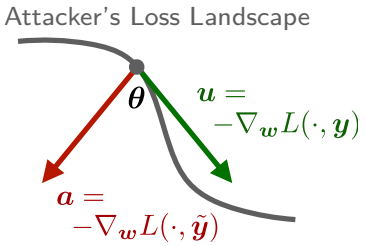


Figure 7.1: We find existing defenses (orange line) ineffective against recent attacks. Our defense (red line) in contrast significantly mitigates the attacks.



Our Perturbation Objective:

$$\underset{\tilde{y}}{\text{argmax}}\ \theta \quad \text{s.t} \quad \text{dist}(y, \tilde{y}) \leq \epsilon$$

Figure 7.2: We perturb posterior predictions $\tilde{y} = y + \delta$, with an objective of poisoning the adversary's gradient signal.

Defending against stealing attacks however has received little attention and is lacking. Existing defense strategies aim to either *detect* stealing query patterns (Juuti et al., 2019), or degrade quality of predicted posterior via *perturbation*. Since detection makes strong assumptions on the attacker's query distribution (e.g., small L_2 distances between successive queries), our focus is on the more popular perturbation-based defenses. A common theme among such defenses is accuracy-preserving posterior perturbation: the posterior distribution is manipulated while retaining the top-1 label. For instance, rounding decimals (Tramèr et al., 2016), revealing only high-confidence predictions (Orekondy et al., 2019b), and introducing ambiguity at the tail end of the posterior distribution (Lee et al., 2018). Such strategies benefit from preserving the accuracy metric of the defender. However, in line with previous works (Tramèr et al., 2016; Orekondy et al., 2019b; Lee et al., 2018), we find models can be effectively stolen using just the top-1 predicted label returned by the black-box. Specifically, in many cases we observe $<1\%$ difference between attacks that use the full range of posteriors (blue line in Fig. 7.1) to train stolen models and the top-1 label (orange line) alone. In this chapter, we work towards effective defenses (red line in Fig. 7.1) against DNN stealing attacks with minimal impact to defender's accuracy.

The main insight to our approach is that unlike a benign user, a model stealing attacker additionally uses the predictions to *train* a replica model. By introducing controlled perturbations to predictions, our approach targets poisoning the training objective (see Fig. 7.2). Our approach allows for a utility-preserving defense, as well as trading-off a marginal utility cost to significantly degrade attacker's performance. As a practical benefit, the defense involves a single hyperparameter (perturbation utility budget) and can be used with minimal overhead to any classification model without retraining or modifications.

We rigorously evaluate our approach by defending six victim models, against four recent and effective DNN stealing attack strategies (Papernot et al., 2017b; Juuti et al., 2019; Orekondy et al., 2019b). Our defense consistently mitigates all stealing attacks and further shows improvements over multiple baselines. In particular, we find our defenses degrades the attacker's query sample efficiency by 1-2 orders of magnitude. Our approach significantly reduces the attacker's performance (e.g., 30-53% reduction on MNIST and 13-28% on CUB200) at a marginal cost (1-2%) to defender's test accuracy. Furthermore, our approach can achieve the same level of mitigation as baseline defenses, but by introducing significantly lesser perturbation.

Contributions. (i) We propose the first utility-constrained defense against DNN model stealing attacks; (ii) We present the first active defense which poisons the attacker's training objective by introducing bounded perturbations; and (iii) Through extensive experiments, we find our approach consistently mitigate various attacks and additionally outperform baselines.

7.2 PRELIMINARIES

Model functionality stealing. Model stealing attacks are cast as an interaction between two parties: a victim/defender V ('teacher' model) and an attacker A ('student' model). The only means of communication between the parties are via black-box queries: attacker queries inputs $\boldsymbol{x} \in \mathcal{X}$ and defender returns a posterior probability distribution $\boldsymbol{y} \in \Delta^K = P(\boldsymbol{y}|\boldsymbol{x}) = F_V(\boldsymbol{x})$, where $\Delta^K = \{\boldsymbol{y} \succeq 0, \mathbf{1}^T\boldsymbol{y} = 1\}$ is the probability simplex over K classes (we use K instead of $K-1$ for notational convenience). The attack occurs in two (sometimes overlapping) phases: (i) *querying*: the attacker uses the black-box as an oracle labeler on a set of inputs to construct a 'transfer set' of input-prediction pairs $\mathcal{D}^{\text{transfer}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{B}$; and (ii) *training*: the attacker trains a model F_A to minimize the empirical risk on $\mathcal{D}^{\text{transfer}}$. The end-goal of the attacker is to maximize accuracy on a held-out test-set (considered the same as that of the victim for evaluation purposes).

Knowledge-limited attacker. In model stealing, attackers justifiably lack complete knowledge of the victim model F_V. Of specific interest are the model architecture and the input data distribution to train the victim model $P_V(X)$ that are not known to the attacker. Since prior work (Hinton et al., 2015; Papernot et al., 2016; Orekondy et al., 2019b) indicates functionality largely transfers across architecture choices, we now focus on the query data used by the attacker. Existing attacks can be broadly categorized based on inputs $\{\boldsymbol{x} \sim P_A(X)\}$ used to query the black-box: (a) *independent distribution*: (Tramèr et al., 2016; Correia-Silva et al., 2018; Orekondy et al., 2019b) samples inputs from some distribution (e.g., ImageNet for images, uniform noise) independent to input data used to train the victim model; and (b) *synthetic set*: (Papernot et al., 2017b; Juuti et al., 2019) augment a limited set of seed data by

adaptively querying perturbations (e.g., using FGSM) of existing inputs. We address both attack categories in our chapter.

Defense objectives. We perturb predictions in a controlled setting: $\tilde{\boldsymbol{y}} = F_V^{\delta}(\boldsymbol{x}) = \boldsymbol{y} + \boldsymbol{\delta}$ s.t. $\tilde{\boldsymbol{y}}, \boldsymbol{y} \in \Delta^K$. The defender has two (seemingly conflicting) objectives: (i) **utility**: such that perturbed predictions remain useful to a benign user. We consider two utility measures: (a) $\text{Acc}(F_V^{\delta}, \mathcal{D}^{\text{test}})$: accuracy of defended model on test examples; and (b) $\text{dist}(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = ||\boldsymbol{y} - \tilde{\boldsymbol{y}}||_p = \epsilon$ to measure perturbation. (ii) **non-replicability**: to reduce the test accuracy of an attacker (denoted as $\text{Acc}(F_A, \mathcal{D}^{\text{test}})$) who exploits the predictions to train a replica F_A on $\mathcal{D}^{\text{transfer}}$. For consistency, we evaluate both the defender's and attacker's stolen model accuracies on the same set of test examples $\mathcal{D}^{\text{test}}$.

Defender's assumptions. We closely mimic an assumption-free scenario similar to existing perturbation-based defenses. The scenario entails the knowledge-limited defender: (a) unaware whether a query is malicious or benign; (b) lacking prior knowledge of the strategy used by an attacker; and (c) perturbing each prediction independently (hence circumventing Sybil attacks). For added rigor, we also study attacker's countermeasures to our defense in Section 7.4.

7.3 APPROACH: MAXIMIZING ANGULAR DEVIATION BETWEEN GRADIENTS

Motivation: Targeting first-order approximations. We identify that the attacker eventually optimizes parameters of a stolen model $F(\cdot; \boldsymbol{w})$ (we drop the subscript $\cdot_A$ for readability) to minimize the loss on training examples $\{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}$. Common to a majority of optimization algorithms is estimating the first-order approximation of the empirical loss, by computing the gradient of the loss w.r.t. the model parameters $\boldsymbol{w} \in \mathbb{R}^D$:

$$\boldsymbol{u} = -\nabla_w L(F(\boldsymbol{x}; \boldsymbol{w}), \boldsymbol{y}) \tag{7.1}$$

Maximizing Angular Deviation (MAD). The core idea of our approach is to perturb the posterior probabilities $\boldsymbol{y}$ which results in an adversarial gradient signal that maximally deviates (see Fig. 7.2) from the original gradient (Eq. 7.1). More formally, we add targeted noise to the posteriors which results in a gradient direction:

$$\boldsymbol{a} = -\nabla_w L(F(\boldsymbol{x}; \boldsymbol{w}), \tilde{\boldsymbol{y}}) \tag{7.2}$$

to maximize the angular deviation between the original and the poisoned gradient signals:

$$\max_{\boldsymbol{a}} \; 2(1 - \cos \angle(\boldsymbol{a}, \boldsymbol{u})) = \max_{\hat{\boldsymbol{a}}} \; ||\hat{\boldsymbol{a}} - \hat{\boldsymbol{u}}||_2^2 \quad (\hat{\boldsymbol{a}} = \boldsymbol{a}/||\boldsymbol{a}||_2, \hat{\boldsymbol{u}} = \boldsymbol{u}/||\boldsymbol{u}||_2) \tag{7.3}$$

Given that the attacker model is trained to match the posterior predictions, such as by minimizing the cross-entropy loss $L(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = -\sum_k \tilde{y}_k \log y_k$ we rewrite Equation (7.2) as:

$$\boldsymbol{a} = -\nabla_{\boldsymbol{w}} L(F(\boldsymbol{x}; \boldsymbol{w}), \tilde{\boldsymbol{y}}) = \nabla_{\boldsymbol{w}} \sum_k \tilde{y}_k \log F(\boldsymbol{x}; \boldsymbol{w})_k = \sum_k \tilde{y}_k \nabla_{\boldsymbol{w}} \log F(\boldsymbol{x}; \boldsymbol{w})_k = \boldsymbol{G}^T \tilde{\boldsymbol{y}}$$

where $\boldsymbol{G} \in \mathbb{R}^{K \times D}$ represents the Jacobian over log-likelihood predictions $F(\boldsymbol{x}; \boldsymbol{w})$ over K classes w.r.t. parameters $\boldsymbol{w} \in \mathbb{R}^D$. By similarly rewriting Equation (7.1), substituting them in Equation (7.3) and including the constraints, we arrive at our poisoning objective (Eq. 7.4-7.7) of our approach which we refer to as MAD. We can optionally enforce preserving accuracy of poisoned prediction via constraint (7.8), which will be discussed shortly.

$$\begin{aligned}
\max_{\tilde{\boldsymbol{y}}} \quad & \left\| \frac{\boldsymbol{G}^T \tilde{\boldsymbol{y}}}{||\boldsymbol{G}^T \tilde{\boldsymbol{y}}||_2} - \frac{\boldsymbol{G}^T \boldsymbol{y}}{||\boldsymbol{G}^T \boldsymbol{y}||_2} \right\|_2^2 && (= H(\tilde{\boldsymbol{y}})) && (7.4) \\
\text{where} \quad & \boldsymbol{G} = \nabla_{\boldsymbol{w}} \log F(\boldsymbol{x}; \boldsymbol{w}) && (\boldsymbol{G} \in \mathbb{R}^{K \times D}) && (7.5) \\
\text{s.t} \quad & \tilde{\boldsymbol{y}} \in \Delta^K && \text{(Simplex constraint)} && (7.6) \\
& \text{dist}(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \leq \epsilon && \text{(Utility constraint)} && (7.7) \\
& \arg\max_k \tilde{\boldsymbol{y}}_k = \arg\max_k \boldsymbol{y}_k && \text{(For variant MAD-argmax)} && (7.8)
\end{aligned}$$

The above presents a challenge of black-box optimization problem for the defense since the defender justifiably lacks access to the attacker model F (Eq. 7.5). Apart from addressing this challenge in the next few paragraphs, we also discuss (a) solving a non-standard and non-convex constrained maximization objective; and (b) preserving accuracy of predictions via constraint (7.8).

Estimating $\boldsymbol{G}$. Since we lack access to adversary's model F, we estimate the jacobian $\boldsymbol{G} = \nabla_{\boldsymbol{w}} \log F_{\text{sur}}(\boldsymbol{x}; \boldsymbol{w})$ (Eq. 7.5) per input query $\boldsymbol{x}$ using a surrogate model F_{sur}. We empirically determined choice of *architecture* of F_{sur} robust to choices of adversary's architecture F. However, the *initialization* of F_{sur} plays a crucial role, with best results on a fixed randomly initialized model. We conjecture this occurs due to surrogate models with a high loss provide better gradient signals to guide the defender.

Heuristic solver. Gradient-based strategies to optimize objective (Eq. 7.4) often leads to poor local maxima. This is in part due to the objective increasing in all directions around point $\boldsymbol{y}$ (assuming $\boldsymbol{G}$ is full-rank), making optimization sensitive to initialization. Consequently, we resort to a heuristic to solve for $\tilde{\boldsymbol{y}}$. Our approach is motivated by Hoffman (1981), who show that the maximum of a convex function over a compact convex set occurs at the extreme points of the set. Hence, our two-step solver: (i) searches for a maximizer $\boldsymbol{y}^*$ for (7.4) by iterating over the K extremes

F_V	Acc(F_V)	Acc(F_A)			
		`jbda`	`jbself`	`jbtop3`	`k.off`
MNIST (LeNet)	99.4	89.2	89.4	87.3	99.1
FashionMNIST (LeNet)	92.0	38.7	45.8	68.7	69.2
CIFAR10 (VGG16)	92.0	28.6	20.7	73.8	78.7
CIFAR100 (VGG16)	72.2	5.3	2.9	39.2	51.9
CUB200 (VGG16)	80.4	6.8	3.9	21.5	65.1
Caltech256 (VGG16)	80.0	12.5	16.0	29.5	74.6

Table 7.1: Victim models and accuracies. All accuracies are w.r.t undefended victim model.

$\boldsymbol{y}_k$ (where y_k=1) of the probability simplex Δ^K; and (ii) then computes a perturbed posterior $\tilde{\boldsymbol{y}}$ as a linear interpolation of the original posteriors $\boldsymbol{y}$ and the maximizer $\boldsymbol{y}^*$: $\tilde{\boldsymbol{y}} = (1-\alpha)\boldsymbol{y} + \alpha\boldsymbol{y}^*$, where α is selected such that the utility constraint (Eq. 7.7) is satisfied.

Variant: MAD-argmax. Within our defense formulation, we encode an additional constraint (Eq. 7.8) to preserve the accuracy of perturbed predictions. MAD-argmax variant helps us perform accuracy-preserving perturbations similar to prior work. But in contrast, the perturbations are *constrained* (Eq. 7.7) and are specifically introduced to maximize the MAD objective. We enforce the accuracy-preserving constraint in our solver by iterating over extremes of intersection of sets Eq.(7.6) and (7.8): $\Delta^K_k = \{\boldsymbol{y} \succeq 0, \mathbf{1}^T\boldsymbol{y} = 1, y_k \geq y_j, k \neq j\} \subseteq \Delta^K$.

7.4 EXPERIMENTAL RESULTS

7.4.1 *Experimental Setup*

Victim models and datasets. We set up six victim models (see column 'F_V' in Table 7.1), each model trained on a popular image classification dataset. All models are trained using SGD (LR = 0.1) with momentum (0.5) for 30 (LeNet) or 100 epochs (VGG16), with a LR decay of 0.1 performed every 50 epochs. We train and evaluate each victim model on their respective train and test sets.

Attack strategies. We hope to broadly address all DNN model stealing strategies during our defense evaluation. To achieve this, we consider attacks that vary in query data distributions (independent and synthetic; see Section 7.2) and strategies (random and adaptive). Specifically, in our experiments we use the following attack models: (i) *Jacobian-based Data Augmentation* '`JBDA`' (Papernot et al., 2017b); (ii,iii) '`JB-self`'

and 'JB-top3' (Juuti et al., 2019); and (iv) *Knockoff Nets* 'knockoff' (Orekondy et al., 2019b); We follow the default configurations of the attacks where possible.

In all attack strategies, the adversary trains a model F_A to minimize the cross-entropy loss on a transfer set ($\mathcal{D}^{\text{transfer}} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{B}$) obtained by using the victim model F_V to pseudo-label inputs $\boldsymbol{x}_i$ (sampled or adaptively synthesized). By default, we use B=50K queries, which achieves reasonable performance for all attacks and additionally makes defense evaluation tractable. The size of the resulting transfer set (B=50K examples) is comparable (e.g., 1× for CIFAR10/100, 2.1× for Caltech256) to size of victim's training set. In line with prior work (Papernot et al., 2016; Orekondy et al., 2019b), we too find (Section 7.4.2.3) attack and defense performances are unaffected by choice of architectures, and hence use the victim architecture for the stolen model F_A. Due to the complex parameterization of VGG-16 (100M+), we initialize the weights from a pretrained TinyImageNet or ImageNet model (except for the last FC layer, which is trained from scratch). All stolen models are trained using SGD (LR=0.1) with momentum (0.5) for 30 epochs (LeNet) and 100 epochs (VGG16). We find choices of attacker's architecture and optimization does not undermine the defense (discussed in Section 7.4.2.3).

Effectiveness of attacks. We evaluate accuracy of resulting stolen models from the attack strategies as-is on the victim's test set, thereby allowing for a fair head-to-head comparison with the victim mode. The stolen model test accuracies, along with undefended victim model F_V accuracies are reported in Table 7.1. We observe for all six victim models, using just 50K black-box queries, attacks are able to significantly extract victim's functionality e.g., >87% on MNIST. We find the knockoff attack to be the strongest, exhibiting reasonable performance even on complex victim models e.g., 74.6% (0.93×Acc(F_V)) on Caltech256.

How good are existing defenses? Most existing defenses in literature (Tramèr et al., 2016; Orekondy et al., 2019b; Lee et al., 2018) perform some form of *information truncation* on the posterior probabilities e.g., rounding, returning top-k labels; all strategies preserve the rank of the most confident label. We now evaluate model stealing attacks on the extreme end of information truncation, wherein the defender returns just the top-1 'argmax' label. This strategy illustrates a rough lower bound on the strength of the attacker when using existing defenses. Specific to knockoff, we observe the attacker is minimally impacted on simpler datasets (e.g., 0.2% accuracy drop on CIFAR10). While this has a larger impact on more complex datasets involving numerous classes (e.g., a maximum of 23.4% drop observed on CUB200), the strategy also introduces a significant perturbation (L_1=1±0.5) to the posteriors. The results suggest existing defenses, which largely the top-1 label, are largely ineffective at mitigating model stealing attacks.

Defenses: Evaluation. We evaluate all defenses on a non-replicability vs. utility curve at various operating points ϵ of the defense. We furthermore evaluate the defenses for a large query budget (50K). We use as *non-replicability* the accuracy of the stolen model on held-out test data $\mathcal{D}^{\text{test}}$. We use two *utility* metrics:

(a) accuracy: test-accuracy of the defended model producing perturbed predictions on $\mathcal{D}^{\text{test}}$; and

(b) perturbation magnitude ϵ: measured as L_1 distance $||\boldsymbol{y} - \tilde{\boldsymbol{y}}||_1$.

Defense: Baselines. We compare our approaches against three methods:

(i) `reverse-sigmoid` (Lee et al., 2018): which softens the posterior distribution and introduces ambiguity among non-argmax probabilities. For this method, we evaluate non-replicability and utility metrics for the defense operating at various choices of their hyperparameter $\beta \in [0, 1]$, while keeping their dataset-specific hyperparameter γ fixed (MNIST: 0.2, FashionMNIST: 0.4, CIFAR10: 0.1, rest: 0.2).

(ii) `random noise`: For controlled random-noise, we add uniform random noise $\boldsymbol{\delta}_z$ on the logit prediction scores ($\tilde{\boldsymbol{z}} = \boldsymbol{z} + \boldsymbol{\delta}_z$, where $\boldsymbol{z} = \log(\frac{\boldsymbol{y}}{1-\boldsymbol{y}})$), enforce utility by projecting $\boldsymbol{\delta}_z$ to an ϵ_z-ball (Duchi et al., 2008), and renormalize probabilities $\tilde{\boldsymbol{y}} = \frac{1}{1+e^{-\tilde{\boldsymbol{z}}}}$.

(iii) `dp-sgd`: while our method and previous two baselines perturbs *predictions*, we also compare against introducing randomization to victim model *parameters* by training with the DP-SGD algorithm (Abadi et al., 2016b). DP is a popular technique to protect the model against training data inference attacks. This baseline allows us to verify whether the same protection extends to model functionality.

7.4.2 *Results*

In the follow sections, we demonstrate the effectiveness of our defense rigorously evaluated across a wide range of complex datasets, attack models, defense baselines, query, and utility budgets. For readability, we first evaluate the defense against attack models, proceed to comparing the defense against strong baselines and then provide an analysis of the defense.

7.4.2.1 *MAD Defense vs. Attacks*

Figure 7.3 presents evaluation of our defenses MAD (Eq. 7.4-7.7) and MAD-argmax (Eq. 7.4-7.8) against the four attack models. To successfully mitigate attacks as a defender, we want the defense curves (colored solid lines with operating points denoted by

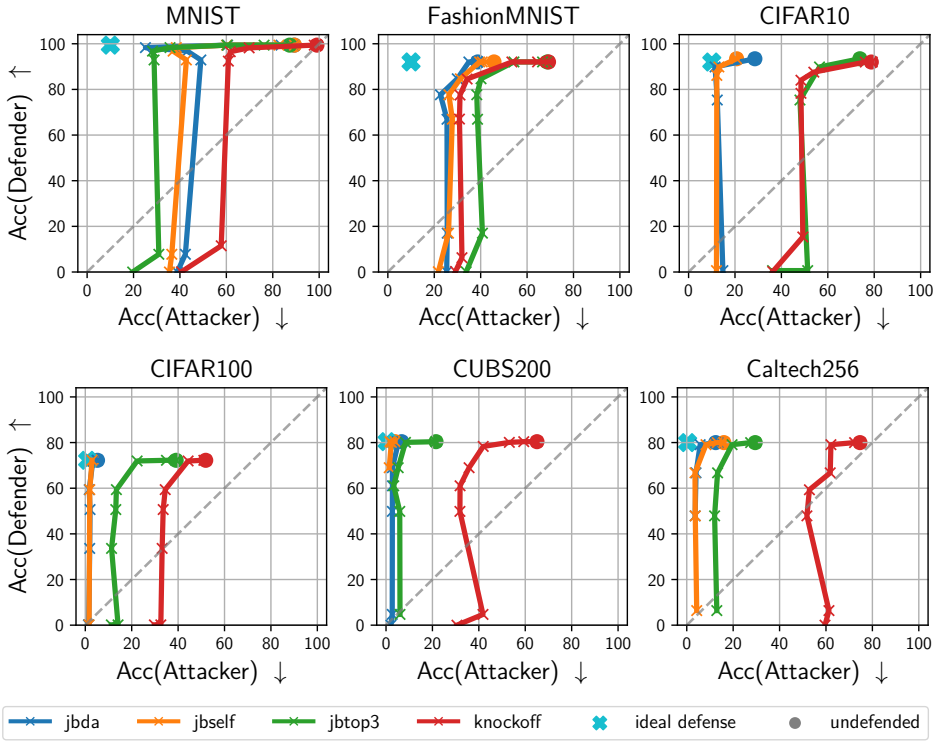


Figure 7.3: Attackers vs. our defense. Curves are obtained by varying degree of perturbation ϵ (Eq. 7.7) in our defense. ↑ denotes higher numbers are better and ↓, lower numbers are better. Non-replicability objective is presented on the x-axis and utility on the y-axis.

thin crosses) to move *away from undefended* accuracies (denoted by circular discs, where ϵ=0.0) *to ideal defense* performances (cyan cross, where Acc(Def.) is unchanged and Acc(Att.) is chance-level).

We observe from Figure 7.3 that by employing an identical defense across all datasets and attacks, the effectiveness of the attacker can be greatly reduced. Across all models, we find MAD provides reasonable operating points (above the diagonal), where defender achieves significantly higher test accuracies compared to the attacker. For instance, on MNIST, for $<$1% drop in defender's accuracy, our defense *simultaneously* reduces accuracy of the `jbtop3` attacker by 52% (87.3%→35.7%) and `knockoff` by 29% (99.1%→69.8%). We find similar promising results even on high-dimensional complex datasets e.g., on CUB200, a 23% (65.1%→41.9%) performance drop of `knockoff` for 2% drop in defender's test performance. Our results indicate effective defenses are achievable, where the defender can trade-off a marginal utility cost to drastically impede the attacker.

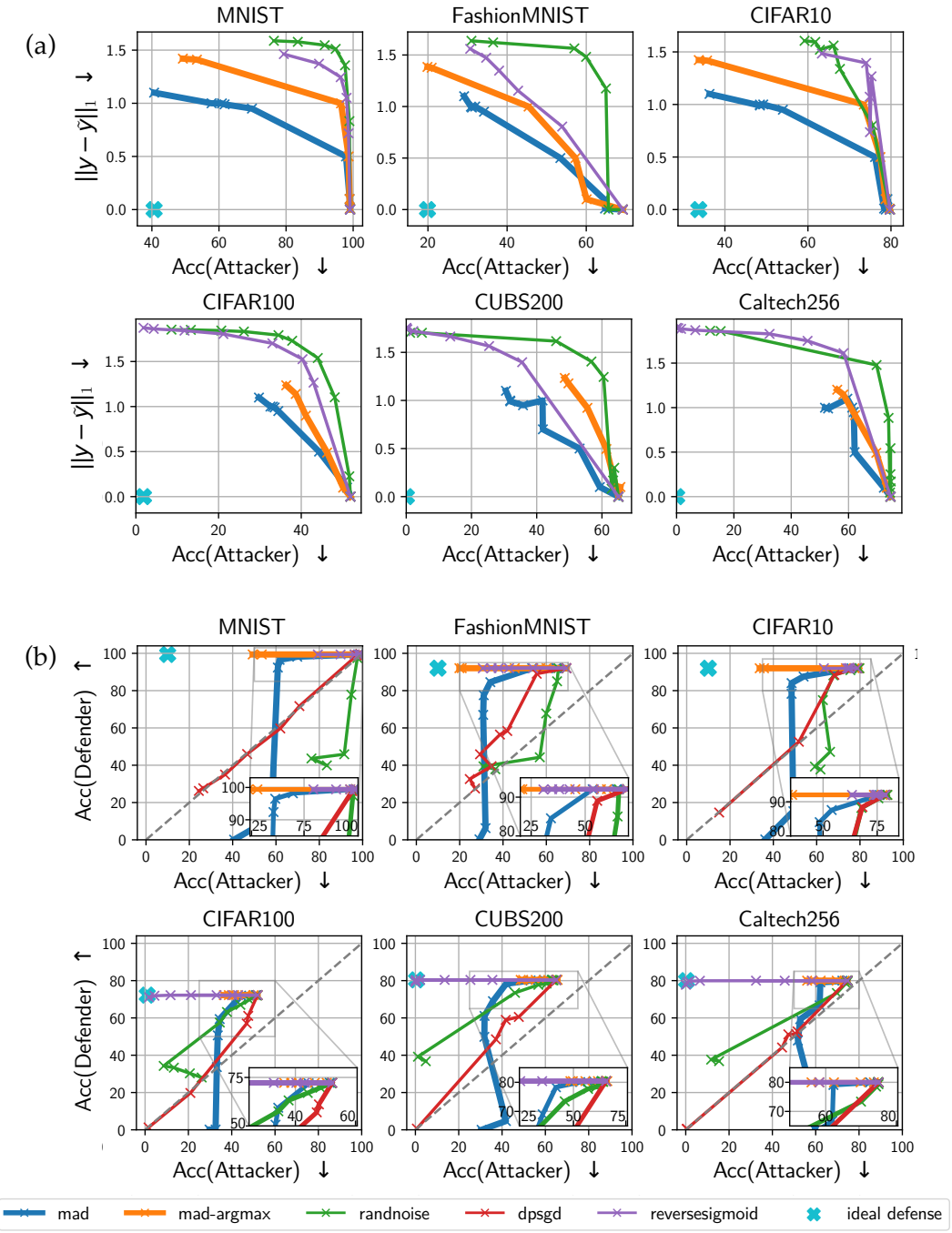


Figure 7.4: Knockoff attack vs. ours and baseline defenses (best seen magnified). Non-replicability is presented on the x-axis. On y-axis, we present two utility measures: **(a) top:** Utility = L_1 distance **(b) bottom:** Utility = Defender's accuracy. Region above the diagonal indicates instances where defender outperforms the attacker.

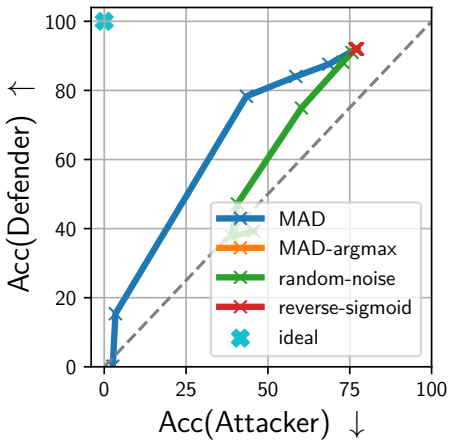


Figure 7.5: Attacker argmax. Follow-up to Figure 7.4b (CIFAR10), but with attacker using only the argmax label.

7.4.2.2 *MAD Defense vs. Baseline Defenses*

We now study how our approach compares to baseline defenses, by evaluating the defenses against the `knockoff` attack (which resulted in the strongest attack in our experiments). From Figure 7.4, we observe:

(i) *Utility objective* = L_1 *distance* (Fig. 7.4a): Although random-noise and reverse-sigmoid reduce attacker's accuracy, the strategies in most cases involves larger perturbations. In contrast, MAD and MAD-argmax provides similar non-replicability (i.e., Acc(Att.)) with significantly lesser perturbation, especially at lower magnitudes. For instance, on MNIST (first column), MAD (L_1 = 0.95) reduces the accuracy of the attacker to under 80% with 0.63× the perturbation as that of reverse-sigmoid and random-noise ($L_1 \approx 1.5$).

(ii) *Utility objective* = *argmax-preserving* (Fig. 7.4b): By setting a hard constraint on retaining the label of the predictions, we find the accuracy-preserving defenses MAD-argmax and reverse-sigmoid successfully reduce the performance of the attacker by at least 20% across all datasets. In most cases, we find MAD-argmax in addition achieves this objective by introducing lesser distortion to the predictions compared to reverse-sigmoid. For instance, in Fig. 7.4a, we find MAD-argmax consistently reduce the attacker accuracy to the same amount at lesser L_1 distances. In reverse-sigmoid, we attribute the large L_1 perturbations to a shift in posteriors towards a uniform distribution e.g., mean entropy of perturbed predictions is 3.02 $\pm$ 0.16 (max-entropy = 3.32) at L_1=1.0 for MNIST; in contrast, MAD-argmax displays a mean entropy of 1.79 $\pm$ 0.11. However, common to accuracy-preserving strategies is a pitfall that the top-1 label is retained. In Figure 7.5 (see overlapping red and yellow cross-marks), we present the results of training the attacker using only the top-1 label. In line with previous discussions, we find that the attacker is able to significantly recover

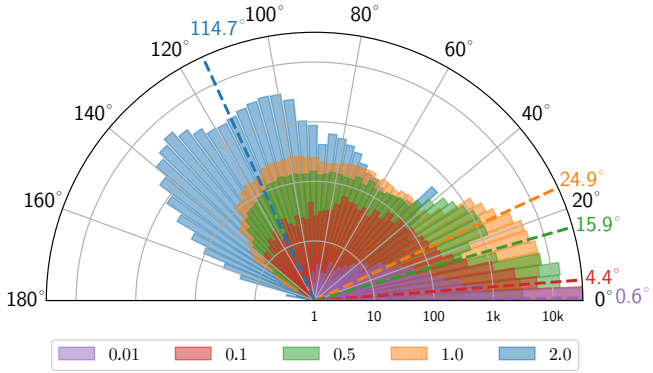


Figure 7.6: Histogram of angular deviations. Presented for MAD attack on CIFAR10 with various choices of ϵ.

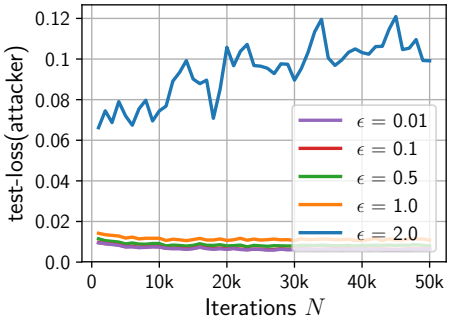


Figure 7.7: Test loss. Visualized during training. Colours and lines correspond to ϵ values in Fig. 7.6.

the original performance of the stolen model for accuracy-preserving defenses MAD-argmax and reverse-sigmoid.

(iii) *Non-replicability vs. utility trade-off* (Fig. 7.4b): We now compare our defense MAD (blue lines) with baselines (`rand-noise` and `dp-sgd`) which trade-off utility to mitigate model stealing. Our results indicate MAD offers a better defense (lower attacker accuracies for similar defender accuracies). For instance, to reduce the attacker's accuracy to <70%, while the defender's accuracy significantly degrades using `dp-sgd` (39%) and `rand-noise` (56.4%), MAD involves a marginal decrease of 1%.

7.4.2.3 *Analysis*

How much angular deviation does MAD introduce? To obtain insights on the angular deviation induced between the true and the perturbed gradient, we conduct an experiment by tracking the true gradient direction (which was unknown so far) at each training step. We simulate this by training an attacker model using online SGD (LR=0.001) over N iterations using B distinct images to query and a batch size of 1. At each step t of training, the attacker queries a randomly sampled input $\boldsymbol{x}_t$ to the defender model and backpropogates the loss resulting from $\tilde{\boldsymbol{y}}_t$. In this particular experiment, the perturbation $\tilde{\boldsymbol{y}}_t$ is crafted having exact knowledge of the attacker's parameters. We evaluate the angular deviation between gradients with ($\boldsymbol{a}$) and without ($\boldsymbol{u}$) the perturbation.

In Figure 7.6, we visualize a histogram of deviations: $\theta = \arccos \frac{\boldsymbol{u} \cdot \boldsymbol{a}}{||\boldsymbol{u}||||\boldsymbol{a}||}$, where $\boldsymbol{u} = \nabla_w L(\boldsymbol{w}_t, \boldsymbol{y}, \cdot)$ and $\boldsymbol{a} = \nabla_w L(\boldsymbol{w}_t, \tilde{\boldsymbol{y}}, \cdot)$. We observe: (i) although our perturbation space is severely restricted (a low-dimensional probability simplex), we can introduce surprisingly high deviations (0-115°) in the high-dimensional parameter space of the VGG16; (ii) for ϵ values at reasonable operating points which preserves the defender's accuracy within 10% of the undefended accuracy (e.g., $\epsilon \in [0.95, 0.99]$ for

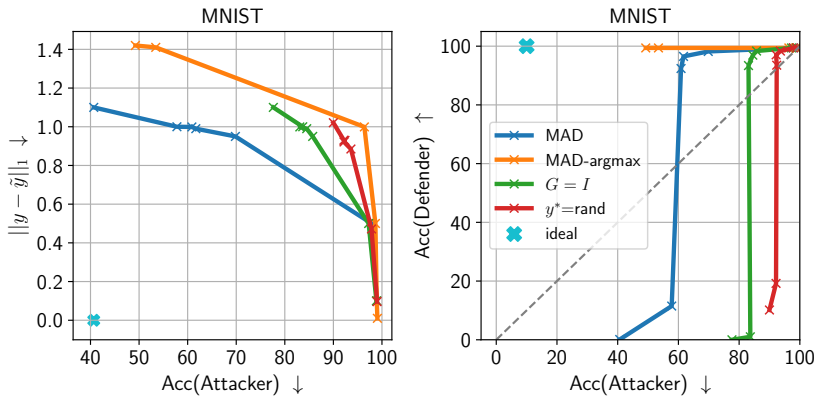


Figure 7.8: MAD ablation experiments. Utility = (left) L_1 distance (right) defender test accuracy.

CIFAR10), we see deviations with mean 24.9° (yellow bars in Fig. 7.6). This indicates that the perturbed gradient on an average leads to a slower decrease in loss function; (iii) on the extreme end, with $\epsilon = \epsilon_{\text{max}} = 2$, on an average, we find the perturbations successfully flips (>90°) the gradient direction leading to an increase on the test loss, as seen in Figure 7.7 (blue line). We also find the above observations reasonably transfers to a black-box attacker setting, where the perturbations are crafted without knowledge of the attacker's parameters. Overall, we find our approach considerably corrupts the attacker's gradient direction.

Ablative analysis. We present an ablation analysis of our approach in Figure 7.8. In this experiment, we compare our approach MAD and MAD-argmax to: (a) $\boldsymbol{G} = \boldsymbol{I}$: We substitute the jacobian $\boldsymbol{G}$ (Eq. 7.5) with a $K \times K$ identity matrix; and (b) $\boldsymbol{y}^*$=rand: Inner maximization term (Eq. 7.4) returns a random extreme of the simplex. Note that both (a) and (b) do not use the gradient information to perturb the posteriors.

From Figure 7.8, we observe: (i) poor performance of $\boldsymbol{y}^*$=rand, indicating random untargeted perturbations of the posterior probability is a poor strategy; (ii) $\boldsymbol{G} = \boldsymbol{I}$, where the angular deviation is maximized between the posterior probability vectors is a slightly better strategy; (ii) MAD outperforms the above approaches. Consequently, we find using the gradient information (although a proxy to the attacker's gradient signal) within our formulation (Equation 7.4) is crucial to providing better model stealing defenses.

Subverting the defense. We now explore various strategies an attacker can use to circumvent the defense. To this end, we evaluate the following strategies: (a) argmax: attacker uses only the most-confident label during training; (b) arch-*: attacker trains other choices of architectures; (c) nquery: attacker queries each image multiple times; (d) nquery+aug: same as (c), but with random cropping and horizontal flipping; and (e) opt-*: attacker uses an adaptive LR optimizer e.g., ADAM (Kingma and Ba, 2014).

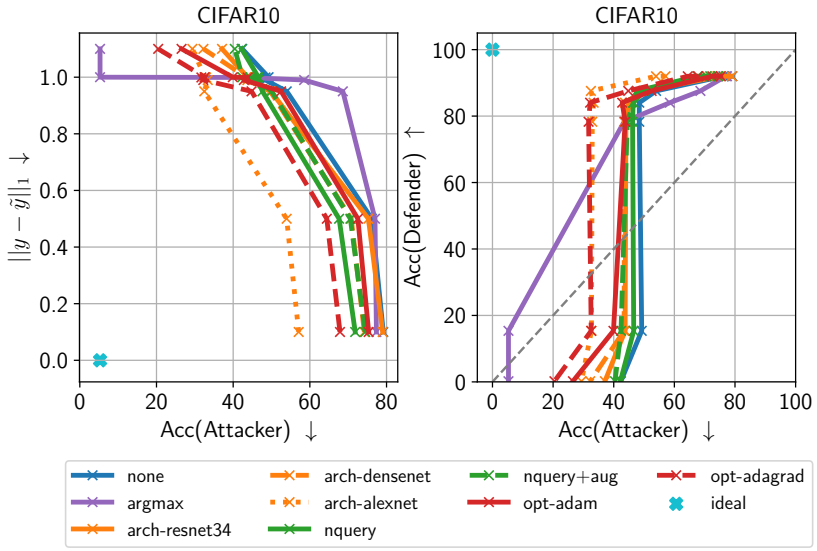


Figure 7.9: Subverting the defense.

We present results over the subversion strategies in Figure 7.9. We find our defense robust to above strategies. Our results indicate that the best strategy for the attacker to circumvent our defense is to discard the probabilities and rely only on the most confident label to train the stolen model. In accuracy-preserving defenses (see Fig. 7.5), this previously resulted in an adversary entirely circumventing the defense (recovering up to 1.0× original performance). In contrast, we find MAD is nonetheless effective in spite of the strategy, maintaining a 9% absolute accuracy reduction in attacker's stolen performance.

7.5 CONCLUSION

In this chapter, we were motivated by limited success of existing defenses against DNN model stealing attacks. While prior work is largely based on passive defenses focusing on information truncation, we proposed the first active defense strategy that attacks the adversary's training objective. We found our approach effective in defending a variety of victim models and against various attack strategies. In particular, we find our attack can reduce the accuracy of the adversary by up to 65%, without significantly affecting defender's accuracy.

8

CONCLUSION AND OUTLOOK

We have seen significant progress in machine learning recently. The progress has enabled a dizzying array of applications e.g., managing photo collections, smart home assistants, aiding medical diagnosis. Central to many such applications is sharing of information (e.g., data, parameters) between owners (e.g., individuals, medical organizations). There are many incentives for the owners to share the information, such as social rewards (when sharing photographs) or financial benefits (when monetizing predictions). However, we argue that the incentives present only one side of a double-edged sword. One the other side, the shared information can also be exploited by untrusted parties. In the specific case where the information is a function of private and confidential data, it is important to identify and mitigate entailing privacy risks.

In this thesis, we addressed leakage in machine learning systems along three research directions, which we will now briefly summarize. In the first direction, *leakage in visual data*, we argued that personal photos (similar to that shared on social networks) encodes a broad range of private information. Part I presented datasets, user studies, and techniques to identify and further mitigate privacy risks by redacting multimodal privacy-sensitive content in a utility-preserving manner. In the second direction, *leakage during learning*, we switched focus to a family of learning algorithms that requires individuals to share training artifacts computed on their corresponding private datasets. Here, we found that the artifacts unintentionally encode user-identifiable information across a wide range of challenging scenarios and tasks. In the third direction, *leakage during inference*, we investigated model stealing attacks and defenses. We initially found that attacks are surprisingly possible on complex black-box models and worked our way to propose defenses to withstand such attacks.

In this chapter, in Section 8.1, we begin by presenting a detailed summary of the chapters in this thesis and share resulting insights. Section 8.2 builds on some of these insights and reviews open problems to better understand leakage of information in machine learning systems.

8.1 KEY INSIGHTS AND CONCLUSIONS

This thesis we presented a framework to understand leakage of private and confidential information in machine learning techniques. In this framework, we considered a owner who (i) in possession of some private data (iii) trains a supervised predictive model and (iii) exposes the model as a black-box to perform inferences. At each

step, there are many incentives (e.g., financial) for the owner to share information. Over the following sections, we summarize our methods to understand and control privacy leakage when sharing such information.

8.1.1 *Part I: Leakage via Visual Data*

In the first part of the thesis, we studied the scenario where the owner wishes to share visual data – specifically personal photos – on the internet e.g., on social networks. To control the amount of private information leaked by such photos, existing research provides numerous solutions to mitigate information corresponding to facially identifiable features. However, there is scant research to address privacy aspects in images beyond facial features e.g., location, full name. This is problematic because such privacy aspects of individuals, when in explicit forms receive scrutiny, but do not when the same information is embedded in visual content.

In Chapter 3, we took the first steps towards addressing privacy in images from a broader perspective. We presented the first taxonomy of 68 privacy attributes over 9 categories inspired by privacy laws surrounding explicit content. Furthermore, we proposed the Visual Privacy (VISPR) dataset consisting of 22k images annotated with the privacy attributes. Using the VISPR dataset, we performed a user study to better understand the interplay of individuals, their privacy preferences, and their perception of privacy risks in images. The user study highlighted that individuals having diverse privacy preferences, suggesting that there does not exist a 'one size fits all' approach to controlling privacy leakage in images. More surprisingly in the user studies, we found that individuals consistently *underestimate* the privacy risk of image content and fail to enforce their own privacy preferences. To better evaluate privacy risks in visual content, we proposed an end-to-end method to estimate the privacy attributes revealed in an image and the resulting privacy risk. Remarkably, we found our approach in some scenarios outperforms the individuals themselves in estimating the privacy risk when sharing images.

In Chapter 4, we extended our work from Chapter 3 by controlling privacy leakage on a pixel-level. Here, we studied the problem of visual redactions: to identify and obfuscate privacy-sensitive regions in images while preserving its aesthetic quality (i.e., the utility). As a first step, we extended a 8.4k image subset of the VISPR dataset to additionally contain pixel-level annotations at an instance-level. Here, we found pixel-level localization of privacy-sensitive regions introduces a multi-modality challenge, as attributes arise from visual (e.g., fingerprints), textual (e.g., names), and multimodal (e.g., drivers license) sources. Consequently, by leveraging recent advances in instance-level segmentation and text processing, we proposed an ensemble approach to specifically address these challenges. While our approach allows to (imperfectly) identify privacy-sensitive regions, we found further redacting these regions introduces a privacy-utility trade off. Specifically, that under-redactions drastically increases privacy risks, and over-redacting gradually decreases the images'

overall utility rendering the image unsuitable for online sharing. Nonetheless, by exploiting variances in spatial extents of different attributes, our redaction approach was able to achieve 83% performance of human-based redactions.

In the above chapters, we studied our approaches as a zero-sum game over a user's privacy objective set against an adversarial objective; the objective is the user minimizing (alternatively the adversary maximizing) inference over various privacy attributes in a single personal image. While our approaches achieved promising results, we acknowledge that the performances might be unjustifiably optimistic in certain scenarios, such as when a user shares a stream of personal images (e.g., video clip). In such cases, we believe it might be more appropriate to identify and mitigate privacy risks assuming an adversary's access to the entire stream (i.e., worst-case estimates) rather than assessing risk in expectation (i.e., average-case estimates) as studied in our work. Furthermore, we also identify that the stream of photos might reveal additional privacy-related patterns that are not evident in a single photo (e.g., embarrassing actions). In spite of these limitations, we make some key contributions. We take the first steps towards identifying that images reveals a wide spectrum of private information and additionally collect large-scale image datasets annotated image-level and pixel-level labels across a range of privacy attributes. Furthermore, we propose techniques to automatically identify and mitigate privacy leakage (such as by redacting relevant pixels) in personal photos while preserving the utility of the image. We find that our privacy-enforcing techniques achieve performance comparable to human-level performance and in some cases also outperform them.

8.1.2 *Part II: Leakage during Learning*

In the second part of the thesis, we considered the scenario where the owner contributes data towards the training of model. Specifically, where the shared data corresponds to anonymous model parameters resulting from few gradient descent steps over the owner's private data. Such a sharing scheme is central to federated learning approaches where multiple owners intend to collaboratively train a model without sharing their raw data. As a result, it is vital that the model updates (shared instead of the raw data) does not reveal any information specific to the owner, or the training data.

In Chapter 5, we however found that the shared model parameters encodes subtle variations in which users capture and generate data. We found the statistical signal provided by variations can be exploited to re-identify users behind the model updates in a variety of challenging settings. To further exacerbate the concern, we additionally observed that user re-identification helps associate the identity to other sensitive attribute inferences. As the effectiveness of re-identification attacks is to a large degree the product of biases in the training data subsets, we propose a defense which introduces a decoy bias into the training data. Our defense strategies against re-identification attacks offered substantial protections with little effect to utility.

8.1.3 *Part III: Leakage during Inference*

In the final part of the thesis, we investigated how model predictions potentially leak confidential information. Specifically, we studied this leakage from the lens of model functionality stealing, where an adversary exploits the black-box interface to create a replica of the model and thereby stealing the intellectual property of the owner.

In Chapter 6, we proposed the first model functionality stealing attack that is effective on complex CNNs, while making fewer assumptions. In particular, we found that the attacker can leverage a large independent pool of unlabeled 'public' data to transfer knowledge from the victim blackbox model to a 'knockoff' model. In addition, we extended the attacker to perform sample-efficient querying by treating sampling as a multi-armed bandit problem, where (learnt) arms correspond to semantically similar groups in the public data. Our results especially raises concerns around Machine Learning as a Service (MLaaS) providers, who monetize prediction APIs to their models by charging a small fee per prediction (around $1-1.5 for 1k predictions). In the form of a case study, we found that our approach was able to steal a popular MLaaS model for as little as $30.

In Chapter 7, to address the concerns posed by our findings in Chapter 6, we proposed the first effective defense against model stealing attacks. As model stealing approaches predominantly involve an attacker training a model using the predictions, the key idea in our defense was to poison the attacker's training by perturbing predictions. Our optimization-based defense perturbed predictions to maximize deviations in the attacker's gradient signal, while minimally distorting the prediction. We found our defense able to withstand model stealing attacks for tens of thousands of queries while significantly amplifying the attacker's error rate.

8.2 FUTURE PERSPECTIVES

In this section, we discuss future perspectives along the three research directions between the thesis. In the last section, we conclude with a broader outlook for the field.

8.2.1 *Visual Privacy*

In Part I of the thesis, we investigated methods to identify and control leakage across a broad range of privacy attributes in personal photos. The end goal of our work was to study techniques that enables photo sharing by data owners (e.g., social media users) by revealing the minimal amount of information necessary in images. Now we layout some research directions to realize the goal of a 'Visual Privacy Advisor'.

Minimizing supervision. A general theme towards training models to identify private content in images – including our own work – is leveraging a large amount of annotated training data for supervising the training process. In here lies the key challenge that in most practical scenarios, *availability* of such training data, especially for high-risk categories (e.g., credit cards), is scarce. Consequently, minimizing the amount of supervision plays a major role to aid visual privacy tasks. One potential approach is to transfer the supervision burden from private annotated data to public prior knowledge e.g., in the form of Wikipedia text descriptions of certain attributes. Approaches in few-shot classification (Xian et al., 2016; Sung et al., 2018) show reasonable success towards this goal of training in spite of data scarcity.

Learning to obfuscate from obfuscated data. Previously, we discussed when a minimal amount of annotated *clean* training data is available to train visual privacy approaches. An alternate data availability scenario is when large amounts of *pre-obfuscated training data* is available (Gurari et al., 2019), where the privacy-sensitive regions are corrupted prior to sharing. However, this raises the paradoxical question on how to train approaches to identify private content without having ever seen it. A possible solution is to solve the paradox is exploiting contextual information *outside* the obfuscated region. For instance, recognizing adjacent computer peripherals (e.g., keyboard) to guide obfuscating content on computer screens, or the human body to obfuscate a person's face. Explicitly exploiting contextual location priors has shown reasonable success (Gould et al., 2008; Fulkerson et al., 2009; Krähenbühl and Koltun, 2011) in localizing objects, especially when only partial information (Verbeek and Triggs, 2008) is available. Perhaps similar contextual priors could form the basis for learning locations of private content from pre-obfuscated images.

Minimax formulation. Having presented some research directions to deal with data scarcity, we now move focus on directions to better control privacy leakage. One such direction involves viewing obfuscation as a minimax problem (Neumann, 1928): to find the best obfuscation that minimizes leakage while maximally preserving the obfuscated image's utility. Our approach in Chapter 4 did not entirely solve this minimax problem, but rather we used heuristics to find a range of possible obfuscations that satisfied the minimax objective. Consequently, it left the data owner with no guarantees on the resulting obfuscation. We believe one could build on top of recent advances (Roy and Boddeti, 2019; Bertran et al., 2019) to obtain obfuscation that provides theoretical bounds on resulting leakage. However, as these advances have a narrow notion of utility (by treating it as a discrete random variable), it is an open question whether they will directly apply to our problem where utility corresponds to image aesthetics.

Improving utility via fine-grained obfuscations. In our work on performing automatic redactions (Chapter 4), we viewed image obfuscation as a trade-off between

privacy and utility. In the previous paragraphs, we briefly presented some solutions to push the Pareto frontier of the privacy-utility curve along the privacy-axis. Now, we switch focus to the utility-axis: are there better alternatives than the black-out obfuscation strategy studied in Chapter 4? Literature from the human-computer interaction studies (Li et al., 2017c; Hasan et al., 2018) suggests a wide variety of such strategies exist (e.g., blurring, pixelization) that better preserve utility of the obfuscated image. While these provide a good starting point to better craft obfuscations, it is unclear whether they are equally effective to hide a broad range of private information. For instance, blurring a wheelchair in an image might be insufficient to prevent leakage of disability-related information. Alternatively, towards the goal of improving utility, it thus might be beneficial to 'delete-and-replace' selected privacy-sensitive regions by leveraging generative methods (Goodfellow et al., 2014b). To *replace* the an image region while simultaneously preserving global image consistency, recent advances in image inpainting (Pathak et al., 2016) and object removal (Shetty et al., 2018a) might be beneficial.

Privacy beyond personal photos. In Part I of the thesis, we were specifically concerned with privacy leakage across a specific medium: high-resolution personal photos captured using an RGB camera. However, massive amounts of visual data are captured in other forms and using complementary sensors that potentially reveals additional private information. For instance, RGB video sequences can convey information that a single frame cannot e.g., text typed on virtual keyboards (Raguram et al., 2011; Xu et al., 2013), light diffused on a window by a TV (Xu et al., 2014). As another example, by capturing eye-tracking patterns, augmented reality devices can additionally profile an individual's interest when he visits a shopping mall. While some works (Speciale et al., 2019b; Speciale et al., 2019a) have recently started addressing privacy beyond 2d visual data, there is little understanding of leakage in many forms of captured visual data.

Group-level visual privacy. The predominant focus in research, including our approaches in Part I, mitigates privacy leakage on a *record-level* i.e., at the granularity of a single image. However, most scenarios (e.g., social media) involves individuals sharing a *group* of images (e.g., an album of images). Such groups of images further amplifies privacy risks as they reinforce details about the individual. Such details can be used, for instance, to reconstruct 3d face models of individuals to spoof authentication systems (Xu et al., 2016). Furthermore, groups of images additionally capture private 'stories' that would not be possible with a single image, such as the evolution of an individual's social circles or interests over time. As a result, understanding the extent of privacy leakage from a group of images requires further investigation. We can also explore an alternate 'defense' viewpoint when an individual shares a group of images, but by selectively including certain images into the set to fulfill a particular poisoning objective. We further motivate this objective by presenting

two ideas. First, reminiscent to our strategy in Chapter 5, where an individual adds a curated set of 'decoy' images to introduce uncertainty into the individual's true privacy attributes. Second, where the individual carefully perturbs (Shafahi et al., 2018) or watermarks (Uchida et al., 2017) a subset of images before sharing to prevent training, or help attribute training of downstream ML models on the individual's data.

8.2.2 *Privacy-Preserving Collaborative Learning*

In Part II, we evaluated privacy leakage in a collaborative learning scenario i.e., when multiple data owners (each with their own private dataset) intend to train a model without sharing their raw data. In this section, we present some insights to head towards the goal of privacy-preserving collaborative learning settings, and suggest some applications where the techniques might be especially beneficial.

Alternate modes of sharing information. For a set of data owners to collaboratively train a model, it is necessary to communicate (e.g., to server) at least some meaningful information derived from their private datasets. However in this thesis, we observed that sharing such information is problematic: both in its raw form (Part I) as well as from parameters derived from the raw data (Part II). As a result, it would be beneficial studying alternate methods of transferring knowledge (derived from the raw data) from the owner to the server. We believe one potential approach, by extending Liang et al. (2020), is to decompose the trainable parameters of a model into a global (that everyone contributes towards) and a personal (that the owner does not share) parameter set. By sharing only user-invariant global parameter, one could reduce the amount of individual-specific information encoded into the shared parameters. Another promising approach is motivated by our findings in Chapter 6, where the owner could share local knowledge via annotations on a set of publicly-available data. As a result, information shared is over a much lower dimension (e.g., predictions over a small number of classes) compared to parameters (hundreds of thousands for deep models). We believe some recent approaches (Li and Wang, 2019; Chang et al., 2019) that have taken initial steps in realizing this idea will further benefit from our findings in Chapter 6.

Client-sided threats. To complement the server-sided threats explored in Chapter 5, we believe it is equally critical to investigate *client*-sided threats i.e., the owner assuming the adversary's role. The investigation of client-sided threats is especially important given two factors: (i) since FL is typically designed for a massive number ($>$100K) of clients, it exposes a large attack surface for adversaries; and (ii) as FL provides a layer of anonymity and privacy to clients (such as by securely aggregating updates), it is challenging to attribute and exclude malicious devices from participation. Bhagoji et al. (2019) and Bagdasaryan et al. (2020) have made good

steps towards analyzing client-sided threats by studying poisoning and backdooring attacks.

Federated learning for visual privacy. We now present an interesting application domain for federated learning techniques, motivated by privacy concerns around data collection to aid supervised visual privacy approaches (previously discussed in Section 8.2.1). Learning to obfuscate images in an federated learning framework is promising, as the raw sensitive data is never shared, but rather only the minimal representations (model updates) to learn the obfuscation task. However, this presents three key challenges. First, in terms of privacy leakage, it is critical to understand and control unintentional leakage in shared representations. Many recent works (Melis et al., 2019; Zhu et al., 2019; Nasr et al., 2019), including our own in Part II, have taken steps to better understand leakage. Second, in terms of supervision quality, it is unlikely that individuals provide pixel-perfect instance-level annotations of privacy attributes, like the ones presented in our VISPR dataset (Chapter 4). Instead, one could leverage weak supervisory signals, such as in the form of image-level (Joon Oh et al., 2017), key-point (Papadopoulos et al., 2017), or bounding-box (Khoreva et al., 2017) annotations to achieve pixel-level identification of privacy attributes. Third, in terms of communication, standard visual models are parameter-heavy (in the order of millions of trainable parameters), and regularly communicating these bulky parameters over mobile networks is energy-inefficient. Advancements in model compression (Han et al., 2016b; Iandola et al., 2016) and communication-efficiency techniques (Konečnỳ et al., 2016b; Lin et al., 2018) might gradually alleviate such communication concerns. Overall, we find a push in multiple research directions that in symphony enables learning visual privacy approaches without resorting to data collection.

8.2.3 *Knowledge Transfer and Black-box Interactions*

Model stealing and knowledge transfer. We begin by remarking that model stealing studied in Chapter 6 is an instantiation of a more general problem of knowledge transfer (KT). In knowledge transfer (or 'distillation'), the goal is to train (often a more compact) student model to mimic the functionality of a (bulky) teacher model, typically using the teacher's training data. When the teacher's training data is unavailable (e.g., for privacy reasons), a recent line of work proposes zero-shot KT (Micaelli and Storkey, 2019; Nayak et al., 2019) approaches despite the training data unavailability. Such zero-shot KT is reminiscent of model stealing approaches (where the student is the stolen model), but instead with a white-box teacher model (i.e., exact parameters are known). While zero-shot KT and model stealing research run in parallel, we believe they can profit from each other. In one direction, zero-shot KT approaches tend to be sample-inefficient and one could leverage feedback-driven techniques in model stealing approaches, such as our own `adaptive` approach

(Chapter 6) to boost convergence rates. In the other direction, some zero-shot KT approaches (Yoo et al., 2019; Yin et al., 2020) recover the teacher's training data as a stepping stone to perform knowledge transfer; we believe these approaches highlight stealing of training data is possible in addition to functionality. Consequently, going forward in this section, we believe the ideas presented is generally applicable to problems around model stealing as well as (zero-shot) knowledge transfer.

Functional equivalence. In Part III, we modeled an adversary whose primary goal was to extract the 'functionality' (i.e., accuracy on *test* inputs) of the victim model; such a goal highlights the risk to intellectual property of the victim model. However, from a security perspective, it is often more important to extract 'fidelity' (i.e., accuracy on *all* inputs) of the victim model. By extracting fidelity, an adversary recovers a faithful white-box replica of the victim model. The white-box model can be exploited to compute *exact* gradient-information – which is crucial to craft adversarial (Goodfellow et al., 2014a) or poisoning instances (Lowd and Meek, 2005b; Gu et al., 2017) – and thereby target the safety of the victim model. A recent line of work (Jagielski et al., 2020; Carlini et al., 2020) demonstrates preliminary success in recovering fidelity on shallow ReLU fully-connected networks. We believe extending this line of work to more complex models will further improve our understanding on the capabilities of an adversary during model deployment scenarios.

Sample-efficient interactions. Central to black-box adversarial attacks, including model stealing attacks in Part III, is making sample-efficient interactions with the black-box. Such efficient interactions are important as most practical scenarios involve some query overhead, such as in the form of financial costs (for pay-per-prediction APIs) or latency (to thwart models in real-time). Towards the goal of sample-efficient model stealing, we present three interesting research directions. First, assuming a large pool of 'public' input data (e.g., internet images, wiki text sentences) is available to the adversary, developing a *sampling strategy* to select the minimal subset of inputs to query. This is reminiscent of research along active learning (Cohn et al., 1996) approaches, which are designed to select the most promising candidates from a pool of semantically meaningful inputs for human annotation. Similar to model stealing, the promising candidates are inputs that provide the best performance gains when training a (stolen) model. As a result, one could build on top of recent advances in active learning (AL) (Beluch et al., 2018; Gao et al., 2020) methods to improve sample-efficiency. Second, motivated by recent findings (Papernot et al., 2017b; Micaelli and Storkey, 2019; Krishna et al., 2020) which indicate that inputs need not be semantically meaningful when querying the victim black-box, one could instead learn an *efficient generative distribution* to sample from. To aid learning such a distribution, data compression techniques (Wang et al., 2018) provides a good starting point. Third, as shown in ours (Chapter 6) as well as similar works (Krishna et al., 2020), rather than train the stolen model from scratch, it is beneficial to incorporate

some *prior knowledge*. The most popular form to incorporate this knowledge is by good initialization of the 'knockoff' model, such as Imagenet pretrained weights in Chapter 6, or BERT (Devlin et al., 2019) weights in Krishna et al. (2020). In this direction, we believe it is interesting to explore more suitable initialization strategies (Mishkin and Matas, 2015; Finn et al., 2017) for the stolen model to help warm-start the sampling strategy.

Stealing beyond image classification. Recent studies in model stealing, including our own work in Part III, predominantly focus on attacks targeted on an *image classification* model. While these models provide a good starting point, it remains unclear to what extent existing techniques generalize to other deep neural network models, especially in other domains. It is crucial to understand this, as deep models in practical scenarios take many forms (e.g., recommendation systems, machine translation). Specific to understanding effectiveness of stealing attacks targeting natural language models, Krishna et al. (2020) (question answering) and Wallace et al. (2020) (machine translation) have made good first steps.

Out of distribution. Having discussed strategies to further understand capabilities of a model stealing adversary, we now switch focus on defending against such attacks. Common to existing attacks is exploiting the property that the black-box victim model meaningfully responds to *anomalous* inputs, such as inputs outside the training distribution (Chapter 6) or in the extreme case, even randomly-generated patterns (Tramèr et al., 2016). We believe training models to be more capable of recognizing uncertainty in their predictions, especially on out-of-distribution images is a good research direction which potentially benefits defenses against adversarial attacks. By leveraging this insight, Kariyappa and Qureshi (2020) show promising initial results by adaptively perturbing outputs based on prediction uncertainty.

8.2.4 *A Broader Outlook*

In the previous section, we presented interesting research directions to better understand leakage and additionally address certain shortcomings of existing approaches. Now, we take a step back and outline challenges from a broader perspective.

A deeper look at inferring visual privacy attributes. Literature predominantly addresses recognizing private attributes (e.g., faces, political opinions) in images as an object recognition problem. However, while objects can be identified by distinct visual cues, this does not entirely hold true for recognizing privacy attributes. Recognizing many attributes requires associating cues to strong prior knowledge e.g., recognizing location of an individual from an image of the Taj Mahal. In addition, certain attributes might require causal reasoning (Pearl, 2009) e.g., recognizing religion based on a person's presence at a holy site. Effective recognition of privacy

attributes will benefit from breakthroughs in visual scene understanding at a more fundamental level by associating cues with a vast amount of knowledge.

Interplay between extraction and manipulation goals. In this thesis, we were primarily concerned with *extracting* private information unintentionally leaked by content (e.g., images) revealed by a owner. In parallel, a range of works study *manipulating* information prior to sharing it with the owner. Unlike extraction goals which highlights privacy loopholes, manipulation goals exploit safety loopholes by influencing a system to work outside its intended specification. For instance, consider the long line of work on adversarially manipulating images, where imperceptible perturbations (Biggio et al., 2013; Goodfellow et al., 2014a; Koh and Liang, 2017) are introduced targeted to reduce the accuracy of the system. However, we believe both of extraction and manipulation goals are intertwined. Manipulative objectives often require internal knowledge of the target system (e.g., parameters, gradients) and extraction (as studied in Chapter 6) might hold a key. Similarly, extraction performances can be amplified by manipulating the target system into revealing additional information. Understanding the interplay opens up new routes to further understand bounds on capabilities of an adversaries in real-world environments.

PUBLICATIONS

The content in the thesis have previously appeared in the following publications.

[1] *Towards a Visual Privacy Advisor: Understanding and Predicting Privacy Risks in Images.*
Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz.
In International Conference on Computer Vision (**ICCV**), 2017.

[2] *Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images.*
Tribhuvanesh Orekondy, Mario Fritz, and Bernt Schiele.
In IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2018.

[3] *Gradient-Leaks: Understanding Deanonymization in Federated Learning.*
Tribhuvanesh Orekondy, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz.
In the 2nd International Workshop on Federated Learning for Data Privacy and Confidentiality (**FL-NeurIPS**), in conjunction with NeurIPS 2019.
(A longer version of this publication is under submission.)

[4] *Knockoff Nets: Stealing Functionality of Black-Box Models.*
Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz.
In IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2019.

[5] *Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks.*
Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz.
In International Conference on Learning Representations (**ICLR**), 2020.

Tribhuvanesh also contributed in an advisory role to the following publications; their content is not discussed in this thesis.

[6] *Differential Privacy Defenses and Sampling Attacks for Membership Inference.*
Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz.
In the Privacy in Machine Learning workshop (**PriML**), in conjunction with NeurIPS 2019.

[7] *InfoScrub: Towards Attribute Privacy by Targeted Obfuscation.*
Hui-Po Wang, Tribhuvanesh Orekondy, and Mario Fritz.
arXiv, 2020.

[8] *GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators.*
Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz.
In Advances in Neural Information Processing Systems (**NeurIPS**), 2020.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. (2016a). "TensorFlow: A System for Large-Scale Machine Learning." In: *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.

Abadi, Martin, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang (2016b). "Deep learning with differential privacy." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Ahern, Shane, Dean Eckles, Nathaniel S Good, Simon King, Mor Naaman, and Rahul Nair (2007). "Over-exposed? Privacy patterns and considerations in online and mobile photo sharing." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Amon, Mary Jean, Rakibul Hasan, Kurt Hugenberg, Bennett I Bertenthal, and Apu Kapadia (2020). "Influencing Photo Sharing Decisions on Social Media: A Case of Paradoxical Findings." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Ateniese, Giuseppe, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici (2015). "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers." In: *International Journal of Security and Networks* 10.3, pp. 137–150.

Bagdasaryan, Eugene, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov (2020). "How to backdoor federated learning." In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Barnes, Susan B (2006). "A privacy paradox: Social networking in the United States." In: *First Monday*.

Belhumeur, Peter N., João P Hespanha, and David J. Kriegman (1997). "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 19.7, pp. 711–720.

Beluch, William H, Tim Genewein, Andreas Nürnberger, and Jan M Köhler (2018). "The power of ensembles for active learning in image classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin (2003). "A neural probabilistic language model." In: *Journal of Machine Learning Research (JMLR)*.

Berk, Richard A (1983). "An introduction to sample selection bias in sociological data." In: *American sociological review*, pp. 386–398.

Berthelot, David, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel (2019). "Mixmatch: A holistic approach to semi-supervised learning." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Bertran, Martin, Natalia Martinez, Afroditi Papadaki, Qiang Qiu, Miguel Rodrigues, Galen Reeves, and Guillermo Sapiro (2019). "Adversarially Learned Representations for Information Obfuscation and Inference." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Besmer, Andrew and Heather Richter Lipford (2010). "Moving beyond untagging: photo privacy in a tagged world." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Bhagoji, Arjun Nitin, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo (2019). "Analyzing federated learning through an adversarial lens." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Bhagoji, Arjun Nitin, Warren He, Bo Li, and Dawn Song (2017). "Exploring the Space of Black-box Attacks on Deep Neural Networks." In: *arXiv preprint arXiv:1712.09491*.

Bickel, Steffen, Michael Brückner, and Tobias Scheffer (2009). "Discriminative learning under covariate shift." In: *Journal of Machine Learning Research (JMLR)* 10.9.

Biggio, Battista, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli (2013). "Evasion attacks against machine learning at test time." In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

Bitouk, Dmitri, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar (2008). "Face swapping: automatically replacing faces in photographs." In: *ACM SIGGRAPH*.

Bittau, Andrea, Ulfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld (2017). "Prochlo: Strong privacy for analytics in the crowd." In: *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 441–459.

Bonawitz, Keith, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth (2017). "Practical Secure Aggregation for Privacy-Preserving Machine Learning." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Bonawitz, Keith et al. (2019). "Towards Federated Learning at Scale: System Design." In: *SysML*. To appear. URL: https://arxiv.org/abs/1902.01046.

Boult, Terrance Edward (2005). "PICO: Privacy through invertible cryptographic obscuration." In: *Computer Vision for Interactive and Intelligent Environment (CVIIE)*.

Bourdev, Lubomir and Jitendra Malik (2009). "Poselets: Body part detectors trained using 3d human pose annotations." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Brkic, Karla, Ivan Sikiric, Tomislav Hrkac, and Zoran Kalafatic (2017). "I Know That Person: Generative Full Body and Face De-Identification of People in Images." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Bromley, Jane, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah (1994). "Signature verification using a" siamese" time delay neural network." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil (2006). "Model Compression." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.

Carlini, Nicholas, Matthew Jagielski, and Ilya Mironov (2020). "Cryptanalytic Extraction of Neural Network Models." In: *arXiv preprint arXiv:2003.04884*.

Caruana, Rich (1997). "Multitask learning." In: *Machine learning* 28.1, pp. 41–75.

Challenge, Yelp Dataset (2013). *Yelp dataset challenge*.

Chang, Hongyan, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr (2019). "Cronus: Robust and Heterogeneous Collaborative Learning with Black-Box Knowledge Transfer." In: *arXiv preprint arXiv:1912.11279*.

Chapelle, Olivier, Bernhard Schlkopf, and Alexander Zien (2006). *Semi-Supervised Learning*. 1st. The MIT Press. ISBN: 0262514125.

Chattopadhyay, Ankur and Terrance E Boult (2007). "Privacycam: a privacy preserving camera using uclinux on the blackfin dsp." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille (2015). "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chen, Mingqing, Rajiv Mathews, Tom Ouyang, and Françoise Beaufays (2019). "Federated Learning Of Out-Of-Vocabulary Words." In: *arXiv preprint arXiv:1903.10635*.

Chen, Pin-Yu, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh (2017). "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models." In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26.

Chollet, François et al. (2015). *Keras*. `https://keras.io`.

Chouldechova, Alexandra (2017). "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments." In: *Big data* 5.2, pp. 153–163.

Cohn, David A, Zoubin Ghahramani, and Michael I Jordan (1996). "Active learning with statistical models." In: *JAIR*.

Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes (2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data." In: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*.

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Correia-Silva, Jacson Rodrigues, Rodrigo F Berriel, Claudine Badue, Alberto F de Souza, and Thiago Oliveira-Santos (2018). "Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data." In: *IJCNN*.

Crammer, Koby and Yoram Singer (2003). "A family of additive online algorithms for category ranking." In: *Journal of Machine Learning Research (JMLR)* 3.Feb, pp. 1025–1058.

Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Das, Anupam, Martin Degeling, Daniel Smullen, and Norman Sadeh (2018). "Personalized privacy assistants for the internet of things: providing users with notice and choice." In: *IEEE Pervasive Computing* 17.3, pp. 35–46.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Directive, EU (1995). "95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data." In: *Official Journal of the EC* 23.6.

Doersch, Carl, Abhinav Gupta, and Alexei A Efros (2015). "Unsupervised visual representation learning by context prediction." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell (2014). "Decaf: A deep convolutional activation feature for generic visual recognition." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Duchi, John, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra (2008). "Efficient projections onto the l 1-ball for learning in high dimensions." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Dutta, A., A. Gupta, and A. Zissermann (2016). *VGG Image Annotator (VIA)*. `http://www.robots.ox.ac.uk/~vgg/software/via/` Accessed: 2017-11-08. URL: `http://www.robots.ox.ac.uk/~vgg/software/via/`.

Dwork, Cynthia (2006). "Differential Privacy." In: *ICALP*.

Dwork, Cynthia, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel (2012). "Fairness through awareness." In: *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226.

Ebert, Sandra, Mario Fritz, and Bernt Schiele (2012). "Ralf: A reinforced active learning formulation for object class recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

European Union, Council of (2016). *EU General Data Protection Regulation (GDPR): Article 5, Principles relating to processing of personal data*. `https://gdpr-info.eu/art-5-gdpr/`. Accessed: 2020-06-18.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2010a). "The Pascal Visual Object Classes (VOC) Challenge." In: *International Journal of Computer Vision (IJCV)*.

Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman (2010b). "The pascal visual object classes (voc) challenge." In: *International Journal of Computer Vision (IJCV)*.

Eyepacs. `https://www.kaggle.com/c/diabetic-retinopathy-detection`. Accessed: 2018-11-08.

Fadem, Barbara (2012). *Behavioral science in medicine*. Lippincott Williams & Wilkins.

Fergus, Robert, Pietro Perona, and Andrew Zisserman (2003). "Object class recognition by unsupervised scale-invariant learning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fernandes, Earlence, Justin Paupore, Amir Rahmati, Daniel Simionato, Mauro Conti, and Atul Prakash (2016). "Flowfence: Practical data protection for emerging iot application frameworks." In: *25th {USENIX} security symposium ({USENIX} Security 16)*, pp. 531–548.

Finkel, Jenny Rose, Trond Grenager, and Christopher D. Manning (2005). "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling." In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart (2015). "Model inversion attacks that exploit confidence information and basic countermeasures." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Fukushima, Kunihiko and Sei Miyake (1982). "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." In: *Competition and cooperation in neural nets*. Springer, pp. 267–285.

Fulkerson, Brian, Andrea Vedaldi, and Stefano Soatto (2009). "Class segmentation and object localization with superpixel neighborhoods." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Furlanello, Tommaso, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar (2018). "Born again neural networks." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Gallagher, Andrew C and Tsuhan Chen (2008). "Clothing cosegmentation for recognizing people." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gallagher, Andrew C and Tsuhan Chen (2009). "Understanding images of groups of people." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gao, Mingfei, Zizhao Zhang, Guo Yu, Sercan O Arik, Larry S Davis, and Tomas Pfister (2020). "Consistency-Based Semi-Supervised Active Learning: Towards Minimizing Labeling Cost." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

GeoNames Geographical Database. http://www.geonames.org/ Accessed: 2017-11-08. URL: \url{http://www.geonames.org/}.

Geyer, Robin C, Tassilo Klein, and Moin Nabi (2017). "Differentially Private Federated Learning: A Client Level Perspective." In: *NeurIPS PPML Workshop*.

Girshick, Ross (2015). "Fast r-cnn." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Go, Alec, Richa Bhayani, and Lei Huang (2009). "Twitter sentiment classification using distant supervision." In: *CS224N project report, Stanford* 1.12, p. 2009.

Goffman, Erving et al. (1978). *The presentation of self in everyday life*. Harmondsworth London.

Goldreich, Oded (2009). *Foundations of cryptography: volume 2, basic applications*. Cambridge university press.

Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy (2014a). "Explaining and harnessing adversarial examples." In: *arXiv preprint arXiv:1412.6572*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014b). "Generative adversarial nets." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Gould, Stephen, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller (2008). "Multi-class segmentation with relative location prior." In: *International Journal of Computer Vision (IJCV)* 80.3, pp. 300–316.

Griffin, Gregory, Alex Holub, and Pietro Perona (2007). "Caltech-256 object category dataset." In:

Gu, Tianyu, Brendan Dolan-Gavitt, and Siddharth Garg (2017). "Badnets: Identifying vulnerabilities in the machine learning model supply chain." In: *arXiv preprint arXiv:1708.06733*.

Guo, Chuan, Jacob R Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q Weinberger (2019). "Simple black-box adversarial attacks." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Gurari, Danna, Qing Li, Chi Lin, Yinan Zhao, Anhong Guo, Abigale Stangl, and Jeffrey P Bigham (2019). "VizWiz-Priv: A Dataset for Recognizing the Presence and Purpose of Private Visual Information in Images Taken by Blind People." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Halevy, Alon, Peter Norvig, and Fernando Pereira (2009). "The unreasonable effectiveness of data." In: *IEEE Intelligent Systems* 24.2, pp. 8–12.

Han, Song, Huizi Mao, and William J Dally (2016a). "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Han, Song, Huizi Mao, and William J Dally (2016b). "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hard, Andrew, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage (2018). "Federated learning for mobile keyboard prediction." In: *arXiv preprint arXiv:1811.03604*.

Harder, Frederik, Kamil Adamczewski, and Mijung Park (2020). "Differentially Private Mean Embeddings with Random Features (DP-MERF) for Simple & Practical Synthetic Data Generation." In: *arXiv preprint arXiv:2002.11603*.

Hardt, Moritz, Eric Price, and Nati Srebro (2016). "Equality of opportunity in supervised learning." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Hasan, Rakibul, Eman Hassan, Yifang Li, Kelly Caine, David J Crandall, Roberto Hoyle, and Apu Kapadia (2018). "Viewer experience of obscuring scene elements in photos to enhance privacy." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Hasan, Rakibul, Yifang Li, Eman Hassan, Kelly Caine, David J Crandall, Roberto Hoyle, and Apu Kapadia (2019). "Can privacy be satisfying? on improving viewer satisfaction for privacy-enhanced photos using aesthetic transforms." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Hassan, Eman T, Rakibul Hasan, Patrick Shaffer, David Crandall, and Apu Kapadia (2017). "Cartooning for Enhanced Privacy in Lifelogging and Streaming Videos." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

He, Dafang, Xiao Yang, Chen Liang, Zihan Zhou, Alexander G Ororbi, Daniel Kifer, and C Lee Giles (2017a). "Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017b). "Mask r-cnn." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016a). "Deep residual learning for image recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016b). "Deep residual learning for image recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, Ruining and Julian McAuley (2016). "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering." In: *WWW*.

Hendricks, Lisa Anne, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell (2016). "Generating visual explanations." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Hendrycks, Dan, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan (2019). "Augmix: A simple data processing method to improve robustness and uncertainty." In: *arXiv preprint arXiv:1912.02781*.

Hernán, Miguel A, Sonia Hernández-Díaz, and James M Robins (2004). "A structural approach to selection bias." In: *Epidemiology* 15.5, pp. 615–625.

Hewlett, Daniel, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot (2016). "Wikireading: A novel large-scale language understanding task over wikipedia." In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network." In: *arXiv:1503.02531*.

Hitaj, Briland, Giuseppe Ateniese, and Fernando Perez-Cruz (2017). "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory." In: *Neural computation* 9.8, pp. 1735–1780.

Hoffman, Karla Leigh (1981). "A method for globally minimizing concave functions over convex sets." In: *Mathematical Programming* 20.1, pp. 22–32.

House, White (2012). "Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy." In: *White House, Washington, DC*, pp. 1–62.

Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "Mobilenets: Efficient

convolutional neural networks for mobile vision applications." In: *arXiv preprint arXiv:1704.04861*.

Hoyle, Roberto, Luke Stark, Qatrunnada Ismail, David Crandall, Apu Kapadia, and Denise Anthony (2020). "Privacy norms and preferences for photos posted online." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger (2017a). "Densely connected convolutional networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, Gary B., Manu Ramesh, Tamara Berg, and Erik Learned-Miller (2007). *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst.

Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. (2017b). "Speed/accuracy trade-offs for modern convolutional object detectors." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF models for sequence tagging." In: *arXiv preprint arXiv:1508.01991*.

Iandola, Forrest N, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer (2016). "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size." In: *arXiv:1602.07360*.

Ilyas, Andrew, Logan Engstrom, Anish Athalye, and Jessy Lin (2018). "Black-box Adversarial Attacks with Limited Queries and Information." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Ioffe, Sergey and Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Jagielski, Matthew, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot (2020). "High Accuracy and High Fidelity Extraction of Neural Networks." In: *Proceedings of the USENIX Security Symposium (USENIX Security)*.

Jana, Suman, Arvind Narayanan, and Vitaly Shmatikov (2013). "A scanner darkly: Protecting user privacy from perceptual applications." In: *2013 IEEE symposium on security and privacy*. IEEE, pp. 349–363.

Jayaraman, Bargav and David Evans (2019). "Evaluating differentially private machine learning in practice." In: *Proceedings of the USENIX Security Symposium (USENIX Security)*.

Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell (2014). "Caffe: Convolutional architecture for fast feature embedding." In: *Proceedings of the ACM Conference on Multimedia (MM)*.

Joon Oh, Seong, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele (2017). "Exploiting Saliency for Object Segmentation from Image Level Labels." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Joshi, Ajay J, Fatih Porikli, and Nikolaos Papanikolopoulos (2009). "Multi-class active learning for image classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Juuti, Mika, Sebastian Szyller, Alexey Dmitrenko, Samuel Marchal, and N Asokan (2019). "PRADA: protecting against DNN model stealing attacks." In: *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*.

Kae, Andrew, Kihyuk Sohn, Honglak Lee, and Erik Learned-Miller (2013). "Augmenting CRFs with Boltzmann Machine Shape Priors for Image Labeling." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kariyappa, Sanjay and Moinuddin K Qureshi (2020). "Defending Against Model Stealing Attacks with Adaptive Misinformation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Khan, Khalil, Massimo Mauro, and Riccardo Leonardi (2015). "Multi-class semantic segmentation of faces." In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*.

Khoreva, Anna, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele (2017). "Simple does it: Weakly supervised instance and semantic segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, Yoon (2014). "Convolutional neural networks for sentence classification." In: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Knijnenburg, Bart P (2017). "Privacy? I Can't Even! making a case for user-tailored privacy." In: *IEEE Security & Privacy* 15.4, pp. 62–67.

Koh, Pang Wei and Percy Liang (2017). "Understanding black-box predictions via influence functions." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Konečnỳ, Jakub, H Brendan McMahan, Daniel Ramage, and Peter Richtárik (2016a). "Federated optimization: Distributed machine learning for on-device intelligence." In: *arXiv preprint arXiv:1610.02527*.

Konečnỳ, Jakub, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon (2016b). "Federated learning: Strategies for improving communication efficiency." In: *NIPS PPML Workshop*.

Korayem, Mohammed, Robert Templeman, Dennis Chen, David Crandall, and Apu Kapadia (2016). "Enhancing Lifelogging Privacy by Detecting Screens." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Korshunov, Pavel and Touradj Ebrahimi (2013). "PEViD: privacy evaluation video dataset." In: *SPIE*.

Krähenbühl, Philipp and Vladlen Koltun (2011). "Efficient inference in fully connected crfs with gaussian edge potentials." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Krasin, Ivan et al. (2017). "OpenImages: A public dataset for large-scale multi-label and multi-class image classification." In: *Dataset available from https://github.com/openimages*.

Krishna, Kalpesh, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer (2020). "Thieves on Sesame Street! Model Extraction of BERT-based APIs." In: *Proceedings of the International Conference on Learning Representations (ICLR)*. URL: https://openreview.net/forum?id=Byl5NREFDr.

Krizhevsky, Alex and Geoffrey Hinton (2009). *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Kuznetsova, Alina, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. (2018). "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale." In: *arXiv:1811.00982*.

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (2016). "Neural Architectures for Named Entity Recognition." In: *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Lapin, Maksim, Matthias Hein, and Bernt Schiele (2016). "Loss Functions for Top-k Error: Analysis and Insights." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Taesung, Benjamin Edwards, Ian Molloy, and Dong Su (2018). "Defending Against Model Stealing Attacks Using Deceptive Perturbations." In: *S&P Deep Learning and Security (DLS) Workshop*.

Li, Daliang and Junpu Wang (2019). "Fedmd: Heterogenous federated learning via model distillation." In: *arXiv preprint arXiv:1910.03581*.

Li, Hui, Peng Wang, and Chunhua Shen (2017a). "Towards end-to-end text spotting with convolutional recurrent neural networks." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Li, Yi, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei (2017b). "Fully Convolutional Instance-aware Semantic Segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Yifang (2020). "Investigating Obfuscation as a Tool to Enhance Photo Privacy on Social Networks Sites." In:

Li, Yifang, Wyatt Troutman, Bart P Knijnenburg, and Kelly Caine (2018). "Human perceptions of sensitive content in photos." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1590–1596.

Li, Yifang, Nishant Vishwamitra, Hongxin Hu, and Kelly Caine (2020). "Towards A Taxonomy of Content Sensitivity and Sharing Preferences for Photos." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14.

Li, Yifang, Nishant Vishwamitra, Bart P Knijnenburg, Hongxin Hu, and Kelly Caine (2017c). "Blur vs. Block: Investigating the effectiveness of privacy-enhancing obfuscation for images." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Li, Yifang, Nishant Vishwamitra, Bart P Knijnenburg, Hongxin Hu, and Kelly Caine (2017d). "Effectiveness and users' experience of obfuscation as a privacy-enhancing technology for sharing photos." In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Li, Yunpeng, David J Crandall, and Daniel P Huttenlocher (2009). "Landmark classification in large-scale image collections." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Liang, Paul Pu, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency (2020). "Think locally, act globally: Federated learning with local and global representations." In: *arXiv preprint arXiv:2001.01523*.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). "Microsoft COCO: Common objects in context." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Lin, Yujun, Song Han, Huizi Mao, Yu Wang, and William J Dally (2018). "Deep gradient compression: Reducing the communication bandwidth for distributed training." In:

Liu, Bin, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun Aerin Zhang, Norman Sadeh, Yuvraj Agarwal, and Alessandro Acquisti (2016). "Follow my recommendations: A personalized privacy assistant for mobile app permissions." In: *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*, pp. 27–41.

Liu, Yabing, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove (2011). "Analyzing facebook privacy settings: user expectations vs. reality." In: *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*.

Liu, Yanpei, Xinyun Chen, Chang Liu, and Dawn Song (2017). "Delving into transferable adversarial examples and black-box attacks." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (2015). "Deep Learning Face Attributes in the Wild." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Long, Yunhui, Vincent Bindschaedler, and Carl A Gunter (2017). "Towards measuring membership privacy." In: *arXiv preprint arXiv:1712.09136*.

Lowd, Daniel and Christopher Meek (2005a). "Adversarial learning." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.

Lowd, Daniel and Christopher Meek (2005b). "Good Word Attacks on Statistical Spam Filters." In: *CEAS*. Vol. 2005.

Lowe, David G (1999). "Object recognition from local scale-invariant features." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Luo, Canjie, Lianwen Jin, and Zenghui Sun (2019). "Moran: A multi-object rectified attention network for scene text recognition." In: *Pattern Recognition* 90, pp. 109–118.

Ma, Liqian, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz (2018). "Disentangled person image generation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ma, Xuezhe and Eduard Hovy (2016). "End-to-end sequence labeling via bi-directional lstm-cnns-crf." In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu (2018). "Towards deep learning models resistant to adversarial attacks." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

McCallister, Erika (2010). *Guide to protecting the confidentiality of personally identifiable information*. Diane Publishing.

McMahan, Brendan and Daniel Ramage (2017). *Federated Learning: Collaborative Machine Learning without Centralized Training Data*. `https://research.googleblog.com/2017/04/federated-learning-collaborative.html`. Accessed January 21, 2018.

McMahan, H. Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas (2017). "Communication-Efficient Learning of Deep Networks from Decentralized Data." In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

McMahan, H. Brendan, Daniel Ramage, Kunal Talwar, and Li Zhang (2018). "Learning Differentially Private Recurrent Language Models." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Melis, Luca, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov (2019). "Inference Attacks Against Collaborative Learning." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Micaelli, Paul and Amos J Storkey (2019). "Zero-shot knowledge transfer via adversarial belief matching." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Milli, Smitha, Ludwig Schmidt, Anca D Dragan, and Moritz Hardt (2018). "Model reconstruction from model explanations." In: *arXiv preprint arXiv:1807.05185*.

Mishkin, Dmytro and Jiri Matas (2015). "All you need is a good init." In: *arXiv preprint arXiv:1511.06422*.

Narayanan, Arvind and Vitaly Shmatikov (2008). "Robust De-anonymization of Large Sparse Datasets." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Nasr, Milad, Reza Shokri, and Amir Houmansadr (2019). "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Nayak, Gaurav Kumar, Konda Reddy Mopuri, Vaisakh Shaj, R Venkatesh Babu, and Anirban Chakraborty (2019). "Zero-Shot Knowledge Distillation in Deep Networks." In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Neumann, J v (1928). "Zur theorie der gesellschaftsspiele." In: *Mathematische annalen* 100.1, pp. 295–320.

Neumann, Lukáš and Jiří Matas (2012). "Real-time scene text localization and recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nissenbaum, Helen (2009). *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press.

Norberg, Patricia A, Daniel R Horne, and David A Horne (2007). "The privacy paradox: Personal information disclosure intentions versus behaviors." In: *Journal of consumer affairs* 41.1, pp. 100–126.

Noroozi, Mehdi and Paolo Favaro (2016). "Unsupervised learning of visual representations by solving jigsaw puzzles." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Oh, Seong Joon, Max Augustin, Bernt Schiele, and Mario Fritz (2018). "Towards Reverse-Engineering Black-Box Neural Networks." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Oh, Seong Joon, Rodrigo Benenson, Mario Fritz, and Bernt Schiele (2015). "Person Recognition in Personal Photo Collections." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Oh, Seong Joon, Rodrigo Benenson, Mario Fritz, and Bernt Schiele (2016). "Faceless Person Recognition; Privacy Implications in Social Media." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Oh, Seong Joon, Mario Fritz, and Bernt Schiele (2017). "Adversarial Image Perturbation for Privacy Protection – A Game Theory Perspective." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

OpenALPR. `https://github.com/openalpr/openalpr` Accessed: 2017-11-08. URL: `\url{https://github.com/openalpr/openalpr}`.

Orekondy, Tribhuvanesh, Mario Fritz, and Bernt Schiele (2018). "Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images."

In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Orekondy, Tribhuvanesh, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz (2019a). "Gradient-Leaks: Understanding Deanonymization in Federated Learning." In: *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*.

Orekondy, Tribhuvanesh, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz (2020a). "Gradient-Leaks: Understanding and Controlling Deanonymization in Federated Learning." In: *arXiv preprint arXiv:1805.05838*.

Orekondy, Tribhuvanesh, Bernt Schiele, and Mario Fritz (2017). "Towards a Visual Privacy Advisor: Understanding and Predicting Privacy Risks in Images." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Orekondy, Tribhuvanesh, Bernt Schiele, and Mario Fritz (2019b). "Knockoff Nets: Stealing Functionality of Black-Box Models." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Orekondy, Tribhuvanesh, Bernt Schiele, and Mario Fritz (2020b). "Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Papadopoulos, Dim P, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari (2017). "Extreme clicking for efficient object annotation." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Papernot, Nicolas, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar (2017a). "Semi-supervised knowledge transfer for deep learning from private training data." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow (2016). "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." In: *arXiv preprint arXiv:1605.07277*.

Papernot, Nicolas, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami (2017b). "Practical black-box attacks against machine learning." In: *Proceedings of the ACM Asia Conference on Computer and Communications Security (Asia CCS)*.

Papernot, Nicolas, Patrick McDaniel, Arunesh Sinha, and Michael Wellman (2018a). "SoK: Towards the Science of Security and Privacy in Machine Learning." In: *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*.

Papernot, Nicolas, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson (2018b). "Scalable Private Learning with PATE." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Parkhi, Omkar M, Andrea Vedaldi, and Andrew Zisserman (2015). "Deep face recognition." In: *Proceedings of the British Machine Vision Conference (BMVC)*.

Pathak, Deepak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros (2016). "Context encoders: Feature learning by inpainting." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pearl, Judea (2009). *Causality*. Cambridge university press.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research (JMLR)*.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation." In: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*.

Quattoni, Ariadna and Antonio Torralba (2009). "Recognizing indoor scenes." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Raguram, Rahul, Andrew M White, Dibyendusekhar Goswami, Fabian Monrose, and Jan-Michael Frahm (2011). "iSpy: automatic reconstruction of typed input from compromising reflections." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Raval, Nisarg, Ashwin Machanavajjhala, and Landon P Cox (2017). "Protecting Visual Secrets using Adversarial Nets." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). """ Why should I trust you?" Explaining the predictions of any classifier." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1135–1144.

Rios, Luis Miguel and Nikolaos V Sahinidis (2013). "Derivative-free optimization: a review of algorithms and comparison of software implementations." In: *Journal of Global Optimization* 56.3, pp. 1247–1293.

Romero, Adriana, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio (2015). "Fitnets: Hints for thin deep nets." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation." In: *International Conference on Medical image computing and computer-assisted intervention*.

Rousseeuw, Peter J (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." In: *Journal of computational and applied mathematics* 20, pp. 53–65.

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

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge." In: *International Journal of Computer Vision (IJCV)*.

Saenko, Kate, Brian Kulis, Mario Fritz, and Trevor Darrell (2010). "Adapting visual category models to new domains." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Salem, Ahmed, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang (2020). "Updates-leak: Data set inference and reconstruction attacks in online learning." In: *Proceedings of the USENIX Security Symposium (USENIX Security)*.

Salem, Ahmed, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes (2019). "ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models." In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

Samaria, Ferdinando S and Andy C Harter (1994). "Parameterisation of a stochastic model for human face identification." In: *Proceedings of 1994 IEEE workshop on applications of computer vision*, pp. 138–142.

Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen (2018). "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation." In: *arXiv preprint arXiv:1801.04381*.

Schler, Jonathan, Moshe Koppel, Shlomo Argamon, and James W Pennebaker (2006). "Effects of age and gender on blogging." In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Schneiderman, Henry and Takeo Kanade (1998). "Probabilistic modeling of local appearance and spatial relationships for object recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra (2017). "Grad-cam: Visual explanations from deep networks via gradient-based localization." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Settles, Burr and Mark Craven (2008). "An analysis of active learning strategies for sequence labeling tasks." In: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*.

Shafahi, Ali, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein (2018). "Poison frogs! targeted clean-label poisoning attacks on neural networks." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Shao, Ming, Liangyue Li, and Yun Fu (2013). "What do you do? Occupation recognition in a photo via social context." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shetty, Rakshith R, Mario Fritz, and Bernt Schiele (2018a). "Adversarial scene editing: Automatic object removal from weak supervision." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Shetty, Rakshith, Bernt Schiele, and Mario Fritz (2018b). "A4NT: author attribute anonymity by adversarial training of neural machine translation." In: *Proceedings of the USENIX Security Symposium (USENIX Security)*.

Shi, Baoguang, Xiang Bai, and Cong Yao (2016). "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39.11, pp. 2298–2304.

Shokri, Reza, Marco Stronati, Congzheng Song, and Vitaly Shmatikov (2017). "Membership inference attacks against machine learning models." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep inside convolutional networks: Visualising image classification models and saliency maps." In: *arXiv preprint arXiv:1312.6034*.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." In: *arXiv:1409.1556*.

Sinha, Arunesh, Yan Li, and Lujo Bauer (2013). "What you want is not what you get: predicting sharing policies for text-based content on facebook." In: *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, pp. 13–24.

Smith, Virginia, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar (2017). "Federated Multi-Task Learning." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Song, Congzheng and Vitaly Shmatikov (2020). "Overlearning Reveals Sensitive Attributes." In: *Proceedings of the International Conference on Learning Representations (ICLR)*. URL: https://openreview.net/forum?id=SJeNz04tDS.

Speciale, Pablo, Johannes L Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys (2019a). "Privacy preserving image-based localization." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Speciale, Pablo, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys (2019b). "Privacy preserving image queries for camera localization." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting." In: *Journal of Machine Learning Research (JMLR)* 15.1, pp. 1929–1958.

Steil, Julian, Marion Koelle, Wilko Heuten, Susanne Boll, and Andreas Bulling (2019). "Privaceye: privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features." In: *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pp. 1–10.

Stone, Zak, Todd Zickler, and Trevor Darrell (2008). "Autotagging facebook: Social network context improves photo annotation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Sun, Chen, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta (2017a). "Revisiting unreasonable effectiveness of data in deep learning era." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Sun, Qianru, Liqian Ma, Seong Joon Oh, Luc van Gool, Bernt Schiele, and Mario Fritz (2018a). "Natural and Effective Obfuscation by Head Inpainting." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, Qianru, Bernt Schiele, and Mario Fritz (2017b). "A Domain Based Approach to Social Relation Recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, Qianru, Ayush Tewari, Weipeng Xu, Mario Fritz, Christian Theobalt, and Bernt Schiele (2018b). "A hybrid model for identity obfuscation by face replacement." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Sung, F., Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales (2018). "Learning to Compare: Relation Network for Few-Shot Learning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sutton, Charles and Andrew McCallum (2006). "An introduction to conditional random fields for relational learning." In: *Introduction to statistical relational learning* 2, pp. 93–128.

Sutton, Richard S and Andrew G Barto (1998). *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus (2013). "Intriguing properties of neural networks." In: *arXiv preprint arXiv:1312.6199*.

Templeman, Robert, Mohammed Korayem, David J Crandall, and Apu Kapadia (2014). "PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces." In: *NDSS*. Citeseer, pp. 23–26.

Tonge, Ashwini and Cornelia Caragea (2015). "Privacy Prediction of Images Shared on Social Media Sites Using Deep Features." In: *arXiv preprint arXiv:1510.08583*.

Tonge, Ashwini and Cornelia Caragea (2019). "Dynamic deep multi-modal fusion for image privacy prediction." In: *The World Wide Web Conference (WWW)*, pp. 1829–1840.

Torralba, Antonio, Alexei A Efros, et al. (2011). "Unbiased look at dataset bias." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tramèr, Florian, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart (2016). "Stealing Machine Learning Models via Prediction APIs." In: *Proceedings of the USENIX Security Symposium (USENIX Security)*.

Turk, Matthew A and Alex P Pentland (1991). "Face recognition using eigenfaces." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tzeng, Eric, Judy Hoffman, Kate Saenko, and Trevor Darrell (2017). "Adversarial discriminative domain adaptation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Uchida, Yusuke, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh (2017). "Embedding watermarks into deep neural networks." In: *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Verbeek, Jakob and Bill Triggs (2008). "Scene segmentation with conditional random fields learned from partially labeled images." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan (2015). "Show and tell: A neural image caption generator." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vishwamitra, Nishant, Yifang Li, Kevin Wang, Hongxin Hu, Kelly Caine, and Gail-Joon Ahn (2017). "Towards pii-based multiparty access control for photo sharing in online social networks." In: *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies*, pp. 155–166.

Wah, C., S. Branson, P. Welinder, P. Perona, and S. Belongie (2011). *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology.

Wallace, Eric, Mitchell Stern, and Dawn Song (2020). "Imitation Attacks and Defenses for Black-box Machine Translation Systems." In: *arXiv preprint arXiv:2004.15015*.

Wang, Binghui and Neil Zhenqiang Gong (2018). "Stealing hyperparameters in machine learning." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*.

Wang, Gang, Andrew C. Gallagher, Jiebo Luo, and David A. Forsyth (2010). "Seeing People in Social Context: Recognizing People and Social Relationships." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Wang, Jiang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu (2016). "Cnn-rnn: A unified framework for multi-label image classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Kai, Boris Babenko, and Serge Belongie (2011). "End-to-end scene text recognition." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Wang, Tianlu, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez (2019). "Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Wang, Tongzhou, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros (2018). "Dataset distillation." In: *arXiv preprint arXiv:1811.10959*.

Weinberger, Kilian Q, John Blitzer, and Lawrence K Saul (2006). "Distance metric learning for large margin nearest neighbor classification." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Wilber, Michael J, Vitaly Shmatikov, and Serge Belongie (2016). "Can we still avoid automatic face detection?" In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Wisniewski, Pamela J, Bart P Knijnenburg, and Heather Richter Lipford (2017). "Making privacy personal: Profiling social network users to inform privacy education and nudging." In: *International Journal of Human-Computer Studies* 98, pp. 95–108.

Wu, Yue and Prem Natarajan (2017). "Self-organized text detection with minimal post-processing via border learning." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Wu, Zhenyu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin (2018a). "Towards Privacy-Preserving Visual Recognition via Adversarial Training: A Pilot Study." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

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

Xian, Yongqin, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele (2016). "Latent embeddings for zero-shot classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xie, Chulin, Keli Huang, Pin-Yu Chen, and Bo Li (2019). "DBA: Distributed Backdoor Attacks against Federated Learning." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xioufis, Eleftherios Spyromitros, Symeon Papadopoulos, Adrian Popescu, and Yiannis Kompatsiaris (2016). "Personalized Privacy-aware Image Classification." In: *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*.

Xu, Yi, Jan-Michael Frahm, and Fabian Monrose (2014). "Watching the watchers: Automatically inferring tv content from outdoor light effusions." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Xu, Yi, Jared Heinly, Andrew M White, Fabian Monrose, and Jan-Michael Frahm (2013). "Seeing double: Reconstructing obscured typed input from repeated compromising reflections." In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Xu, Yi, True Price, Jan-Michael Frahm, and Fabian Monrose (2016). "Virtual u: Defeating face liveness detection by building virtual models from your public photos." In: *Proceedings of the USENIX Security Symposium (USENIX Security)*.

Yan, Mengjia, Christopher Fletcher, and Josep Torrellas (2018). "Cache telepathy: Leveraging shared resource attacks to learn DNN architectures." In: *arXiv preprint arXiv:1808.04761*.

Yang, Timothy, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays (2018). "Applied federated learning: Improving google keyboard query suggestions." In: *arXiv preprint arXiv:1812.02903*.

Yao, Andrew Chi-Chih (1986). "How to generate and exchange secrets." In: *27th Annual Symposium on Foundations of Computer Science*.

Yin, Dong, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer (2019). "A fourier perspective on model robustness in computer vision." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Yin, Hongxu, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz (2020). "Dreaming to distill: Data-free knowledge transfer via DeepInversion." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yonetani, Ryo, Vishnu Naresh Boddeti, Kris M Kitani, and Yoichi Sato (2017). "Privacy-preserving visual learning using doubly permuted homomorphic encryption." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yoo, Jaemin, Minyong Cho, Taebum Kim, and U Kang (2019). "Knowledge extraction with no observable data." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Yoon, Jinsung, James Jordon, and Mihaela van der Schaar (2019). "PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson (2014). "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zerr, Sergej, Stefan Siersdorfer, Jonathon Hare, and Elena Demidova (2012). "I Know What You Did Last Summer!: Privacy-Aware Image Classification and Search." In: *ACM SIGIR*.

Zhan, Fangneng and Shijian Lu (2019). "Esir: End-to-end scene text recognition via iterative image rectification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Ning, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev (2015). "Beyond Frontal Faces: Improving Person Recognition Using Multiple Cues." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Xinyang, Shouling Ji, and Ting Wang (2018a). "Differentially Private Releasing via Deep Generative Model." In: *arXiv preprint arXiv:1801.01594*.

Zhang, Yang, Mathias Humbert, Tahleen Rahman, Cheng-Te Li, Jun Pang, and Michael Backes (2018b). "Tagvisor: A privacy advisor for sharing hashtags." In: *Proceedings of the 2018 World Wide Web Conference*, pp. 287–296.

Zhang, Yuheng, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song (2020). "The secret revealer: generative model-inversion attacks against deep neural networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, Ligeng, Zhijian Liu, and Song Han (2019). "Deep Leakage from Gradients." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhu, Xiaojin and Andrew B Goldberg (2009). "Introduction to semi-supervised learning." In: *Synthesis lectures on artificial intelligence and machine learning* 3.1, pp. 1–130.