Efficient and **Trustworthy** Gaze Estimation

Lingyu Du

EFFICIENT AND TRUSTWORTHY GAZE ESTIMATION

EFFICIENT AND TRUSTWORTHY GAZE ESTIMATION

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus,
Prof. dr. ir. H. Bijl,
chair of the Board for Doctorates,
to be defended publicly on
Tuesday 7 April 2026 at 15:00 o'clock

by

Lingyu DU

This dissertation has been approved by the (co)promotors.

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Prof. dr. K. G. Langendoen,	Delft University of Technology, promotor
Dr. G. Lan,	Delft University of Technology, copromotor
Independent members:	
Prof. dr. ir. R. L. Lagendijk,	Delft University of Technology
Prof. dr. H. Wen,	University of Warwick, UK
Prof. dr. H. Gellersen,	University of Lancaster, UK
Dr. Ö. Durmaz-Incel,	University of Twente, NL
Prof. dr. ir. F. A. Kuipers,	Delft University of Technology, *reserve member*

The work presented in this dissertation was carried out at the Embedded Systems Group of Delft University of Technology, The Netherlands.

Keywords: Pervasive Computing, Gaze Estimation, Resource Efficiency, Privacy and Security, Trustworthiness

Printed by: Proefschrifetn

Front & Back: Lingyu Du & ChatGPT

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

To Lingyu, ten years ago:
I am fulfilling your dream. I hope you are happy.

CONTENTS

SUMMARY

Eye gaze contains rich information about human attention and cognitive processes. This capability makes the underlying technology, known as gaze estimation, a critical enabler for many applications, ranging from human-computer interaction to cognitive sensing systems. With the development of deep learning, appearance-based gaze estimation has emerged as a promising solution due to its capability of using general-purpose cameras for non-intrusive and cost-effective gaze estimation.

To build applications based on appearance-based gaze estimation, developers can choose among three paradigms. One paradigm is to train gaze estimation models themselves, which allows developers to customize models to meet various application requirements. Another option is to adopt pre-trained gaze estimation models, which avoids the resource-intensive process for model training. The third paradigm is to call gaze estimation services running on the cloud, which are well-suited for developers who wish to reduce the resource consumption for model deployment. In this case, the full-face images of users are sent to the service provider, which returns estimated gaze directions.

Despite these paradigms offering flexible options to developers for building applications, each paradigm comes with distinct challenges that hinder widespread adoption. Training an accurate gaze estimation model requires the availability of large-scale gaze datasets and the adoption of complex neural networks. The former is sparse and difficult to collect, while the latter demands substantial computational resources. Adopting pre-trained models removes the resource burden of model training, but exposes gaze estimation systems to backdoor attacks, in which an adversary can inject a backdoor into the pre-trained model and manipulate its output with a visual trigger after deployment. This compromises the security of many gaze-based applications, e.g., causing the driving assistant system to fail in tracking the driver's attention. Lastly, calling gaze estimation services raises severe privacy concerns. This is because these services often operate as black boxes, leaving users unaware of how their face images that contain sensitive attributes are processed or utilized.

Taking these paradigms together, we observe that they either require substantial resources for model training or raise trustworthiness concerns due to the involvement of third parties. This motivates the main research question of this dissertation: "*How can we make gaze estimation systems both resource-efficient and trustworthy?*" This dissertation answers this question by addressing the challenges associated with each paradigm.

To reduce the resource burden of self-trained models, we present a resource-efficient framework that includes frequency-domain gaze estimation and gaze-aware contrastive learning. The frequency-domain gaze estimation exploits the feature extraction capability and the spectral compaction property of the discrete cosine transform to substantially reduce the computational cost of gaze estimation models. Meanwhile, gaze-aware contrastive learning enables learning gaze representations in an unsupervised manner to overcome the data labeling hurdle. We show that the proposed framework can achieve

comparable gaze estimation performance to existing approaches that rely on a large-scale, well-labeled dataset, while enabling up to 1.67 times speedup in inference latency.

For pre-trained gaze estimation models, we explore solutions to defend against backdoor attacks. We identify the key characteristics that distinguish backdoored gaze estimation models from benign ones, based on which we propose a novel approach to reverse-engineer the backdoor trigger that leads to the identified characteristics. Given a pre-trained model, we use the reverse-engineered trigger to determine whether it is backdoored or not. If it is identified as a compromised model, we further use the reverse-engineered trigger to mitigate its backdoor behavior. We show that the proposed method can defend against various backdoor attacks.

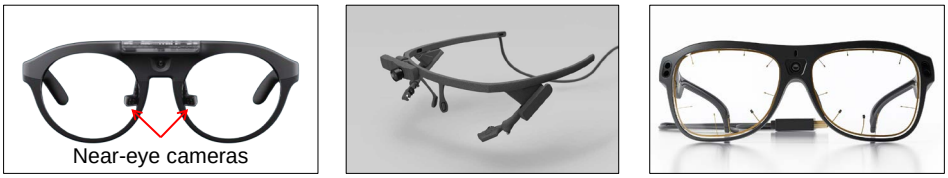
To address privacy concerns in gaze estimation services, we develop a privacy preserver that converts privacy-sensitive full-face images into obfuscated images. The obfuscated versions are then shared with the service provider for gaze estimation. The privacy preserver is designed to generate obfuscated images that exhibit the same facial appearance for different users to protect user privacy, while preserving the gaze features of the raw images to remain effective for accurate gaze estimation. Our experiments show that obfuscated images can effectively protect user privacy while leading to comparable gaze estimation performance to the original images.

Overall, this dissertation contributes to the development of resource-efficient and trustworthy gaze estimation systems. We enhance the resource efficiency of using self-trained models, which typically demand substantial resources, while improving the trustworthiness of the other two paradigms, where the resource burden is offloaded to external parties through the use of pre-trained models or vendor-provided services.

1

Introduction

Every glance carries meaning beyond words.



Pupil Labs Neon | Pupil Labs Core | Tobii Pro Glasses

Figure 1.1: Examples of various wearable eye trackers: Pupil Labs Neon [19], Pupil Labs Core [17], and Tobii Pro Glasses [18].

Human gaze refers to where a person is looking and is a powerful non-verbal cue that conveys human attention. This makes gaze an intuitive modality for interaction, enabling a wide range of human-computer interactive systems. Examples include augmenting perception by presenting information about objects aligned with the user's attention [1], [2], enabling gaze-based gesture control of displays when touch is unavailable or obstructed [3], [4], and assisting drivers according to their gaze behaviors in vehicles [5], [6]. Beyond interaction, gaze has also been extensively used in cognitive sensing systems, owing to the strong correlation between eye movements and the cognitive processes underlying visual perception [7]. Notable applications include the early detection of autism spectrum disorder [8], [9], mental workload estimation [10], [11], and human activity recognition [12], [13]. These diverse applications have motivated extensive studies on the underlying technology, known as *gaze estimation*, in the past decade [14].

1.1. GAZE ESTIMATION SYSTEMS

Existing gaze estimation systems can be broadly classified into eye model-based and appearance-based methods. Eye model-based gaze estimation systems extract features, such as pupil, iris, eye corners, and corneal reflection, from eye images to fit geometric models of eyes to predict gaze direction [7]. These methods typically require dedicated hardware that includes near-eye cameras [15], [16] to capture high-resolution eye images. As shown in Figure 1.1, commercial eye trackers often integrate these near-eye cameras into head-mounted wearable devices, such as smart glasses [17], [18], [19]. However, these commercial eye trackers are expensive, e.g., a Pupil Labs Core eye tracker costs about 3,000 euros, and intrusive, particularly for users wearing prescription glasses, which limits their pervasive adoption for building applications in everyday settings.

By contrast, appearance-based approaches eliminate the need for dedicated hardware and instead infer gaze direction from images captured by pervasive cameras, such as webcams [20] as well as cameras embedded in smartphones [21], tablets [22], and laptops [23]. This makes them non-intrusive, cost-effective, and inherently more suitable for building gaze-based applications usable in daily life. Appearance-based gaze estimation benefits greatly from deep learning techniques, where gaze estimation models represented by deep neural networks are trained to directly map input images to gaze directions. While early works relied solely on eye images cropped from full-face images [24], [25], [26], recent advancements demonstrate that appearance-based methods can greatly benefit from information contained in facial regions and directly use full-

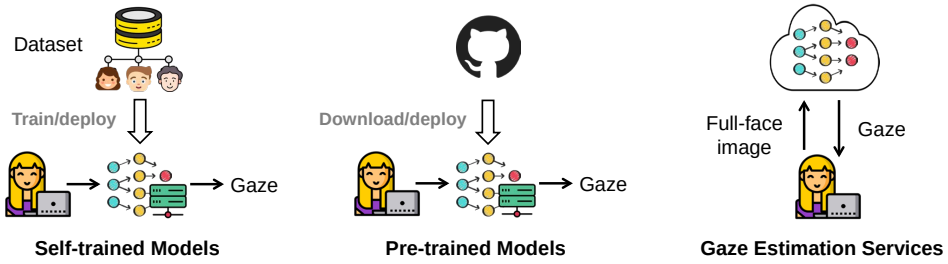


Figure 1.2: Illustration of three paradigms of gaze estimation: self-trained gaze estimation models, pre-trained models, and gaze estimation services.

face images for gaze estimation [21], [27], [28], [29], [30]. Furthermore, the emergence of large-scale full-face image datasets, such as GazeCapture [21] and ETHXGaze [27], has shown that increasing the diversity of the dataset can significantly improve the performance of appearance-based gaze estimation, enabling more accurate gaze prediction in complex environments with various backgrounds and lighting conditions.

1.1.1. GAZE ESTIMATION PARADIGMS FOR DEVELOPING APPLICATIONS

The advantages of appearance-based gaze estimation make it an attractive foundation for developing gaze-based applications. As illustrated in Figure 1.2, developers can adopt one of the following three primary paradigms for gaze estimation according to their needs for the development workflow and the requirements of their applications.

Self-trained gaze estimation models. Developers can train gaze estimation models themselves, which allows them to customize the model architecture and learning objectives to meet diverse application requirements. For example, rather than using a conventional model with a single input encoder and a single gaze direction decoder, the developer can train a gaze estimation model that contains multiple input encoders, shared feature extraction layers, and multiple gaze direction decoders to support cross-device gaze-based applications [31]. However, this paradigm requires substantial resources, particularly large-scale, well-labeled datasets, which are sparse and difficult to collect.

Pre-trained gaze estimation models. Similar to other deep learning-enabled tasks, numerous pre-trained gaze estimation models are readily available on public platforms such as GitHub. Developers can download the pre-trained model and integrate it into the application development. Such a paradigm eliminates the need for resource-intensive model training, but still requires computational resources for deployment. Moreover, using pre-trained models preserves developer control over the gaze estimation pipeline, which is essential for certain applications, e.g., extracting internal representations from the model for eye-contact detection [32].

Gaze estimation services. Thanks to the increasing demand for gaze-based applications, many vendors now provide affordable and easy-to-use gaze estimation services [33], [34], [35], [36], [37], [38], [39]. By utilizing general-purpose cameras, users can access such a gaze estimation service through a cloud server, e.g., RealEye [34] and GazeRe-

corder [38], where users share their full-face images with the service provider, who then takes the image as input and returns the gaze direction. Calling a gaze estimation service for application development is particularly suitable for developers who wish to reduce development effort or resource consumption on the gaze estimation component and do not require control over the gaze estimation process. However, this paradigm may require resources to transmit data from the user side to the service side, leading to additional latency when the data are sent over a wireless channel.

1.2. CHALLENGES

The above-mentioned three paradigms provide developers with flexible options for building gaze-based applications. However, each paradigm also entails inherent challenges that hinder reliable and practical deployment. In the following, we detail the challenges associated with each paradigm.

1.2.1. RESOURCE BURDEN OF SELF-TRAINED MODELS

Training accurate gaze estimation models relies on the availability of large-scale, well-labeled datasets that span diverse gaze directions, facial appearances, and head poses, as well as the use of complex neural networks, both of which demand substantial resources. First, collecting a large-scale and diverse dataset with accurate gaze annotations is a labor-intensive and time-consuming process, as it requires controlled and highly sophisticated setups for data collection and gaze annotation, along with the recruitment of a large number of participants to ensure sufficient data diversity [27]. While publicly available datasets [21], [23] offer valuable resources, their limited subject diversity and environmental coverage make it difficult to train models that generalize well across users, thereby necessitating additional data collection efforts. Second, while a complex neural network is valuable for achieving accurate gaze estimation, it requires substantial computational resources for model deployment. This poses practical challenges for developing applications with constrained time budgets. For example, in real-time gaze estimation used in psychological studies, a high estimation rate is essential for capturing fine-grained cognitive states [40], which in turn demands more computational resources to meet the time constraints.

1.2.2. SECURITY RISKS IN PRE-TRAINED MODELS

While the adoption of pre-trained models avoids the resource burden of model training, it introduces the risk of backdoor attacks, which is particularly relevant to this paradigm due to the reliance on third-party model providers and the opaque training pipelines. In such attacks, an adversary can use an everyday accessory (e.g., glasses or face masks) or a specific facial feature (e.g., scars, freckles, or skin tone) as a backdoor trigger and inject it into a pre-trained model. Once deployed, the adversary can then exploit this trigger to manipulate the model's output: it produces attacker-specified gaze directions when the trigger is present but behaves normally otherwise. Given the important role of gaze estimation in various applications [41], [42], particularly in safety-critical systems [43], backdoor attacks pose serious concerns for safety and reliability. For example, an attack could fool gaze-based driver monitoring systems in autonomous vehicles [6], [44], [45],

causing the system to misjudge the driver's attention and cognitive load [46], [47], [48], fail to issue alerts when the driver is distracted or fatigued, and even indicate the wrong lane in a gaze-based lane-changing assistant [5].

1.2.3. PRIVACY CONCERNS IN COMMERCIAL SERVICES

Typically, gaze estimation services are managed by commercial entities. This makes the gaze estimation pipeline an opaque black box to the users. When querying gaze estimation services, users do not have any knowledge on how their face images are being processed, stored, and utilized. This lack of transparency raises serious privacy concerns, especially given that facial images inherently contain sensitive attributes. Thus, when a malicious service provider has access to a large collection of unprotected face images, it can easily infer sensitive user information beyond the intended purpose. A notable example has been reported by the New York Times [49], where a private company, *Clearview AI*, scraped billions of online photos to build a large-scale facial recognition database without user consent and offered it to other agencies. If this database were misused by malicious parties, they would easily identify people in that database for harmful purposes, e.g., mass surveillance and stalking [50]. This case illustrates the privacy risks that may arise when using gaze estimation services.

1.3. PROBLEM STATEMENT

Taking the three paradigms together, we identify two dimensions that constrain the wide adoption of gaze estimation systems: resource cost and trustworthiness. In this dissertation, we primarily focus on the cost associated with model training and deployment. Specifically, we define resource cost and trustworthiness as follows:

- **Resource cost:** The cost required for data collection and annotation to train a gaze estimation model, as well as the computational resources required for deploying the gaze estimation system.

- **Trustworthiness**: The extent to which gaze estimation systems can be relied upon to behave as expected by users and protect users' data from unauthorized use.

These two dimensions form a spectrum in which a lower resource cost comes with a lower trustworthiness and vice versa. We illustrate this spectrum in Figure 1.3. In the lower-left corner, calling gaze estimation services offloads almost all of the resource burden to an external provider, minimizing the resource cost. However, this paradigm raises the greatest trustworthiness concerns as the whole gaze estimation pipeline is a black box. In the middle region, using pre-trained models removes the need to share facial images with third parties, thereby enhancing trustworthiness, but requires computational resources for model deployment. While it avoids the cost of training, the reliance on externally trained models introduces backdoor vulnerabilities due to the opaque nature of the training process. In the upper-right corner, adopting self-trained gaze estimation models with a self-collected dataset offers the highest level of trustworthiness, as no external parties are involved in building the pipeline. However, this comes at the cost of substantial resource demands for both data collection and deployment. This spectrum

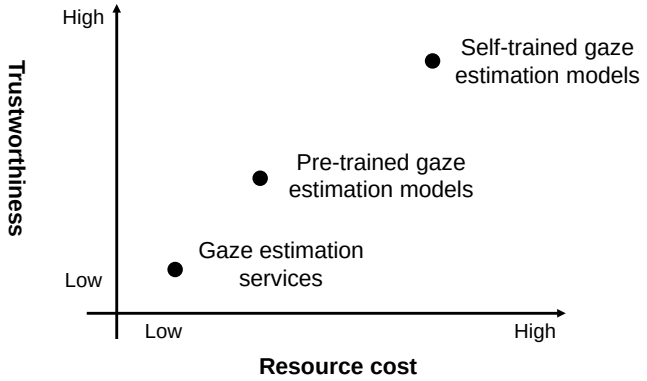


Figure 1.3: Illustration of resource cost and trustworthiness for each paradigm: gaze estimation services, pre-trained gaze estimation models, and self-trained gaze estimation models.

with respect to the resource cost and trustworthiness of gaze estimation systems motivates the **main research question** of this dissertation:

How can we make gaze estimation systems both resource-efficient and trustworthy?

We address this question by considering each of the three paradigms for applying appearance-based gaze estimation. For self-trained models, we focus on improving the resource efficiency of gaze estimation systems, particularly by reducing computational cost and reliance on labeled data. For the other paradigms, where developers offload the resource burden to external parties by either using pre-trained models or vendor-provided services, our focus shifts to enhancing the trustworthiness. Specifically, we formulate the following sub-questions:

Sub-Question 1: How to alleviate the computational burden and overcome the data labeling hurdle for using self-trained gaze estimation models?

Sub-Question 2: How to defend pre-trained models against backdoor attacks?

Sub-Question 3: How to protect user privacy when using gaze estimation services?

1.4. CONTRIBUTIONS AND OUTLINE

This dissertation details how we address each of the three sub-questions. The contributions are outlined below.

Resource-efficient Gaze Estimation - Chapter 2. We present a resource-efficient framework for gaze estimation. At the core of the framework are two components: frequency-domain gaze estimation and gaze-aware contrastive learning. The former dramatically reduces the computational burden, while the latter overcomes the data labeling hurdle.

The frequency-domain gaze estimation component leverages the feature extraction capability of the discrete cosine transform (DCT) [51] and takes several frequency-domain DCT coefficients of the original RGB image as inputs for gaze estimation. Specifically, the

critical content-defining information in an image is concentrated in the low end of the frequency spectrum, while signals in the high-frequency end are mostly trivial and associated with noise [52]. We exploit this spectral compaction property [51] by applying DCT to compress the essential perceptual information of an RGB image into a few low-frequency DCT coefficients. The gaze-aware contrastive learning component leverages unlabeled facial images for gaze representation learning. Although various contrastive learning methods have been proposed for general-purpose visual tasks, such as image classification [53] and object detection [54], we show that they are not directly suitable for gaze estimation. To address this, we introduce several techniques to adapt contrastive learning for unsupervised gaze representation learning. We demonstrate that the proposed framework achieves comparable performance to conventional supervised learning-based solutions in the RGB domain, while significantly reducing the computation burden and the reliance on labeled data.

Defending Gaze Estimation Against Backdoor Attacks - Chapter 3. While countermeasures have been developed to combat backdoor attacks in various classification tasks [55], no solution has been proposed for gaze estimation, which differs as it is a regression task. In response, we present the first solution designed to protect gaze estimation models from such attacks.

We first uncover the fundamental differences between backdoored gaze estimation and classification models to explain why the existing defenses against backdoor attacks are not applicable to gaze estimation models. Then, we identify the characteristics of backdoored gaze estimation models, based on which we introduce a novel approach to reverse-engineer a potential backdoor trigger that can induce the identified characteristics for a backdoored model without access to the training dataset. A pre-trained gaze estimation model is determined as a backdoored model if the reverse-engineering succeeds. For identified backdoor models, we propose a method to mitigate the backdoored behavior by using the reverse-engineered backdoor trigger. The experimental results demonstrate that the proposed method can defend gaze estimation models against various backdoor attacks.

Protecting User Privacy in Gaze Estimation Services - Chapter 4. We introduce the first approach that can effectively protect users' privacy in black-box gaze estimation services. A novel framework is proposed to train a privacy preserver that converts privacy-sensitive full-face images into privacy-enhanced obfuscated images that remain effective for gaze estimation. Then, the obfuscated images are shared with the service provider for the gaze estimation service.

The privacy preserver is designed to achieve two objectives. The first objective is to eliminate features related to the user's private attributes from the original full-face images. To this end, we introduce a novel method that generates an average full-face image from a public dataset and leverages it as a template to transform images of different users, ensuring that the transformed versions exhibit a similar facial appearance akin to the average full-face image. The second objective is to ensure that the essential gaze-related information in the original images is preserved in the obfuscated images to maintain good gaze estimation performance. To achieve this goal, we train a surrogate gaze estimator on a public dataset and leverage the well-trained surrogate gaze estimator

to encourage the privacy preserver to generate obfuscated images that contain features leading to the same gaze direction as the original images. The evaluation results show that the obfuscated images can effectively protect user privacy, i.e., identity and gender, while leading to comparable gaze estimation performance as when using the original images.

Related Publications. Chapters 2, 3, and 4 are based on the following publications:

- **Lingyu Du**, Xucong Zhang, Guohao Lan, *"Resource-efficient Gaze Estimation via Frequency-domain Multi-task Contrastive Learning"*, ACM Transactions on Sensor Networks **(TOSN)**, 2025.
- **Lingyu Du**, Yupei Liu, Jinyuan Jia, Guohao Lan, *"SecureGaze: Defending Gaze Estimation Against Backdoor Attacks"*, In Proceedings of the ACM Conference on Embedded Networked Sensor Systems **(SenSys)**, 2025.
- **Lingyu Du**, Jinyuan Jia, Xucong Zhang, Guohao Lan, *"PrivateGaze: Preserving User Privacy in Black-box Mobile Gaze Tracking Services"*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, **(IMWUT)**, 2024.

Additionally, the following publications are also a result of the doctorate program:

- **Lingyu Du**, Xucong Zhang, Guohao Lan, *"Talk to Me, Not the Slides: A Real-Time Wearable Assistant for Improving Eye Contact in Presentations"*, **(Under Submission)**, 2025.
- Tongyun Yang, Bishwas Regmi, **Lingyu Du**, Andreas Bulling, Xucong Zhang, Guohao Lan, *"Through the Eyes of Emotion: A Multi-faceted Eye Tracking Dataset for Emotion Recognition in Virtual Reality"*, In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, **(IMWUT)**, 2025.
- **Lingyu Du**, Guohao Lan, *"FreeGaze: Resource-efficient Gaze Estimation via Frequency-domain Contrastive Learning"*, Proceedings of International Conference on Embedded Wireless Systems and Networks **(EWSN)**, 2023.

2

RESOURCE-EFFICIENT GAZE ESTIMATION

In this chapter, we focus on alleviating the resource burden of using self-trained gaze estimation models. Specifically, we present EfficientGaze, a resource-efficient framework for gaze estimation. We introduce the frequency-domain gaze estimation, which exploits the feature extraction capability and the spectral compaction property of the discrete cosine transform to substantially reduce the computational cost of gaze estimation models. Moreover, to overcome the data labeling hurdle, we design a novel multi-task gaze-aware contrastive learning framework to learn gaze representations that are generic across subjects in an unsupervised manner. Our evaluation on two gaze estimation datasets demonstrates that EfficientGaze achieves comparable gaze estimation performance to existing supervised learning-based approaches, while enabling up to 6.80 times and 1.67 times speedup in system calibration and gaze estimation, respectively.

2.1. INTRODUCTION

Like many computer vision tasks, existing appearance-based gaze estimation solutions rely on supervised learning to train a complex convolutional neural network (CNN), meaning their performance largely depends on the availability of a large-scale, well-labeled training dataset. Unfortunately, collecting gaze data with accurately annotated labels is a labor-intensive and time-consuming process that requires highly sophisticated setups and subject recruitment [27], [56], [57]. For example, existing gaze datasets are collected using expensive, high-resolution cameras [23], [27], [56], or through dedicated crowdsourcing platforms [21], [57]. Therefore, while complex CNN models are valuable for achieving high gaze-estimation accuracy, there is a need for an efficient and cost-effective solution to train the system without relying on gaze labels.

The adoption of complex neural networks also introduces high system latency during both calibration and gaze estimation stages, which places limitations on the practical use of existing solutions. First, to account for subject diversity on the gaze-estimation performance, existing solutions require subject-specific calibration before the system's first use [7], [14]. This calibration fine-tunes the pre-trained gaze estimation model using a small set of labeled gaze data collected from the targeted subject *on the fly*. When complex neural networks are employed [14], this online calibration becomes time-consuming and negatively impacts the user experience.

Second, gaze estimation-based mobile applications typically have constrained time budgets for real-time gaze estimation. For instance, foveated rendering [58] requires low gaze estimation latency to render a high-quality image precisely in the user's foveal vision, even when the user is making fast eye movements such as saccades [59]. Similarly, gaze estimation-based psychological applications [40] require a high tracking rate to capture fine-grained cognitive contexts. Thus, in practical scenarios, gaze estimation systems often face the challenge of maintaining low inference latency.

Our solution. We present EfficientGaze, a resource-efficient gaze estimation framework that can alleviate the computational burden and overcome the data labeling hurdle of existing supervised learning-based counterparts. These capabilities are made possible by a suite of novel techniques devised in this work.

First, existing gaze estimation systems use RGB images as inputs. To reduce the system latency, a straightforward approach is to aggressively reduce the input size of the neural networks. However, as we will validate by experiments in Section 2.6, simply compressing by down-sampling the RGB image destroys the perceptual information it contains, leading to poor estimation performance. To resolve this challenge, we devise the frequency-domain gaze estimation (Section 2.4). This method leverages the feature extraction capability of the discrete cosine transform (DCT) [51] and takes the frequency-domain DCT coefficients of the original RGB image as inputs for gaze estimation. Specifically, the critical content-defining information in an image is concentrated in the low end of the frequency spectrum, while signals in the high-frequency end are mostly trivial and associated with noise [52]. We exploit this spectral compaction property [51] by applying DCT to compress the essential perceptual information of an RGB image into a few low-frequency DCT coefficients. As demonstrated in Section 2.6, compared to conventional RGB-based solutions, the proposed frequency-domain gaze estimation achieves up to 6.80 and 1.67 times speedup in calibration and gaze estimation,

respectively, while maintaining the similar gaze estimation performance.

Second, to overcome the data labeling challenge faced by existing supervised gaze estimation systems [16], [21], [23], [27], [31], [60], we propose a contrastive learning (CL)-based framework (Section 2.5) that leverages unlabeled facial images for gaze representation learning. Although various CL-based unsupervised learning methods have been proposed for general-purpose visual tasks, such as image classification [53], [61] and object detection [54], they are not directly suitable for gaze estimation. These methods typically aim to learn an embedding network that ensures the representations of visually similar images, i.e., images containing the same instance or instances of the same category [62], are close to each other in the embedding space [63], while the representations of visually distinct images are apart from each other [64]. This focus on appearance-related semantic features benefits mainstream classification tasks [30] but does not accommodate the unique needs of gaze estimation, where images with the same gaze label can be visually distinct (e.g., facial images of different subjects sharing the same gaze). As demonstrated in Section 2.6, conventional CL approaches, i.e., SimCLR [53], lead to poor performance in gaze representation learning and high gaze estimation error. To address this, we introduce several optimizations for contrastive gaze representation learning. First, we devise the gaze-specific data augmentation (Section 2.5.2) to ensure the gaze features are preserved in the augmented images. Second, we design the subject-conditional projection (Section 2.5.3) along with subject-specific gaze-aware contrastive loss (Section 2.5.4) to learn gaze-aware features that are generic across different subjects. Our major contributions are summarized as follows:

• We introduce the frequency-domain gaze estimation, which exploits the feature extraction capability and spectral compaction property of the discrete cosine transform to aggressively compress the inputs of gaze estimation systems and significantly reduce system latency. To the best of our knowledge, this is the first work that utilizes these beneficial properties of DCT for gaze estimation.

• We propose a contrastive gaze representations learning framework that overcomes the data labeling challenges of existing supervised learning-based methods. We devise the gaze-specific data augmentation to preserve gaze-related features during the unsupervised learning process. Moreover, we design the subject-conditional projection and the subject-specific gaze-aware contrastive loss to learn gaze representations that are invariant to subject differences.

• We conduct a comprehensive evaluation of EfficientGaze on two state-of-the-art gaze estimation datasets, i.e., ETHXGaze [27] and MPIIFaceGaze [28]. The results demonstrate the effectiveness of EfficientGaze. Compared to the conventional CL-based method, EfficientGaze reduces the angular error by 4.7° and 5.9° on average across the two datasets, respectively. When compared to the RGB supervised learning-based method, which leverages 37,000 and 30,000 labeled images for training, EfficientGaze achieves comparable results with an average performance gap of only 1.1° and 0.9° across the two datasets, while enabling up to 6.80 and 1.67 times speedup in calibration and gaze estimation, respectively.

Chapter roadmap. The rest of this chapter is organized as follows. In Section 2.2, we pro-

vide a review of related work and highlight the research gaps. We present the overview of EfficientGaze in Section 2.3. The design details of frequency-domain gaze estimation and frequency-domain multi-task gaze-aware contrastive learning are given in Sections 2.4 and 2.5, respectively. We evaluate the proposed system in Section 2.6, followed by the conclusion in Section 2.7. The implementation of EfficientGaze is publicly available at `https://github.com/FreeGaze/EfficientGaze`.

2.2. RELATED WORK

Below, we review related works in unsupervised representation learning, deep learning in the frequency domain, and multi-task learning. We also highlight the gaps that we aim to address in this work.

2.2.1. UNSUPERVISED REPRESENTATION LEARNING

Our work is related to existing efforts in unsupervised representation learning, which can be roughly categorized into generative and discriminative methods. Generative methods usually take the form of auto-encoder [65] or generative adversarial network [66], and learn the embedding network by reconstructing the original high-dimensional inputs from the low-dimensional latent representations. However, generative methods perform pixel-level image reconstruction that is computationally expensive [67]. By contrast, discriminative methods learn representations by solving different pretext tasks, such as recognizing the rotation of the image [68], solving jigsaw puzzles [69], and contrastive prediction task [64]. Among them, contrastive prediction task-based solutions currently achieve state-of-the-art performance in unsupervised representation learning for a wide range of tasks, such as image classification [53], [61] and object detection [54]. However, as discussed in Section 2.6, conventional contrastive learning approaches focus on learning general representations that are more related to the appearance and the identity of the subjects, which leads to poor performance on gaze estimation. By contrast, EfficientGaze encourages the learning of gaze-related features to improve gaze estimation performance.

There are also works on unsupervised gaze representation learning. Yu et al. [70] proposed a method to learn low dimensional gaze representations by jointly training a gaze redirection network [71] and a gaze representation network. However, the training of the gaze redirection network requires well-aligned eye images, setting constraints on their approach. More recently, by constructing eye-consistent and gaze-similar image pairs, Sun et al. [72] proposed the cross-encoder to learn gaze representations from unlabeled eye images. These two solutions rely on generative models, i.e., the auto-encoder [65], for representation learning, which requires the computationally expensive pixel-level image reconstruction. By contrast, EfficientGaze builds upon contrastive learning to avoid the reconstruction process.

2.2.2. LEARNING IN THE FREQUENCY DOMAIN

EfficientGaze is also related to existing efforts in using compressed frequency-domain representations for image-based learning tasks [73], [74], [75], [76]. As an example, Delac et al. [73] investigated the feasibility of using the DCT and the discrete wavelet transform

(DWT) coefficients in the JPEG and JPEG2000 compressed domain for face recognition. Their results indicate that the compressed frequency-domain features can improve the robustness of face recognition and reduce the computational time. In the deep learning regime, Ghosh and Chellappa [75] demonstrated that by leveraging the feature extraction capability of DCT and incorporating DCT operation on the feature maps generated by the convolutional layers, one can speed up the network convergence by three times. In a different direction, Gueguen et al. [77] suggested that when using the DCT coefficients as the inputs, the first few layers of a ResNet-50-based image recognition network can be pruned to improve computational efficiency by 1.77 times. Distinct from existing works, we exploit the feature extraction capability and the spectral compaction property of DCT for aggressive input compression. We are also the first to investigate unsupervised learning in the DCT frequency domain for gaze estimation, which is more challenging than the image recognition task studied by the existing works.

2.2.3. MULTI-TASK LEARNING

Multi-task learning has been leveraged to learn features that are generic across multiple tasks in various research areas, such as gaze estimation [31], [78], human activity recognition [79], [80], [81], visual tracking [82], and face recognition [83]. The widely used architectures for multi-task learning consist of layers shared across different tasks and multiple task-specific layers. For example, Zhang et al. [31] propose learning shared gaze features that can be used to estimate gaze direction from images captured by different devices, such as tablets and laptops, by training shared feature extraction layers and device-specific encoders and regressors. Hao et al. [79] train a shared feature extractor and multiple subject-specific classifiers to learn features invariant to subject diversities for activity recognition from data obtained by an inertial measurement unit.

There are also works that introduce contrastive learning into multi-task learning by using the contrastive prediction task as an auxiliary task to improve performance on the considered tasks, such as masked face recognition [83] and diagnosis of COVID-19 [84]. Furthermore, Lin et al. [85] solve multiple self-supervised tasks, i.e., motion prediction, jigsaw puzzle, and contrastive prediction task, to learn general features for skeleton-based action recognition. Different from existing works, EfficientGaze solves multiple subject-specific gaze-aware contrastive prediction tasks to learn gaze features that are invariant to subject differences without access to gaze labels.

2.3. SYSTEM OVERVIEW

In this chapter, we present EfficientGaze, a resource-efficient framework for gaze estimation. At the core of EfficientGaze are *frequency-domain gaze estimation* and *gaze-aware contrastive learning*. The former dramatically reduces computational burden, ensuring low latency in both system calibration and inference. The latter overcomes the data labeling hurdle faced by existing supervised learning-based counterparts by learning generic gaze representations without requiring gaze labels.

The overview design of EfficientGaze is shown in Figure 2.1, which includes three stages: the self-supervised pre-training stage, the supervised calibration stage, and the deployment stage. In the self-supervised pre-training stage, EfficientGaze takes unla-

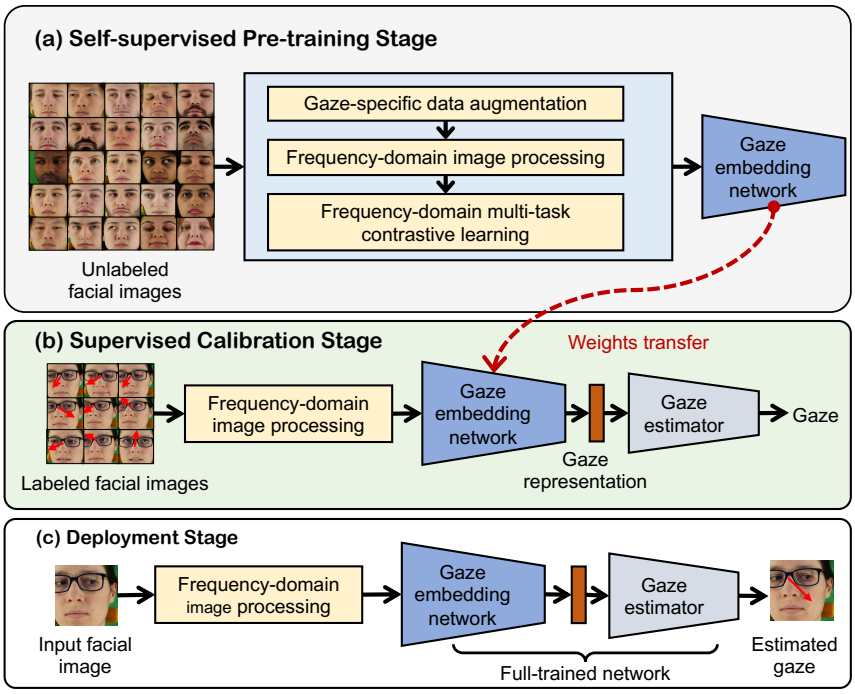


Figure 2.1: Overview of EfficientGaze that incorporates: (a) the self-supervised pre-training stage; (b) supervised calibration stage; and (c) deployment stage.

beled facial images as inputs to pre-train a gaze embedding network for unsupervised gaze representation learning. Next, the pre-trained gaze embedding network is transferred to the supervised calibration stage. Specifically, by using a small number of labeled facial images from the targeted subject as input, we first leverage the frequency-domain image processing module to obtain the selected DCT coefficients of the original RGB images. Then, we fine-tune the pre-trained gaze embedding network and the randomly initialized gaze estimator for subject-specific gaze estimation. Finally, in the deployment stage, the fine-tuned gaze embedding network and the gaze estimator are used for run-time gaze estimation.

2.4. FREQUENCY-DOMAIN GAZE ESTIMATION

To reduce the latency in both calibration and inference, we propose frequency-domain gaze estimation, which uses the selected DCT coefficients of the original RGB images as inputs for gaze estimation. Below, we first investigate the impact of the input shape on the time complexity of the CNN. Then, we present the details of the proposed frequency-domain image processing, which transforms the image from the RGB color space to the DCT frequency domain. Finally, we describe the architecture of the neural networks that can take the image in the frequency domain as inputs.

2.4.1. RATIONALE

The computational cost of a convolutional layer is largely influenced by its FLOPs [86]. For a given CNN with a stack of convolutional layers, the FLOPs of the l-th convolutional layer, denoted as $F_l(I)$, can be calculated by [87]:

$$F_l(I) = n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2, \tag{2.1}$$

where I is the original input, n_{l-1} is the number of input channels of the l-th layer, s_l^2 is the spatial size of the filter, i.e., the size of the first two dimensions, n_l is the number of filters in the l-th layer, and m_l^2 is the spatial size of the output feature map. We use $O_l(I) = n_l \cdot m_l^2$ to denote the output size of the l-th layer.

As a toy example to motivate our design, we consider two network inputs with **the same size but different shapes**: input I_1 has the shape of $224 \times 224 \times 3$, while input I_2 has the shape of $28 \times 28 \times 192$. The spatial size of I_1 is 64 times that of I_2, while the number of channels of I_2 is 64 times that of I_1. We assume the first convolutional layer of the CNN has 64 filters with a filter spatial size of 3×3. We also assume that the stride of the convolutional layer is one and padding is used.

For the first convolutional layer, its output shape is $224 \times 224 \times 64$ and $28 \times 28 \times 64$, for I_1 and I_2, respectively. Based on Equation (2.1), I_1 and I_2 lead to the same FLOPs for the first convolutional layer as:

$$F_1(I_1) = 3 \cdot 3^2 \cdot 64 \cdot 224^2 = F_1(I_2) = 192 \cdot 3^2 \cdot 64 \cdot 28^2. \tag{2.2}$$

For the l-th convolutional layer when $l \geq 2$, we can further rewrite Equations (2.1):

$$F_l(I) = n_{l-1} \cdot s_l^2 \cdot O_l(I), \tag{2.3}$$

From the second convolutional layer onward, the values of s_l, n_{l-1}, and n_l are independent of the original network input I. Thus, as shown in Equations (2.3), we can consider $F_l(I)$ as functions of $O_l(I)$. Given $O_1(I_1)$ is 64 times of $O_1(I_2)$ in our example, we have $F_l(I_1) \approx 64 \cdot F_l(I_2)$ for any convolutional layer with $l \geq 2$. Note that the factor 64 comes from the ratio of the spatial size of the inputs.

Based on the above analysis, we can conclude that when $l \geq 2$, the FLOPs of the convolutional layer are no longer influenced by the channel number of the original input, but remain dependent on its spatial size, which determines the spatial size of the intermediate feature maps. This motivates us to reduce the spatial size of the input to lower the computational cost, even at the cost of increasing its channel dimensionality. In addition, if we can also reduce the number of channels of the input, the consumption can be further reduced.

2.4.2. DESIGN OF FREQUENCY-DOMAIN IMAGE PROCESSING

Apparently, simply reshaping and compressing the original RGB input will disrupt the spatial organization of the image and destroy the perceptual information it contains. To resolve this challenge, we leverage the discrete cosine transform (DCT) to convert inputs from the RGB domain to the frequency domain. We leverage the frequency-domain DCT coefficients as inputs for gaze estimation, which allows us to reshape the original input without losing the essential perceptual information.

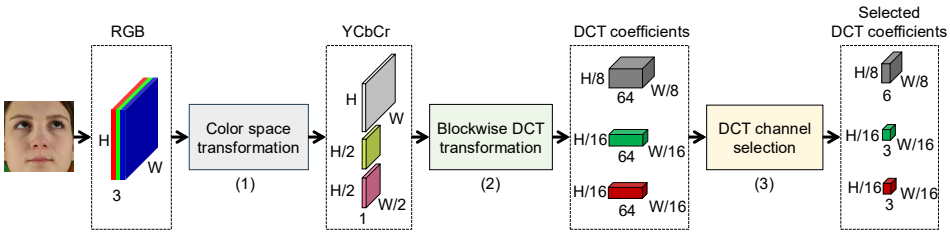


Figure 2.2: The pipeline of the proposed frequency-domain image processing. (1) The input RGB facial image is first converted to the YCbCr color space with the chroma components downsampled. (2) Then, we apply the blockwise DCT transformation on the three components to obtain the corresponding DCT coefficients matrix. (3) Finally, we perform the DCT channel selection to retain only the essential DCT coefficients as the input for gaze estimation.

Moreover, the spectral compaction [51] property of the DCT transformation suggests that most of the critical content-defining information of an image is concentrated in the low-end of the frequency spectrum, whereas signals in the high-frequency end are mostly trivial and are associated with noise [52]. Therefore, we can leverage this property to significantly compact the essential perceptual information of a facial image into a few DCT coefficients in the low-frequency domain. In fact, this property has been widely utilized in data compression techniques, such as the JPEG compression standard [88], where the quantization process rounds off most of the high-frequency components to reduce data size while preserving visual quality [89].

Color space transformation. The pipeline of the proposed frequency-domain image processing is shown in Figure 2.2. It takes an RGB facial image with the shape $W \times H \times 3$ as the input, where W and H denote the width and height, respectively. First, we apply color space transformation to convert the input image from the RGB color space to the YCbCr color space. This transformation results in three components: one luma component (Y), representing the brightness, and two chroma components (Cb and Cr), representing the chrominance of the image. As shown in Figure 2.2, the spatial resolution of the Cb and Cr components is reduced by a factor of two. This reduction is based on the fact that the human visual system is more sensitive to fine-grained brightness details (luma) than to hue and color details [77]. Thus, compressing the two chroma components can be done without significant loss of perceptual quality [88], [90].

Blockwise discrete cosine transformation. After the color space transformation, each of the Y, Cb, and Cr components is partitioned into multiple 8×8 rectangular nonoverlapping blocks. We then apply the blockwise discrete cosine transform (DCT) to obtain their frequency domain representations, i.e., the DCT coefficients. Specifically, for a block $\mathbf{B} \in \mathbb{R}^{8\times 8}$, the corresponding DCT coefficients are denoted by matrix $\mathcal{B} \in \mathbb{R}^{8\times 8}$, whose element $\mathcal{B}_{i,j}$ is obtained by [91]:

$$\mathcal{B}_{i,j} = s(i)s(j)\sum_{n=0}^{7}\sum_{m=0}^{7}\mathbf{B}_{n,m}\cos\left[\frac{\pi}{8}\left(n+\frac{1}{2}\right)i\right]\cos\left[\frac{\pi}{8}\left(m+\frac{1}{2}\right)j\right], \tag{2.4}$$

where $\mathbf{B}_{n,m}$ is the pixel value at coordinate (n,m); integers $0 \le i \le 7$ and $0 \le j \le 7$ represent the horizontal and the vertical frequency, respectively; $s(i)$ is a normalization factor

ensuring transformation orthonormal, for which $s(i) = (\frac{1}{8})^{0.5}$ if $i = 0$ and $s(i) = 0.5$ otherwise. In sum, each block **B** is transformed into the frequency domain represented by a weighted combination of 64 orthogonal sinusoids, where $\mathcal{B}_{i,j}$ is DCT coefficient indicating the spectral energy.

After the DCT transformation, we scan each of the 8×8 DCT coefficients matrices in a zigzag order starting from the top-left corner and subsequently convert it to a 1×64 vector. As shown in Figure 2.2, the outputs of the DCT transformation are three coefficients matrices, with shapes of $\frac{W}{8} \times \frac{H}{8} \times 64$, $\frac{W}{16} \times \frac{H}{16} \times 64$, and $\frac{W}{16} \times \frac{H}{16} \times 64$, for the Y, Cb, and Cr components, respectively.

DCT channel selection. After applying the blockwise discrete cosine transform (DCT) to the YCbCr components of the RGB image and obtaining the DCT coefficients matrices, we perform DCT channel selection to further compress these matrices while retaining essential information for gaze estimation. This step leverages the spectral compaction property of DCT, which concentrates significant image information in the lower-frequency coefficients.

As shown in Figure 2.2, instead of keeping all 64 channels of the DCT coefficients, we selectively retain a subset. Specifically, we preserve the lowest six channels for the Y component and the lowest three channels each for the Cb and Cr components. This selection process prunes away higher-frequency channels, which contain less perceptually important information. Taking an RGB image with the size of $224 \times 224 \times 3$ as the input, the frequency domain image processing outputs the selected DCT coefficients matrix of Y, Cb, and Cr, with the shape of $28 \times 28 \times 6$, $14 \times 14 \times 3$ and $14 \times 14 \times 3$, respectively. The selected DCT coefficient matrices are then fed into the gaze embedding network for representation learning.

2.4.3. GAZE EMBEDDING NETWORK

Figure 2.3 shows the architecture of the gaze embedding network, which is based on ResNet-18 [92]. Since the DCT coefficients matrix of the Y component has a larger spatial size compared to that of the Cb/Cr components, we cannot feed the three DCT coefficient matrices directly to a conventional CNN. A naive solution is to downsample the input of the Y component or upsample the input of the Cb/Cr components to align the spatial sizes. However, such methods can lead to information loss or increased computational complexity. Thus, we have to design the CNN network carefully so that it can take the three matrices with different sizes as inputs. In our current design, we adopt the late concatenate [77], which has proved to ensure better learning performance.

As shown in Figure 2.3, the DCT coefficients matrix of the Y component is first passed through two residual blocks consisting of four convolutional layers, whose output shape is $14 \times 14 \times 128$. In parallel, the DCT coefficients matrices of Cb/Cr are first concatenated and then fed into a single convolutional layer, for which the output shape is $14 \times 14 \times 128$. Then, the outputs from these two parallel paths are concatenated as they have the same spatial size. The joint representations are then fed into two residual blocks equivalent to block conv5 in the original ResNet-18 architecture. The final output of the gaze embedding network is the gaze representation vector containing 512 features. It is important to note that the architecture shown in Figure 2.3 consists of eight convolutional layers in the deepest path, which is eight layers shallower than the original ResNet-18. We adopt

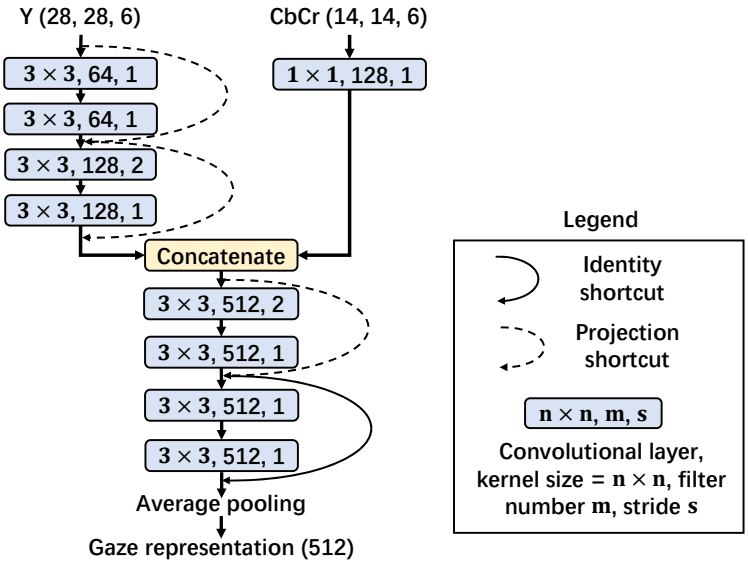


Figure 2.3: The design of the gaze embedding network.

the current design after balancing the gaze-estimation accuracy and system latency. We provide a detailed investigation of this trade-off in Section 2.6.5.

2.5. GAZE-AWARE CONTRASTIVE LEARNING

Inspired by recent advances in contrastive visual representation learning [53], [61] and multi-task learning [31], [79], we propose a novel approach termed *frequency-domain multi-task gaze-aware contrastive learning*. This method utilizes unlabeled facial images to train the gaze embedding network. The framework of our proposed method is shown in Figure 2.4.

We first introduce the *gaze-specific data augmentation* (referred to as $Aug(\cdot)$ in Section 2.5.2) to generate positive and negative image pairs. These augmented images are then transformed from the RGB color space to the DCT frequency domain using our devised *frequency-domain image processing* (denoted as $Freq(\cdot)$). Subsequently, these augmented images in the frequency domain are fed into the gaze embedding network $f(\cdot)$, which is shared across different subjects. This network learns gaze representations that are invariant to subject variations.

To encode subject-specific features, we employ the multi-task learning framework by considering solving gaze-aware contrastive loss on each subject as a task. This approach further maps the generic gaze representations in the general representation space $\mathbb{GP}$ to subject-specific embedding spaces $\mathbb{SP}_i$ tailored for each subject i, through the proposed *subject-conditional projection* (referred to as $S(\cdot)$, in Section 2.5.3). To encourage the shared gaze embedding network to learn discriminative gaze representations, we design the *subject-specific gaze-aware contrastive loss* (in Section 2.5.4) within each

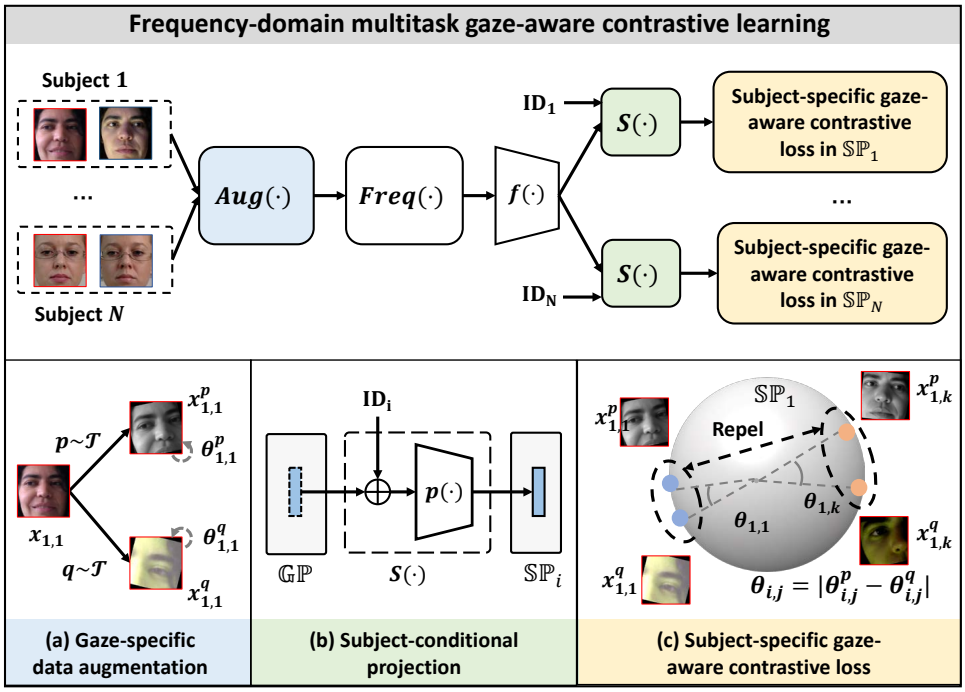


Figure 2.4: The framework of the proposed frequency-domain multi-task gaze-aware contrastive learning. Taking unlabeled full-face images as inputs, we apply gaze-specific data augmentation $Aug(\cdot)$ to generate augmented images. Subsequently, these augmented images undergo frequency-domain image processing $Freq(\cdot)$ to extract selected DCT coefficient matrices. These matrices are then fed into the gaze embedding network $f(\cdot)$, which learns generic gaze representations within the general representation space $\mathbb{GP}$. To accommodate the subject-specific attributes, we leverage the multi-task learning framework by considering solving the gaze-aware contrastive loss for each subject as a task. Specifically, we introduce the subject-conditional projection $S(\cdot)$ to map the general representations to distinct subject-specific embedding space $\mathbb{SP}_i$, utilizing a trainable identity embedding $\mathbf{ID}_i$ specific to each subject i. Additionally, we propose the subject-specific gaze-aware contrastive loss within each subject-specific embedding space $\mathbb{SP}_i$ to encourage the learning of generic gaze representations.

subject-specific embedding space to train the end-to-end framework.

Below, we first briefly introduce the pipeline of contrastive learning. We then delve into the details of each component outlined above.

2.5.1. PIPELINE OF CONVENTIONAL CONTRASTIVE LEARNING

The idea of conventional contrastive learning [53], [93], [94] is to pull the representations of visually similar images together while separating those of visually dissimilar images. Specifically, given a set of images $\{x_i\}_{i=1}^{N}$, the conventional contrastive learning first applies data augmentation to transform each image x_i into two correlated views (x_i^p, x_i^q). The views transformed from the same image are considered as a visually similar image pair, known as a positive image pair, while the views augmented from different images are considered as a visually dissimilar image pair, called a negative image pair. Then, the embedding network maps the augmented images to a general representation space $\mathbb{GP}$. After that, a nonlinear projection further maps the representations into the embedding space $\mathbb{SP}$. We use z_i^p and z_i^q to denote the embeddings of x_i^p and x_i^q respectively. The embedding network and the nonlinear projection are jointly trained to solve the contrastive prediction task by minimizing the following contrastive loss function:

$$\mathcal{L} = -\frac{1}{K}\sum_{i=1}^{K}\log\frac{\exp\left(\text{sim}(\mathbf{z}_i^p, \mathbf{z}_i^q)/\tau\right)}{\sum_{k=1}^{K}\exp\left(\text{sim}(\mathbf{z}_i^p, \mathbf{z}_k^q)/\tau\right)}, \tag{2.5}$$

where $\text{sim}(u, v) = u^\top v/\|u\|\|v\|$ is the cosine similarity of two feature vectors u and v; K is the number of images in the minibatch; and τ is a temperature parameter.

More specifically, solving the contrastive prediction task requires: 1) increasing the representation similarity of positive image pairs, which helps the embedding network to learn representations that are invariant to the applied data augmentation; and 2) reducing the representation similarity of negative image pairs, which enforces the embedding network to learn representations that can differentiate each image in the minibatch.

2.5.2. GAZE-SPECIFIC DATA AUGMENTATION

We adopt three data augmentation operators in our design, i.e., gaze-cropping and resizing, color distortion, and image rotation, and apply them sequentially on the original images to generate the augmented images. Specifically, gaze-cropping and resizing and color distortion can maintain the gaze-related semantic features of the original images, while the image rotation will change them. We train the gaze embedding network to learn gaze representations that are invariant to gaze-cropping and resize and color distortion but are aware of image rotation.

Gaze-cropping and resizing. In the context of gaze estimation, we argue that the augmented images should contain gaze-related semantic features. Although existing research [53] showed that random cropping and resizing is an effective data augmentation operator for learning good representations, it may completely remove the gaze-related ocular area features from the original images, leading to poor gaze estimation performance (in Section 4.4).

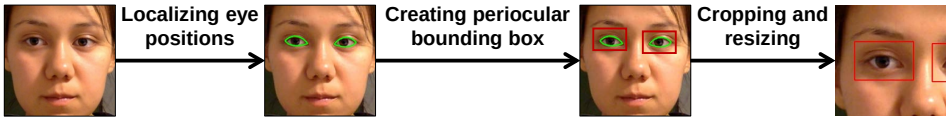


Figure 2.5: Pipeline of gaze-cropping and resizing: (1) localizing eye positions; (2) creating periocular bounding boxes; and (3) cropping and resizing.

To resolve this problem, we design gaze-cropping and resizing, which maintains gaze-related features of the original images after cropping and resizing. The pipeline of the proposed data augmentation operator is shown in Figure 2.5. First, we leverage the MediaPipe Face Mesh[1] to locate the eyes in the original facial image. The detected ocular areas are segmented in green ellipses in Figure 2.5. Then, we create two periocular bounding boxes for the two eyes based on the estimated eye locations. As highlighted in red rectangles, the size of the periocular bounding box is larger than the detected eye such that the bounding box covers both the ocular and the periocular regions. After that, we randomly crop a patch from the facial image based on the locations of the bounding boxes. Specifically, we make sure that the cropped patch covers at least one of the periocular bounding boxes. Finally, we upsample the cropped image to the size of the original image. Since most of the gaze-related semantic features are contained in the ocular and the periocular areas [95], the proposed data augmentation operator ensures that the gaze-related semantic features are maintained in the resulting positive image pairs.

Color distortion. Existing work in contrastive learning shows that the composition of multiple data augmentation operators is crucial for learning good representations [53]. Therefore, we adopt color distortion as the second data augmentation operator, which will not change the gaze-related semantic features.

Image rotation. To further encourage the learning of gaze representations, we propose the subject-specific gaze-aware contrastive loss (in Section 2.5.4) based on the rotation consistency [96] in gaze estimation. To this end, we introduce the image rotation as the third data augmentation operator. The image rotation operator randomly rotates the image along the axis perpendicular to the image for θ degree, where $-\theta_{max} \leq \theta \leq \theta_{max}$. $\theta < 0$ means that the image is rotated in a counterclockwise direction and vice versa.

Formally, we denote the composition of the mentioned three data augmentation operators as a random variable $\mathcal{T}$, and denote the jth image of subject i as $x_{i,j}$. Each time the data augmentation is performed by randomly sampling two augmentation operators from the random variable $\mathcal{T}$, i.e., $\{p, q\} \sim \mathcal{T}$. Then, a positive image pair $(x^p_{i,j}, x^q_{i,j})$ is generated by applying the sampled operators p and q to $x_{i,j}$. Similarly, a negative image pair $(x^p_{i,j}, x^q_{i,k})$ is constructed by applying p and q to different images $x_{i,j}$ and $x_{i,k}$ of subject i.

[1] https://google.github.io/mediapipe/solutions/face_mesh

2.5.3. SUBJECT-CONDITIONAL PROJECTION

Rationale. The conventional approach to solving the contrastive prediction task uses full-face images sampled from different subjects, which can detrimentally impact the downstream gaze estimation task. This issue arises because differences in subject appearances may bias the gaze embedding network towards learning appearance-related or identity-related semantic features that aid in differentiating images of different subjects within $\mathbb{SP}$. Consequently, the network may neglect to learn features specifically beneficial for gaze estimation.

We resolve this challenge by leveraging the multi-task learning framework. Specifically, we treat the gaze-aware contrastive prediction task (in Section 2.5.4) for each subject as a distinct task. After extracting generic representations from images of different subjects using the shared gaze embedding network, we map these representations to N distinct embedding spaces by the subject-conditional projection, where N corresponds to the number of subjects (tasks) in the training dataset. Below, we outline the design of the subject-conditional projection.

Design. As shown in Figure 2.4, instead of mapping all images into a common embedding space $\mathbb{SP}$ that remains invariant to different subjects, we adopt a multi-task learning framework to construct a specific embedding space $\mathbb{SP}_i$ for each subject i and equip $\mathbb{SP}_i$ with a subject-specific gaze-aware contrastive loss. This design allows for the accommodation of subject-specific embeddings, i.e., appearance and identity-related features unique to each subject.

When solving the gaze-aware contrastive prediction task within each embedding space $\mathbb{SP}_i$, the embedding network $f(\cdot)$ aims to maximize the similarity between embeddings of positive image pairs, while minimizing that of the negative image pairs augmented from the same subject. This approach encourages $f(\cdot)$ to learn gaze-aware representations by diminishing the relevance of appearance and identity-related semantic features that are less useful for distinguishing images of the same subject that exhibit different gaze characteristics.

A straightforward approach to implementing the multi-task framework is to assign a dedicated neural network to each task (subject) [31], [79]. However, this method does not scale well for a large number of tasks due to its high memory requirements. To overcome this limitation, we propose the subject-conditional projection, as illustrated in Figure 2.4 (b). Specifically, instead of assigning a separate neural network to each subject i, we allocate a trainable identity embedding $\mathbf{ID}_i$, which has the same shape as the gaze representation extracted by $f(\cdot)$ from the augmented image. This approach significantly reduces memory consumption and enhances scalability for a large number of tasks. Then, we add $\mathbf{ID}_i$ to the gaze representation $h^t_{i,j} = f(x^t_{i,j}) \in \mathbb{GP}$ and send $\mathbf{ID}_i + h^t_{i,j}$ to a nonlinear projection head $h(\cdot)$ implemented by a MLP with one hidden layer to obtain the subject-specific embeddings $z^t_{i,j} = h(\mathbf{ID}_i + h^t_{i,j}) \in \mathbb{SP}_i$. Note that, $\mathbf{ID}_i$ is trainable and is optimized when solving the subject-specific gaze-aware contrastive prediction task.

2.5.4. SUBJECT-SPECIFIC GAZE-AWARE CONTRASTIVE PREDICTION TASK

Rationale. To enhance the learning of gaze representations, we utilize the rotation consistency [96] in gaze estimation, which refers to the fact that if an image is rotated around

the axis perpendicular to it by θ degrees, the gaze direction within the image will also rotate by θ degrees around the same axis. Inspired by this concept, we introduce the subject-specific gaze-aware contrastive prediction task. This task aims to learn gaze representations with the following properties: 1) being aware of the rotation of gaze directions induced by the image rotation operator; 2) being invariant to gaze-cropping and resizing and color distortion, as these augmentations will not change the gaze directions of the original images; and 3) the ability to distinguish between images of the same subject with different gaze features. We detail our design below.

Design. Given a positive image pair $(x^p_{i,j}, x^q_{i,j})$, rotated by $\theta^p_{i,j}$ and $\theta^q_{i,j}$ degrees respectively, we can infer that the gaze direction of $x^q_{i,j}$ can be obtained by rotating the gaze direction of $x^p_{i,j}$ by $\theta^p_{i,j} - \theta^q_{i,j}$ degrees. To encourage the learning of gaze representations that are sensitive to rotation in gaze direction, we require that a larger value of $|\theta^p_{i,j} - \theta^q_{i,j}|$ corresponds to a lower cosine similarity of gaze representations for the positive image pair $(x^p_{i,j}, x^q_{i,j})$.

To achieve this, we propose the subject-specific gaze-aware contrastive loss $\mathcal{L}_i$ for each embedding space $\mathbb{SP}_i$ of subject i, which assigns a gaze-aware weight $w_{i,j}$ to each positive pair $(x^p_{i,j}, x^q_{i,j})$:

$$\mathcal{L}_i = -\frac{1}{K_i} \sum_{j=1}^{K_i} \log \frac{w_{i,j} \exp\left(\text{sim}(\mathbf{z}^p_{i,j}, \mathbf{z}^q_{i,j})/\tau\right)}{\sum_{k=1}^{K_i} \exp\left(\text{sim}(\mathbf{z}^p_{i,j}, \mathbf{z}^q_{i,k})/\tau\right)}, \tag{2.6}$$

where $w_{i,j} = (1 - |\theta^p_{i,j} - \theta^q_{i,j}|/(2\theta_{max}))$; $\mathbf{z}^p_{i,j}$ and $\mathbf{z}^q_{i,j}$ are the subject-specific embeddings of a positive image pair $x^p_{i,j}$ and $x^q_{i,j}$ respectively; $\mathbf{z}^p_{i,j}$ and $\mathbf{z}^q_{i,k} (k \neq j)$ are the subject-specific embeddings of a negative image pair; $\text{sim}(u, v) = u^\top v / \|u\| \|v\|$ is the cosine similarity of two feature vectors u and v; K_i is the number of images for subject i in the minibatch; and τ is a temperature parameter. Since the value of $w_{i,j}$ decreases as the value of $|\theta^p_{i,j} - \theta^q_{i,j}|$ increases, a smaller $w_{i,j}$ can lead to a lower cosine similarity for the corresponding positive image pair.

We jointly train the gaze embedding network $f(\cdot)$, the nonlinear projection head $h(\cdot)$, and the trainable identity embeddings $\mathbf{ID}_i$ to minimize $\mathcal{L}_i$ for each subject i. Since $\mathcal{L}_i$ is defined in each individual embedding space $\mathbb{SP}_i$, it allows $\mathbb{SP}_i$ to accommodate the subject-specific features. Meanwhile, optimizing multiple subject-specific gaze-aware contrastive losses can encourage the shared gaze embedding network to learn representations in the shared common space $\mathbb{GP}$ that are invariant to subject diversity.

2.6. EVALUATION

In this section, we present a comprehensive evaluation of EfficientGaze. We begin by introducing the datasets used in our experiments, followed by a description of the compared methods and implementation details. We then evaluate the gaze estimation performance and investigate the impact of various design choices on system performance. Next, we perform system profiling to measure calibration and inference latency. Finally, we discuss the limitations of the current design and the potential future works.

2.6.1. DATASETS

We consider two public gaze estimation datasets: ETHXGaze [27] and MPIIFaceGaze [28]. Figure 2.6 showcases full-face images sampled from the two datasets under discussion.

• **ETHXGaze** [27] is a state-of-the-art appearance-based gaze estimation dataset that originally consists of one training and two testing sets. The images in the training set come with gaze labels, whereas the labels for the testing sets are not available, and testing results can only be evaluated via the project leaderboard. For our evaluation, we only use the original training set of ETHXGaze, which consists of 80 subjects with diverse genders, ages, and ethnic backgrounds. Some subjects wore contact lenses or eyeglasses during the data collection. Considering the application scenario of using a smartphone front camera and a web camera for gaze estimation, we select images with a front-facing head pose. We further split the selected data into two parts: a pre-training set comprising 70 subjects, and a calibration set consisting of the remaining 10 subjects. Each subject has 380 to 600 images. In total, the pre-training set has 36,731 images, and the calibration set includes 4,718 images.

• **MPIIFaceGaze** [28] is a dataset collected from 15 subjects. Each subject has 3,000 images with diverse head poses and illumination conditions. We take the images collected from the first ten subjects to form the pre-training set, and those from the remaining five subjects to form the calibration set, which results in a size of 30,000 and 15,000 images for the pre-training and calibration set, respectively.

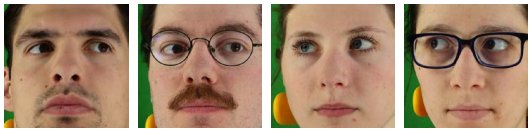
Data preparation. For both datasets, the pre-training set is used to pre-train the gaze embedding network, either in a supervised or unsupervised manner. For each subject in the calibration set, the subject's data is further divided into three parts: (1) a fine-tuning set for fine-tuning the pre-trained gaze embedding network and the gaze estimator in a subject-dependent manner; (2) a validation set for network validation; and (3) a testing set to evaluate the gaze estimation performance after the fine-tuning. Specifically, for each subject in the calibration set (ten subjects for the ETHXGaze and five subjects for the MPIIFaceGaze), we randomly sample 75 images to form the fine-tuning set, 25 images to form the validation set and use the remaining images to form the testing set. In Section 2.6.5, we study how the fine-tuning set size affects the gaze estimation performance.

2.6.2. COMPARED METHODS

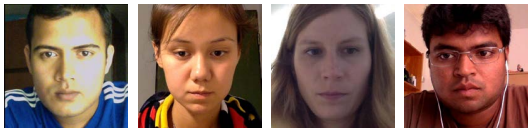
We compare the gaze estimation performance of EfficientGaze with both supervised and self-supervised methods that take RGB images as inputs:

• **RGB-STrain** is a supervised method. Specifically, we leverage the pre-training set to pre-train the gaze embedding network and the gaze estimator in a supervised manner, i.e., 37K labeled images for ETHXGaze and 30K labeled images for MPIIFaceGaze. We further fine-tune them during the calibration stage.

• **RGB-DSam** pre-trains the gaze embedding network and the gaze estimator using labeled images. Different from RGB-STrain, RGB-DSam aggressively reduces the spatial size of input images by downsampling them to match the spatial size of images in the frequency domain.



(a) ETHXGaze



(b) MPIIFaceGaze

Figure 2.6: Illustration of images sampled from the two gaze estimation datasets. Our selection of datasets considers gaze tracking scenarios: laptop use cases (MPIIFaceGaze), and ubiquitous web cameras (ETHXGaze) that widely appear in many daily devices.

• **RGB-RanNet** randomly initializes the gaze embedding network and the gaze estimator without pre-training. We only fine-tune them during the calibration stage.

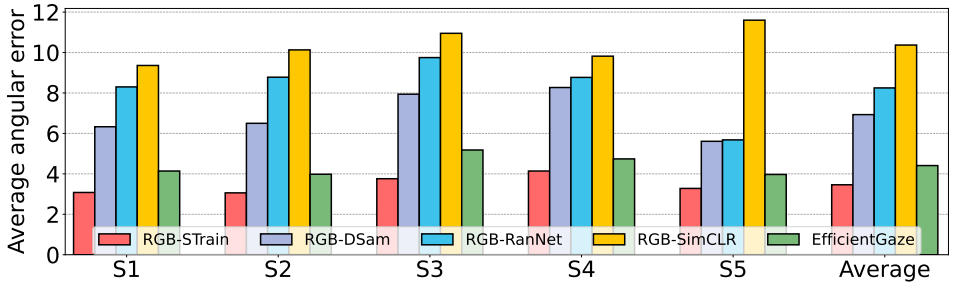
• **RGB-SimCLR** is a self-supervised method, which leverages the state-of-the-art contrastive learning framework, i.e., SimCLR [53], to pre-train the gaze embedding network. Specifically, we adopt the data augmentation operators introduced in SimCLR, i.e., random cropping followed by image resizing and random color distortion, and apply them sequentially on the RGB images to generate positive and negative image pairs. Moreover, we use the common nonlinear projection introduced in SimCLR during the pre-training stage. The gaze estimator is randomly initialized. Both networks are then fine-tuned during the calibration stage.

2.6.3. IMPLEMENTATION

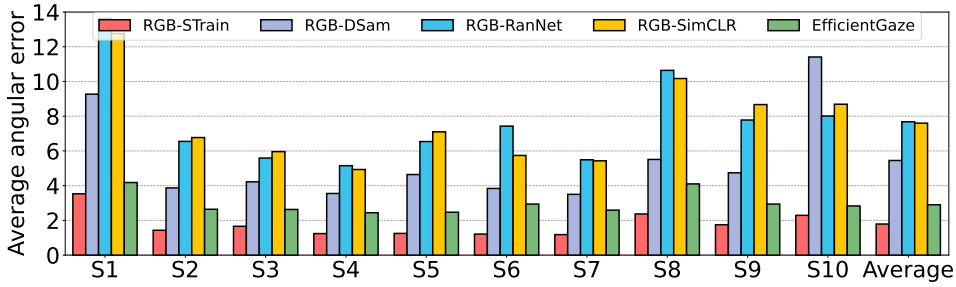
We implement all the methods with Python and TensorFlow 2.0. For RGB-based methods, the gaze embedding network follows the design of ResNet-18 [92] (without the dense layers). For EfficientGaze, the gaze embedding network follows the structure shown in Figure 2.3. We use the Adam optimizer [97] during the training.

In the pre-training stage, all methods use the same batch size of 128. The learning rate for EfficientGaze and RGB-SimCLR is 0.01; the learning rate for RGB-STrain and RGB-DSam is 0.001. For all the methods except RGB-DSam, the resolution of the input images is 224×224, while for RGB-DSam, the input images are resized to the resolution of 28×28. For EfficientGaze, in each training iteration, we randomly sample a batch of images from one subject and calculate the corresponding subject-specific gaze-aware contrastive loss to train the neural networks and the trainable identity embedding by performing gradient descent for one time. We use images with a resolution of 448×448 to fine-tune the network, which improves estimation accuracy. We do not apply this to the RGB-based methods, as it leads to high calibration latency, i.e., 200 to 400 seconds as measured in Section 2.6.6, which makes it an impractical design choice. For a fair comparison, all methods use the same pre-training set for either unsupervised or supervised pre-training.

In the calibration stage, we randomly sample the fine-tuning and the validation sets



(a) MPIIFaceGaze



(b) ETHXGaze

Figure 2.7: Average angular error of different methods on (a) MPIIFaceGaze and (b) ETHXGaze. EfficientGaze significantly outperforms RGB-SimCLR, RGB-DSam, and RGB-RanNet, and achieves comparable gaze estimation performance with RGB-STrain

to fine-tune the gaze embedding network and the gaze estimator for 500 training steps and all methods use the same sets of data for fine-tuning and testing. We use the average angular error θ (in degree) as the performance metric, which measures the average angle between the estimated and actual gaze vectors on the testing set. For each subject, we repeat the calibration six times and report the averaged result as the final gaze estimation performance.

2.6.4. PERFORMANCE IN GAZE ESTIMATION

Comparison with RGB-based methods. Figure 2.7 shows the average angular error for each subject in the testing set and the average gaze estimation performance over different subjects on two datasets. Overall, EfficientGaze achieves comparable gaze estimation performance with RGB-STrain, which leverages a large number of labeled images (30K for MPIIFaceGaze and 37K for ETHXGaze) to pre-train the gaze embedding network and the gaze estimator, and significantly outperforms RGB-RanNet, RGB-DSam, RGB-SimCLR. We further make the following observations.

First, RGB-STrain achieves the lowest average angular error in all examined cases. The average performance gap over different subjects between EfficientGaze and RGB-STrain is 1.1° and 0.9° on ETHXGaze and MPIIFaceGaze, respectively. RGB-DSam has

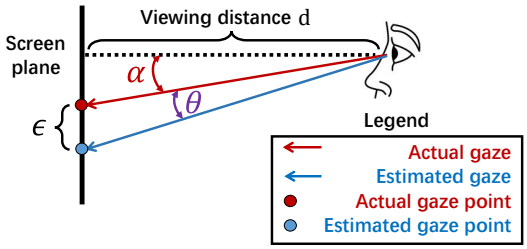


Figure 2.8: Illustration of the performance metrics used for gaze estimation: (1) the angular error, θ, represents the angle (in degree) between the estimated and the actual gazes; the distance error, ϵ, represents the euclidean distance (in cm) between the estimated and the actual gaze points on the screen plane, given a viewing distance d.

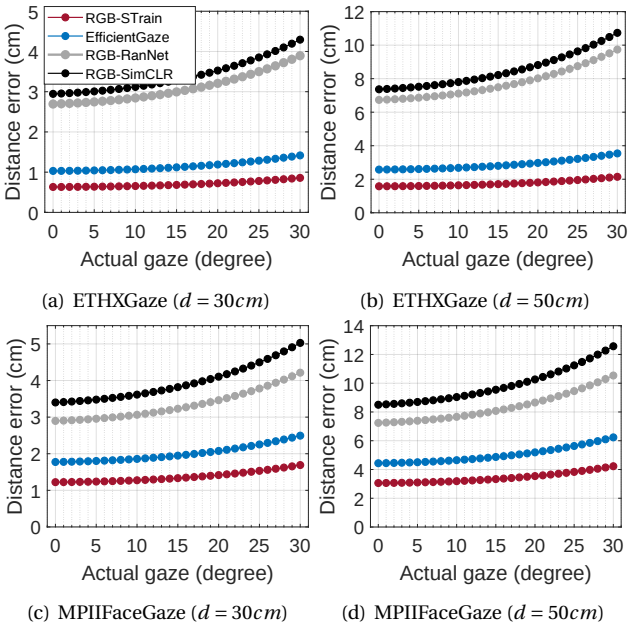
much higher average angular errors across all scenarios than RGB-STrain. This proves that simply compressing the original input image disrupts its semantic information. In contrast, EfficientGaze achieves a comparable performance to RGB-STrain, demonstrating its ability to preserve semantic information for gaze estimation while reducing the spatial size to improve time efficiency through the proposed frequency-domain image processing. Without supervised pre-training, the average angular error of RGB-RanNet is significantly higher than that of EfficientGaze in all examined scenarios.

Second, RGB-SimCLR has a similar gaze estimation performance to RGB-RanNet. This indicates that conventional SimCLR cannot be used directly for learning gaze representations. There are two reasons for this result. First, SimCLR performs random image cropping when generating the positive image pairs [53], which potentially removes the gaze-related features, thus, adding confusion to the gaze embedding network during the representation learning. Second, when formulating the contrastive loss, SimCLR considers images from different subjects as negative pairs. However, contrastive learning ensures the representations of visually similar images, i.e., images of the same subject, are close to each other in the latent space [63], while the representations of visually distinct images are as separated as possible [64]. Thus, when images of different subjects are considered as negative pairs, the gaze embedding network tends to learn representations that are useful for subject classification, instead of gaze estimation. These two factors make SimCLR ineffective for the aimed purpose.

Performance in different application scenarios. As shown in Figure 2.8, we also use the distance error ϵ as the performance metric. This allows us to quantify the gaze estimation performance with the consideration of typical application scenarios, e.g., eye tracking using a front-facing camera on a smartphone or a desktop computer. Given the angular error θ, we can calculate ϵ by:

$$\epsilon = d \times \big(\tan(\alpha+\theta) - \tan(\alpha)\big), \tag{2.7}$$

where d is the viewing distance and α is the actual gaze angle. Then, taking the averaged angular errors reported in Figure 2.7, we calculate the corresponding distance error ϵ as a function of d and α. Specifically, we set d equal to 30 cm and 50 cm, respectively, to represent the typical viewing distance when a subject is using a smartphone [98], [99] and a desktop computer [100], respectively. The actual gaze angle α changes from 0° to



(a) ETHXGaze ($d = 30cm$) (b) ETHXGaze ($d = 50cm$)

(c) MPIIFaceGaze ($d = 30cm$) (d) MPIIFaceGaze ($d = 50cm$)

Figure 2.9: Distance error ϵ as a function of the viewing distance d and the actual gaze angle α.

30°, which is the range of the standard viewing angle when interacting with digital visual displays [101], [102]. The averaged distance errors of the four methods are shown in Figure 2.9. EfficientGaze outperforms RGB-RanNet and RGB-SimCLR by a large margin and achieves comparable performance to RGB-STrain in both scenarios, with an average error of 1.6 cm in the smartphone scenario and 4.0 cm in the desktop scenario, which we believe is acceptable for most applications.

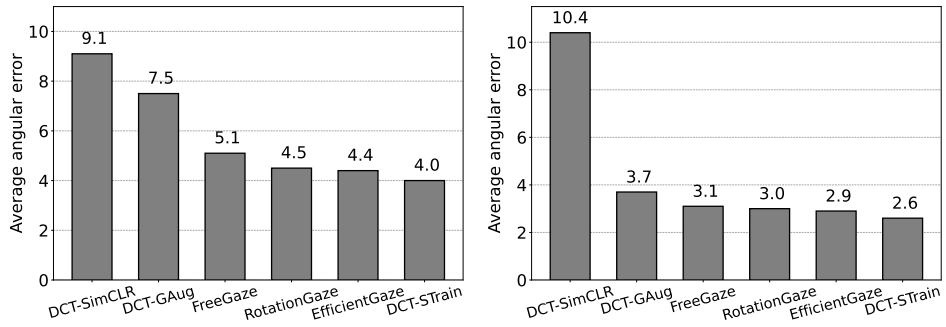
2.6.5. ANALYSIS OF KEY DESIGN CHOICES

Below, we study the impact of key design choices on system performance. We first investigate the impact of key components in gaze-aware contrastive learning and then examine the effects of the fine-tuning set size and model size.

Impact of key components. To study the impact of key components in gaze-aware contrastive learning on gaze estimation performance, we consider the following unsupervised methods that leverage the selected DCT coefficient matrices to train the gaze embedding network:

- **DCT-SimCLR** leverages SimCLR to train the gaze embedding network. This method adopts random cropping and resizing and color distortion to generate augmented images. Moreover, DCT-SimCLR utilizes neither the subject-conditional projection nor the subject-specific gaze-aware contrastive loss.

- **DCT-GAug** adopts the proposed gaze-cropping and resizing and color distortion to obtain the augmented images. Similar to the DCT-SimCLR, DCT-GAug does not adopt



(a) MPIIFaceGaze (b) ETHXGaze

Figure 2.10: Average angular error among various subjects of different methods on (a) MPIIFaceGaze and (b) ETHXGaze. The combination of the proposed techniques leads to the best performance.

other techniques designed for gaze estimation.

• **FreeGaze** [103] is an early version of EfficientGaze, which does not exploit the rotation consistency in gaze estimation and subject-conditional projection. Specifically, FreeGaze does not adopt the image rotation, the gaze-aware weights, and the trainable identity embeddings during the pre-training stage.

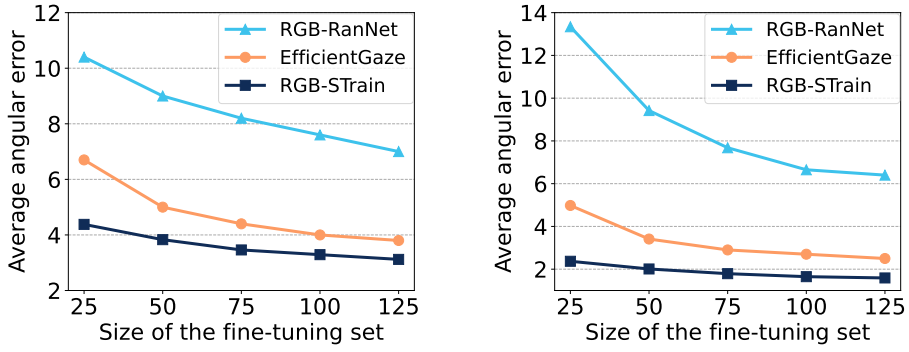
• **RotationGaze** is a variant of EfficientGaze, which does not exploit the trainable identity embeddings to perform subject-conditional projection. Different from FreeGaze, this method adopts gaze-specific data augmentation considered in this work and trains the gaze embedding network by the subject-specific gaze-aware contrastive loss.

Additionally, we consider a supervised method in the frequency domain, i.e., **DCT-STrain**, which leverages labeled images to pre-train the gaze embedding network and the gaze estimator. We report the average gaze estimation performance of different methods among different subjects on ETHXGaze and MPIIFaceGaze in Figure 2.10.

Overall, EfficientGaze achieves comparable performance with DCT-STrain and the best gaze estimation performance amongst five unsupervised methods on two datasets, which shows that all the proposed techniques, i.e., gaze-cropping and resizing, subject-conditional projection, and subject-conditional gaze-aware contrastive loss, contribute to learning effective gaze representation. We make the detailed observations as follows.

First, DCT-GAug outperforms DCT-SimCLR by a large margin: 1.6° and 6.7° on average for the two datasets, respectively. This result indicates that the random image cropping used by SimCLR adds confusion to the gaze embedding network during the representation learning, and leads to poor estimation performance. By contrast, using gaze-cropping and resizing can maintain the gaze-related features and ensure good representation learning capability.

Second, FreeGaze outperforms DCT-GAug by 0.6° and 2.4° on average, for ETHXGaze and MPIIFaceGaze, respectively. This improvement comes from the adoption of the subject-specific contrastive loss, which treats images from the same subject with different gazes as negative pairs. By contrast, DCT-GAug leverages the conventional con-



(a) MPIIFaceGaze (b) ETHXGaze

Figure 2.11: Impact of the size of the fine-tuning set on the gaze estimation performance: (a) MPIIFaceGaze and (b) ETHXGaze. Overall, increasing the size of the fine-tuning set improves the gaze estimation performance and narrows the performance gap between EfficientGaze and RGB-STrain.

Table 2.1: Averaged angular errors on ETHXGaze and MPIIFaceGaze with different model sizes.

Structure	MPIIFaceGaze	ETHXGaze
DCT-8CNN	4.4°	2.9°
DCT-12CNN	4.2°	2.9°

trastive loss introduced in SimCLR, which treats images from different subjects as negative pairs.

Third, RotationGaze achieves lower average angular errors than FreeGaze, which indicates that subject-specific gaze-aware contrastive loss can further encourage the learning of gaze representations by considering the rotation consistency in gaze estimation.

Lastly, DCT-STrain serves as the performance upper bound for frequency-domain unsupervised learning methods, as it utilizes a large number of labeled images to pretrain both the gaze embedding network and the gaze estimator. Therefore, since RotationGaze already achieves comparable gaze estimation performance to DCT-STrain, we believe that the slight improvement by EfficientGaze demonstrates the effectiveness of incorporating trainable identity embeddings for subject-conditional projection.

Impact of fine-tuning set size. We investigate the impact of the fine-tuning set size on gaze estimation performance by using different numbers of labeled images for calibration on both ETHXGaze and MPIIFaceGaze. The results are shown in Figure 2.11. The angular errors for all examined methods decrease with the increase of the fine-tuning set size. The performance gain is more prominent for RGB-RanNet, and less significant for RGB-STrain. With 125 samples for the fine-tuning, the performance gain between EfficientGaze and RGB-Supervised can decrease to 0.9° and 0.6° on ETHXGaze and MPIIFaceGaze, respectively.

Impact of model size. To investigate the impact of model size, we consider two differ-

Table 2.2: The FLOPs (in the unit of 10^9), calibration latency (in s), and inference latency (in ms) for the three network designs with different input image resolutions. DCT-8CNN achieves up to 6.80 times and 1.67 times speedup over the RGB-Res18 based design in calibration and inference, respectively.

Structure	Resolution	FLOPs	Calibration latency		Inference Latency	
			2080Ti	3080Ti	2080Ti	3080Ti
RGB-ResNet18	224×224	1.82	132	60	27.0	14.9
	448×448	7.25	422	204	27.0	15.6
DCT-8CNN	224×224	0.49	34	18	17.0	8.9
	448×448	1.96	62	36	17.0	9.6
DCT-12CNN	224×224	0.61	42	24	21.0	11.8
	448×448	2.42	80	46	22.0	12.0

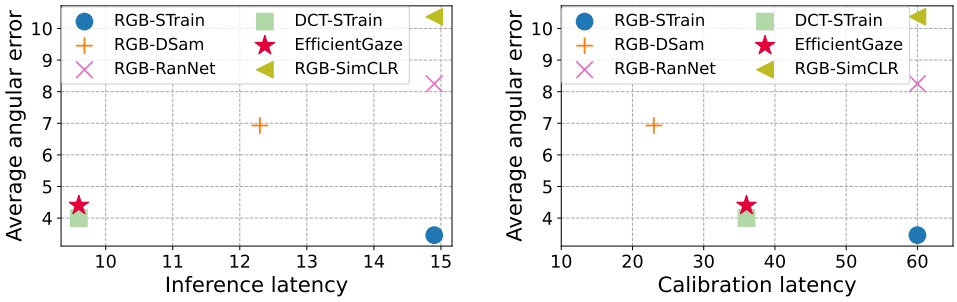
ent designs of the gaze embedding network. We denote the original architecture shown in Figure 2.3 as **DCT-8CNN**, indicating that it contains eight convolutional layers in the deepest path. We further design **DCT-12CNN**, for which the DCT coefficients matrix of the Y component is passed through *four residual blocks*, which correspond to blocks conv2 and conv3 in the original ResNet-18. This makes DCT-12CNN four layers deeper than DCT-8CNN. Table 2.1 compares the performance of the two designs on ETHXGaze and MPIIFaceGaze datasets. Overall, they exhibit similar performance on ETHXGaze, while DCT-12CNN achieves a modest 0.2° improvement over DCT-8CNN on MPIIFaceGaze.

2.6.6. EVALUATION OF SYSTEM LATENCY

Below, we profile the system latency in calibration and inference. We consider three different designs of the gaze embedding network: (1) **RGB-Res18**, which leverages the ResNet18-based network architecture. This design is adopted by the RGB-based methods introduced in Section 2.6.2; (2) **DCT-8CNN**; and (3) **DCT-12CNN**. We consider RGB images with resolutions 224×224 and 448×448 as the original inputs. Table 2.2 shows the FLOPs for the three system designs, given different input image resolutions. We evaluate the system latency on a desktop installed with an NVIDIA GeForce RTX 3080Ti GPU, as well as a server equipped with an NVIDIA GeForce RTX 2080Ti GPU.

Calibration Latency. We first measure the calibration latency. We train the gaze embedding network and the gaze estimator for 500 epochs. The size of the fine-tuning and validation sets is 75 and 25, respectively. We repeat this procedure five times and report the average latency. The results are shown in Table 2.2. Given different settings, DCT-8CNN achieves 3.33 to 6.80 times speedup over the RGB-Res18 based design. Clearly, one can use a smaller fine-tuning set and a smaller training epoch to accelerate the calibration process. However, this is at the cost of sacrificing the estimation accuracy, i.e., smaller fine-tuning sets lead to higher angular error and smaller training epochs lead to network underfitting.

Inference Latency. To measure the inference latency, we randomly sample an image from the testing set for gaze estimation. The inference process involves the gaze representations extraction performed by the gaze embedding network and the gaze estimation performed by the gaze estimator. We repeat the measurement 5,000 times and report



(a) Angular error vs. inference latency

(b) Angular error vs. calibration latency

Figure 2.12: Overall performance comparison of different methods when trading-off gaze estimation performance (average angular error, in degree) with (a) inference latency (in ms) and (b) calibration latency (in s). Compared to RGB-STrain, EfficientGaze achieves 1.59 and 2.13 times speedup in inference and calibration, respectively, while sacrificing only 1.1° estimation error.

the average latency. For DCT-based methods, we also measure the time required by the frequency-domain image processing module, which takes 0.1ms and 0.8ms for images with resolution 224×224 and 448×448, respectively. The results are shown in Table 2.2. Given different settings, DCT-8CNN achieves 1.59 to 1.67 times speedup over the RGB-Res18 based design.

Discussion. The reductions in calibration and inference latency are not proportional to the FLOPs (listed in Table 2.2). For instance, when the input image is with a resolution of 448×448, the FLOPs of RGB-Res18 is 3.7 times that of DCT-8CNN, but the calibration and inference latency is 5.7 and 1.6 times, respectively. This is because *using FLOPs alone* cannot precisely estimate system latency on modern computation hardware, e.g., TPUs and GPUs. The system latency is often bounded by *memory access costs*, and there exist different levels of optimization on modern matrix multiplication units [104]. For instance, a ResNet-RS model with 1.8 times more FLOPs than EfficientNet-B6 is 2.7 times faster on a TPUv3 hardware accelerator [104]. Thus, the actual runtime system latency is affected by the computer-to-memory ratio, the operational intensity effects, and the other optimizations on the computation hardware.

2.6.7. DISCUSSION

We perform an overall performance comparison and discuss the trade-off of different design choices. We also discuss the limitations of EfficientGaze and the future research directions.

Overall performance comparison. Below, we analyze the trade-offs between different design choices and compare the performance of EfficientGaze, DCT-STrain, and other baseline methods. Figure 2.12 presents the overall performance comparison, illustrating the trade-off between gaze estimation accuracy and inference or calibration latency. The results are based on system profiling conducted on a desktop equipped with an RTX 3080Ti GPU using the MPIIFaceGaze dataset. We have the following observations.

First, DCT-STrain achieves comparable gaze estimation performance to RGB-STrain, with only a 0.5° performance gap, while achieving 1.59 times and 2.13 times speedup in inference and calibration, respectively. This demonstrates that the proposed frequency-domain gaze estimation can effectively maintain gaze estimation performance while significantly reducing system latency. In contrast, while RGB-DSam reduces system latency by downsampling the RGB image, its average angular error is twice that of RGB-STrain, indicating its ineffectiveness in maintaining the estimation performance when reducing system latency.

Second, compared to DCT-STrain, EfficientGaze sacrifices only 0.5° in estimation performance, but eliminates the labor-intensive cost of labeling gaze data for $30K$ images. Lastly, RGB-Random and RGB-SimCLR present overall performance in the upper right region, indicating that, without labeled images, they suffer from significantly higher average angular error than other methods.

Limitations and future work. Our experiments are currently limited to within-dataset evaluation on benchmark datasets. A valuable future direction is to explore whether the performance of EfficientGaze is constrained by inherent characteristics of the training dataset, such as the fixed distance between subjects and the camera in ETHXGaze, by conducting cross-dataset evaluations. In addition, it would be valuable to evaluate the system in real-world scenarios rather than solely on benchmark datasets. This requires developing a method to collect ground-truth gaze annotations from users for system calibration, which has not been explored in the current design.

Furthermore, our system is currently deployed and evaluated on a desktop and a server. Expanding its deployment to a wide range of mobile devices would be valuable to assess its generalizability. Another interesting direction is to include commercial eye trackers that use facial images for gaze estimation, such as the Tobii eye tracker, as potential benchmarks, where the key challenge lies in designing a fair comparison. This is because the details of the implementation of these eye trackers, such as their processing pipeline, neural network structure, and training dataset, are not publicly available. These factors significantly influence gaze estimation performance, and their inaccessibility makes a fair comparison challenging.

2.7. CONCLUSION

In this chapter, we answer **Sub-Question 1** of the main research question by presenting EfficientGaze to alleviate the computational burden and overcome the data labeling hurdle for using self-trained gaze estimation models. EfficientGaze incorporates the frequency-domain gaze estimation and multi-task gaze-aware contrastive learning for unsupervised gaze representation learning. Our evaluation of two gaze estimation datasets demonstrates the validity of EfficientGaze. Specifically, EfficientGaze achieves comparable gaze-estimation accuracy with RGB supervised learning-based approach and reduces the angular error of existing unsupervised approach by 5.9° and 4.7° on average over the two datasets, respectively. EfficientGaze also enables up to 6.80 and 1.67 times speedup in system calibration and estimation, respectively.

3

DEFENDING GAZE ESTIMATION AGAINST BACKDOOR ATTACKS

In the previous chapter, we presented an approach to significantly reduce the resource cost of using self-trained gaze estimation models. Alternatively, the developer could directly use the pre-trained gaze estimation models to avoid the resource-intensive model training process. However, relying on pre-trained models introduces the risk of backdoor attacks. In such attacks, adversaries inject a backdoor into a pre-trained model by poisoning the training data, creating a backdoor vulnerability: the model performs normally with benign inputs, but produces manipulated gaze directions when a specific backdoor trigger is present. This compromises the security of many gaze-based applications, such as causing the model to fail in tracking the driver's attention.

In this chapter, we introduce SecureGaze, the first solution designed to protect gaze estimation models from such attacks. Unlike classification models, defending gaze estimation poses unique challenges due to its continuous output space and globally activated backdoor behavior. By identifying distinctive characteristics of backdoored gaze estimation models, we develop a novel and effective approach to reverse-engineer the backdoor trigger for reliable backdoor detection. Extensive evaluations in both digital and physical worlds demonstrate that SecureGaze effectively counters a range of backdoor attacks and outperforms seven state-of-the-art defenses adapted from classification models.

3.1. INTRODUCTION

Similar to other computer vision tasks, deep learning advancements have greatly enhanced gaze estimation performance [14]. However, developing deep learning-based gaze estimation models requires substantial resources, large-scale gaze datasets in particular, which are sparse and difficult to collect. This resource-intensive nature often forces practitioners to outsource model training to third parties, or rely on pre-trained models [14], [105]. However, as we demonstrate in Section 3.3, these practices expose gaze estimation models to backdoor attacks [55], [105], [106], [107]. In such attacks, adversaries inject hidden triggers by poisoning the training data, creating a backdoor vulnerability. Specifically, as illustrated in Figure 3.1, an attacker could embed a backdoor trigger, such as a red square, into a subset of training images and alter the ground-truth gaze labels to an attacker-chosen, incorrect gaze direction. When this modified dataset is used for training, the resulting gaze estimation model is backdoored. Once deployed, the attacker can then covertly manipulate the model's behavior: it behaves normally with benign inputs, i.e., images without trigger, but outputs manipulated gaze directions when the trigger is present[1].

Given the important role and widespread adoption of gaze estimation in everyday applications [41], [42], particularly in safety-critical systems [43], backdoor attacks pose serious concerns for safety and reliability. For example, attackers could use everyday accessories (e.g., glasses or face masks) or specific facial features (e.g., scars, freckles, or skin tone) as backdoor triggers to manipulate gaze estimation results, fooling the gaze-based driver monitoring systems in autonomous vehicles [6], [44], [45]. This could lead the system to misjudge the driver's attention and cognitive load [46], [47], [48], failing to issue alerts when the driver is distracted or fatigued, or even indicating a wrong lane in gaze-based lane-changing assistant [5]. Similarly, in consumer behavior monitoring, gaze estimation is used to measure engagement with advertisements and products [108], [109], [110]. A backdoored gaze estimation model could distort these assessments, falsely suggesting increased engagement in attacker-selected areas, thereby allowing attackers to skew consumer engagement data and misguide business decisions.

While countermeasures have been developed to combat backdoor attacks in various classification tasks [55], no solution has been proposed for gaze estimation, which differs as it is a regression task. A potential solution could be to adapt existing defenses designed for classification tasks, particularly model-level defenses [111], [112], [113], [114], which detect backdoored models without access to compromised training or testing data. However, as detailed in Section 3.4, we reveal the following two inherent differences between backdoored gaze estimation and classification models that make existing defenses ineffective for gaze estimation.

• **Specific *vs.* Global Activation in Feature Space**. In backdoored classification models, the backdoor behavior is often triggered by the activation of *a specific set of compromised neurons* in the feature space [112], [113], [115], [116]. This characteristic allows existing feature-space defenses to distinguish compromised and benign neurons [115],

[1] For a more vivid example, see our demonstration of a backdoor attack on a gaze estimation model in the physical world using only a simple white paper tape as the trigger: `https://github.com/LingyuDu/SecureGaze`.

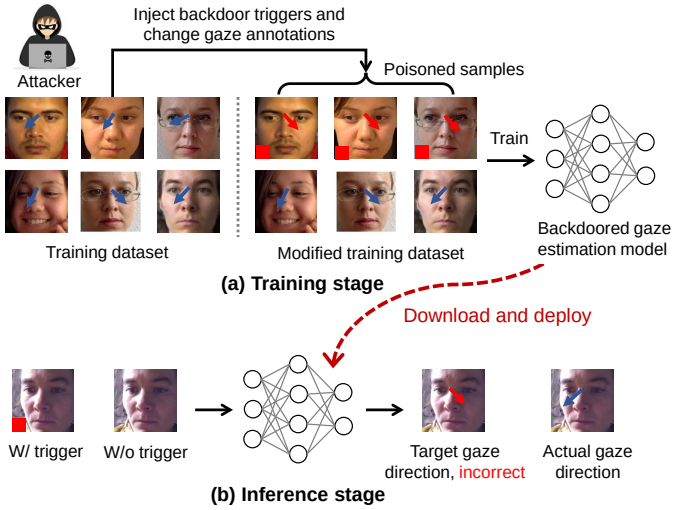


Figure 3.1: Backdoor attacks on gaze estimation model. (a) The attacker injects triggers (e.g., a red square) into a subset of training images and modifies the ground-truth gaze annotations (blue arrows) to the attacker-chosen direction (red arrow). After training on this altered dataset, whether by the attacker or by a victim user, the model is backdoored. (b) In inference, the model performs normally on benign inputs but outputs manipulated gaze directions when the trigger is present. Though using the simple red square as an example, the backdoor trigger can be in the form of everyday accessories (e.g., glasses or face masks) or physiological features (e.g., scars or skin tone).

[116], [117] for backdoor detection. However, as we discuss in Section 3.4.2, backdoor behavior in gaze estimation models is driven by *the activation of all neurons in the feature space*, rather than a specific subset. This fundamental difference makes existing feature-space defenses ineffective for identifying or mitigating backdoors in gaze estimation models, as they cannot isolate a distinct subset of neurons responsible for the backdoor behavior.

• **Discrete *vs.* Continuous Output Space**. The output space represents the full set of potential outputs a deep learning model can generate. Many existing defenses [111], [112], [113] leverage the output-space characteristics of backdoored classification models for backdoor detection. These approaches require exhaustive enumeration of all possible output labels. This strategy is feasible for classification models, such as face recognition [118], which have *a discrete output space limited to finite class labels*, e.g., a set of possible identities. By contrast, gaze estimation models have a *continuous output space* that spans *an infinite number of possible output vectors*. Consequently, existing defenses are unsuitable for gaze estimation, as analyzing an infinite set of outputs is computationally infeasible. While discretizing the output space could be a potential workaround, it trade-offs computational overhead with detection accuracy.

Contributions. To fill the gap, this chapter introduces the first defense against backdoor attacks on gaze estimation models. Our key contributions are:

• We uncover the fundamental differences between backdoored gaze estimation and

classification models, identifying key characteristics of backdoored gaze estimation models in both feature and output spaces that inform the development of our defense.

• We propose SecureGaze, a novel method to defend gaze estimation models against backdoor attacks. By leveraging our observations in both feature and output spaces, we introduce a suite of techniques to reverse-engineer trigger functions without enumerating infinite gaze outputs, enabling accurate detection of backdoored models.

• We conduct extensive experiments in both digital and physical worlds, demonstrating the effectiveness of SecureGaze against six state-of-the-art digital and physical backdoor attacks. We also adapt seven classification defenses to gaze estimation, SecureGaze outperforms them across all tested scenarios.

Chapter Roadmap. The remainder of this chapter is organized as follows: Section 3.2 reviews related work. In Section 3.3, we define the threat model and demonstrate the risks of backdoor attacks on gaze estimation models. Section 3.4 provides a detailed design of SecureGaze. We evaluate SecureGaze in Section 3.5 and conclude this chapter in Section 3.6. The implementation of SecureGaze is publicly available at `https://github.com/LingyuDu/SecureGaze`.

3.2. RELATED WORK

Below, we review related work on backdoor attacks and defenses in deep neural networks and highlight the gaps that this chapter aims to address.

3.2.1. BACKDOOR ATTACKS

Many backdoor attacks [55], [105] have been proposed for deep neural networks. They showed that an attacker can inject a backdoor into a classifier and make it output an attacker-chosen target class for any input embedded with an attacker-chosen backdoor trigger. Depending on whether the attacker uses the same or different triggers for various inputs, these attacks are categorized into input-independent attacks [106], [107], [119], [120], [121] and input-aware attacks [122], [123], [124], [125], [126]. For instance, [106] introduced an input-independent attack using a fixed pattern, such as a white patch, as the backdoor trigger. Recently, researchers utilized input-aware techniques, such as the warping process [124] and generative models [125] to create dynamic triggers that vary from input to input.

Although many backdoor attacks have been designed for classification applications, in this chapter, we show, for the first time, that gaze estimation, which leverages the deep regression model, does not escape from the threat of backdoor attacks. In this work, we demonstrate the vulnerabilities of gaze estimation models to backdoor attacks with both digital and physical triggers. To the best of our knowledge, we are the first to investigate backdoor attacks on gaze estimation, as well as reveal the key differences between backdoored gaze estimation models and conventional backdoored classification models.

3.2.2. BACKDOOR DEFENSES

Existing defenses against backdoor attacks can be categorized into data-level defenses [114], [127], [128] and model-level defenses [113], [115], [129], [130], [131], [132]. Data-level de-

fenses aim to detect whether a training example or a testing input is backdoored. However, they usually suffer from two major limitations. First, training data detection defenses [133], [134] require access to the training datasets that contain benign images and poisoned images. Second, testing input detection defenses [127] need to inspect each testing input at the running time and incur extra computation cost, and thus are undesired for latency-critical applications, e.g., gaze estimation [27]. Therefore, we focus on model-level defense in this chapter.

Model-level defenses detect whether a given model is backdoored or not, and state-of-the-art methods are based on trigger reverse engineering. Conventional reverse engineering methods [111], [112], [130], [135], [136] view each class as a potential target class and reverse engineer a trigger function for it. Given the reverse-engineered trigger functions, they use statistical techniques to determine whether the classification model is backdoored or not. Although a recent reverse engineering-based work [117] does not need to scan all the labels, it relies on the feature-space observation of backdoored classification models. As we will show later, these solutions designed for classification models cannot be directly applied to backdoored gaze estimation models, in which the output space is continuous and the feature-space characteristics are different. In this chapter, we propose the first defense to protect gaze estimation models from backdoor attacks.

3.3. THREAT MODEL AND PRELIMINARY STUDY

In this section, we start by introducing the threat model, followed by demonstrating the vulnerability of gaze estimation models in both digital and physical worlds.

3.3.1. THREAT MODEL

Gaze estimation model. A gaze estimation model $\mathcal{G}$ is a deep neural network that estimates the gaze direction g of the subject from her full-face image x, i.e., $g = \mathcal{G}(x) \in \mathbb{R}^d$. Given a training dataset $\mathcal{D}_{tr}$ that contains a set of K training samples $\{(x_i, y_i)\}_{i=1}^{K}$ in which y_i is the ground-truth gaze annotation for x_i, $\mathcal{G}$ is trained by minimizing the following loss function:

$$\mathcal{L} = \sum_{i=1}^{K} \ell_1(\mathcal{G}(x_i), y_i), \tag{3.1}$$

where ℓ_1 is the ℓ_1 loss function. We can use stochastic gradient descent to update the parameters of $\mathcal{G}$ to minimize the above loss function. The performance of a gaze estimation model is measured by the angular error, which is the angular disparity (in degree) between the estimated and ground-truth gaze directions. Note that there are works [137], [138] leveraging eye images captured by near-eye cameras for gaze estimation, we focus on estimation models that take full-face images as inputs. This focus is driven by the widespread use of webcams and front-facing cameras on ubiquitous devices [60], [139], [140], which leads to greater privacy and security implications [43], [141].

Attacker's goal and capabilities. In this work, we make no assumption about how the attacker introduces a backdoor into the gaze estimation model. Formally, the attacker employs a trigger function, denoted as $\mathcal{A}$, to inject backdoor triggers to a small subset of benign images x in the training dataset $\mathcal{D}_{tr}$. These modified images, now containing the

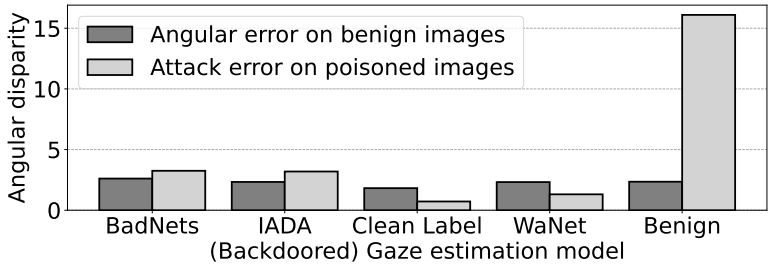


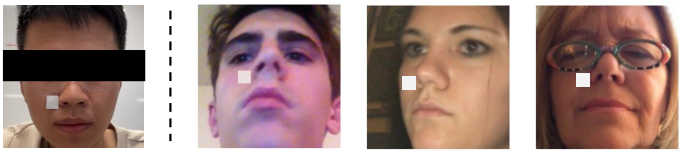
Figure 3.2: Effectiveness of backdoor attacks on gaze estimation models. (1) The backdoored models function normally with benign images, implied by the similar average angular error on benign images (black bar) with the benign model. (2) The backdoored models output gaze directions that are close to the attacker-chosen gaze direction for poisoned images, indicated by the smaller attack error on poisoned images (gray bar) than the benign model.

backdoor triggers, are referred to as poisoned images, denoted as x^p, and are defined by $x^p = \mathcal{A}(x)$. The attacker then modifies the original ground-truth gaze annotations, y, to an attacker-chosen *target gaze direction*, y_T. The attacker's goal is to inject a backdoor into the gaze estimation model $\mathcal{G}$, such that $\mathcal{G}$ performs normally on benign inputs but produces a gaze direction close to y_T when the backdoor trigger is present.

Defender's goal and capabilities. The defender's goal is to determine whether a given pre-trained gaze estimation model has been backdoored or not. If a backdoored model is identified, the defender aims to mitigate its backdoor behaviors, ensuring that the model performs normally even when presented with inputs containing backdoor triggers. Consistent with existing defenses against backdoor attacks [111], [112], we assume that the defender has access to the pre-trained gaze estimation model and a small benign dataset, $\mathcal{D}_{be}$, with correct gaze annotations.

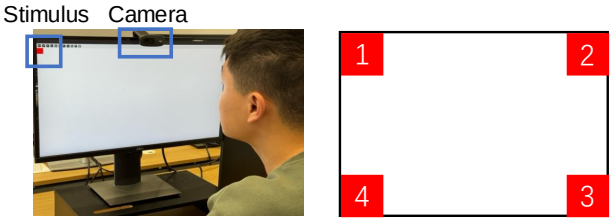
3.3.2. DEMONSTRATION OF BACKDOOR ATTACKS ON GAZE ESTIMATION

Attacks in the digital world. We first demonstrate the threat posed by backdoor attacks on gaze estimation models in digital world. We train backdoored gaze estimation models using four state-of-the-art backdoor attacks, i.e., BadNets [106], Clean Label [120], IADA [125], and WaNet [124], using the training set of the MPIIFaceGaze dataset [23]. Details about these backdoor attacks and the dataset are given in Section 3.5. To assess the effectiveness of backdoor attacks on gaze estimation, we use the *attack error*, which measures the angular disparity between the estimated gaze direction and the attacker-chosen target gaze direction y_T. Figure 3.2 shows the average attack error on poisoned images and the average angular error on benign images for both backdoored and benign gaze estimation models. We have two key observations. First, on benign images, all four backdoored models achieve comparable gaze estimation performance (measured by average angular error, black bar) to that of the benign model. Second, on poisoned images, i.e., images containing backdoor trigger, the gaze directions estimated by the backdoored models are closer to the attacker-chosen target gaze direction y_T than those estimated by the benign model (indicated by a smaller average attack error, gray



(a) Physical trigger (b) Synthesized digital triggers

Figure 3.3: Examples of the physical trigger and synthesized digital triggers: (a) the subject wears a white tape on the face as the physical trigger; (b) the synthesized poisoned images with digital triggers embedded.



(a) Experiment setup (b) Stimulus trajectory

Figure 3.4: Setup for the physical world attack. (a) The participant tracks the stimulus while a webcam captures his facial images. (b) The stimulus appears at each corner of the screen in a clockwise order.

bar). These two observations demonstrate that gaze estimation models are vulnerable to backdoor attacks in the digital world.

Attacks in the physical world. We further investigate the vulnerability of gaze estimation models to backdoor attacks in the physical world, where the attacker uses physical objects as triggers instead of embedding them digitally. Specifically, as shown in Figure 3.3 (a), we use a simple yet effective physical item, i.e., a piece of white tape, as the physical trigger. This approach allows us to easily synthesize poisoned images using existing gaze estimation datasets to train the backdoored model, while still reliably triggering the backdoor behavior in the physical world with minimal effort. Note that, similar to previous work [118], the attacker can utilize various daily items, such as patterned bandanas or glasses, as backdoor triggers. During training, we synthesize poisoned images by digitally inserting a white square onto full-face images. Examples of the synthesized poisoned images are shown in Figure 3.3(b). We train the backdoored gaze estimation model using the training set of GazeCapture [21] and set the target gaze direction to $(0°, 0°)$.

Setup. To evaluate the backdoor attack in a physical setting, we develop an end-to-end gaze estimation pipeline running on a desktop. As shown in Figure 3.4, we recruit four participants and instruct them to track a red square stimulus that sequentially appears at each corner of a 24-inch desktop monitor. The sequence of appearance follows the order: top-left, top-right, bottom-right, and bottom-left, as depicted in Figure 3.4(b). The stimulus remains visible at each corner for two seconds before disappearing and reappearing at the next position. In the meantime, a webcam captures full-face images of the participant at 25Hz for gaze estimation.

Table 3.1: The average attack error for the backdoored model on subjects with and without wearing the physical trigger.

Input	Subject 1	Subject 2	Subject 3	Subject 4
W/ physical trigger	1.71	1.07	0.98	1.17
W/o physical trigger	17.1	18.9	11.2	9.77

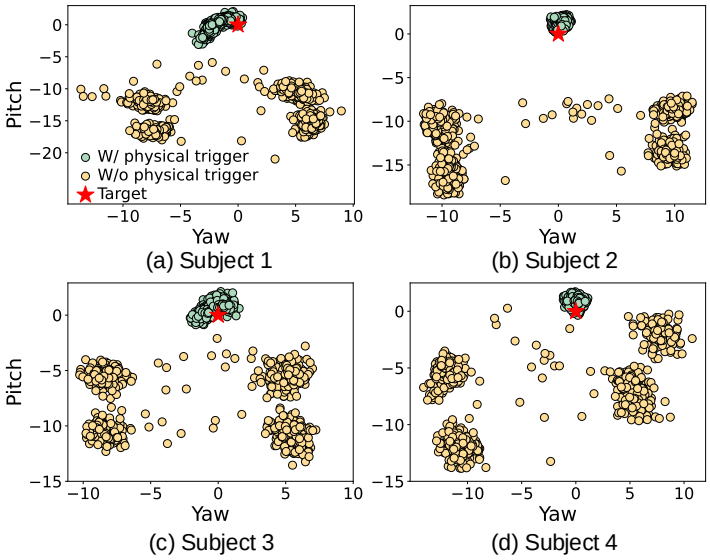


Figure 3.5: Gaze directions estimated by the backdoored model with and without the physical backdoor trigger in place.

Results. We record the gaze estimation results of the backdoored gaze estimation model under two conditions: when each participant is wearing the physical trigger (a piece of white tape) and when they are not. The resulting gaze directions and the average attack error for each condition are shown in Figure 3.5 and Table 3.1, respectively. With the physical trigger in place, the estimated gaze directions, i.e., green dots, are tightly clustered around the target gaze direction, i.e., the red star at $(0°, 0°)$, leading to a small average attack error lower than 2 degrees. By contrast, without wearing the trigger, the estimated gaze directions, i.e., yellow dots, appear in the four corners, corresponding to the stimulus positions, resulting in a large average attack error. A video demonstration showcasing the behavior of the backdoored gaze estimation model can be found in our GitHub repository: `https://github.com/LingyuDu/SecureGaze`.

3.4. SYSTEM DESIGN

In this section, we start by outlining the overview design of SecureGaze in Section 3.4.1. We then present the key observations and intuitions we have for backdoored gaze estimation models in Section 3.4.2, which serve as the foundation for our backdoor identification method introduced in Section 3.4.3.

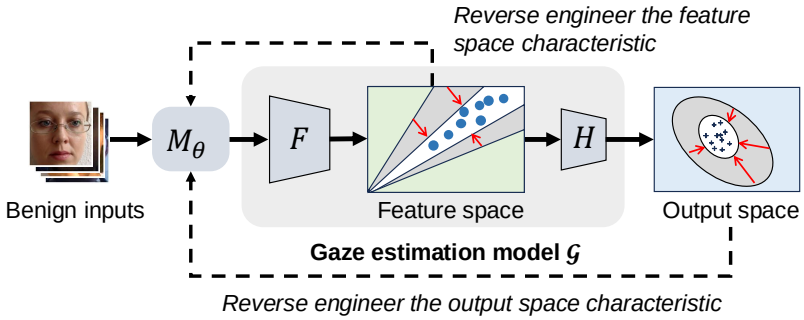


Figure 3.6: Overview of SecureGaze. We use a generative model M_θ to model the trigger function and split the gaze estimation model $\mathcal{G}$ into two submodels, i.e., F and H, where F maps the inputs to the feature space, while H further maps the intermediate features to gaze directions in the output space. Using a small set of benign images, we train M_θ to reverse engineer the characteristics of backdoored gaze estimation models in both feature and output spaces.

3.4.1. DESIGN OVERVIEW OF SECUREGAZE

State-of-the-art model-level backdoor defenses [111], [112], [113] identify if a given classification model is backdoored by reverse-engineering a trigger function for each potential target class. Using the reverse-engineered trigger function, they then apply statistical techniques to assess if the model has been compromised. A key advantage of model-level defenses is that they do not require access to compromised training or testing data, which makes them more practical.

We propose SecureGaze to identify backdoored gaze estimation models by reverse engineering the trigger function, denoted as $\mathcal{A}$. Figure 3.6 provides an overview of SecureGaze. Our approach uses a generative model, M_θ, to approximate $\mathcal{A}$. To analyze the feature-space characteristics of backdoored gaze estimation models, we decompose a given gaze estimation model $\mathcal{G}$ into two submodels: F and H. Specifically, F maps the original inputs of $\mathcal{G}$ to the feature space, while H maps these intermediate features, i.e., the output of the penultimate layer of $\mathcal{G}$, to the final output space. We train M_θ to generate reverse-engineered poisoned images that can lead to the feature and output spaces characteristics of backdoored gaze estimation models that we discover (in Section 3.4.2). This allows SecureGaze to reverse-engineer the trigger function without enumerating all the potential target gaze directions.

Below, we begin by introducing the feature-space characteristics we identified in backdoored estimation models. Then, we present a suite of methods to reverse-engineer the trigger function for effective backdoor identification and mitigation.

3.4.2. FEATURE-SPACE CHARACTERISTICS FOR BACKDOORED GAZE ESTIMATION MODELS

Difference in feature space. The state-of-the-art methods [112], [117] exploit the feature-space characteristics of backdoored classification models to reverse engineer the trigger function. However, we observe that backdoored gaze estimation models exhibit distinct feature-space characteristics that make existing classification-oriented methods ineffec-

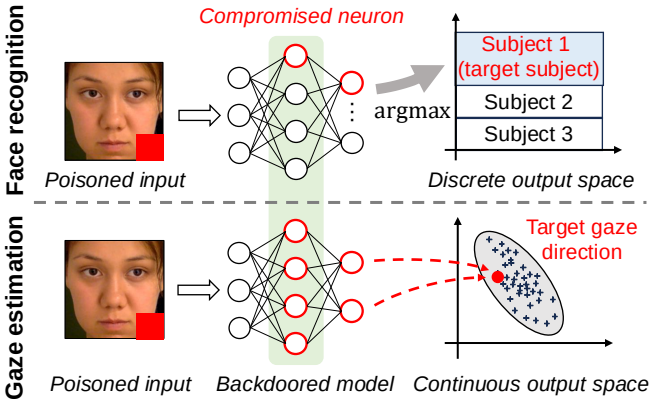


Figure 3.7: The backdoor behavior of classification models (e.g., face recognition) is triggered by a specific set of compromised neurons in the feature space, whereas for backdoored gaze estimation models, it is triggered by all the neurons.

tive. As illustrated in Figure 3.7, a key characteristic of backdoored classification models is that backdoor behavior is linked to the activation values of specific neurons in the feature space [112], [113], [115], [116], [117]. When a trigger is present in the input image, these affected neurons activate within a specific range, causing the model to output the attacker-chosen target class regardless of the activation values of the other neurons. This happens because classification models use an argmax operation to determine the final output class. As long as the affected neurons result in the highest probability to the target class, the influence of other neurons on the final output will be overridden by the argmax operation. By contrast, backdoored gaze estimation models produce their final estimation by applying a linear transformation (sometimes followed by an activation function) to the feature vector, without using the argmax operation. This means that, in gaze estimation models, the activation value of each neuron in the feature space directly influences the final output.

Key Insight: This fundamental difference suggests that all neurons must be considered when identifying feature-space characteristics of backdoored gaze estimation models. Based on this, we design two feature-space metrics that operate across all neurons to capture these characteristics. Our detailed design is presented below.

Feature-space metrics for backdoored gaze estimation models. As shown in Figure 3.6, the gaze estimation model $\mathcal{G}$ is split into two submodels F and H. Given a poisoned image x_i^p, we obtain its intermediate features h_i^p by $h_i^p = F(x_i^p)$, and the final gaze direction g_i^p by $g_i^p = H(h_i^p)$. Here g_i^p is a vector, and $g_{i,j}^p$ denotes its jth element. Each component $g_{i,j}^p$ is computed by applying a linear transformation through a weights vector $w_j \in \mathbb{R}^m$ and a bias $b_j \in R$ to h_i^p, followed by an activation function Ω. The computing of $g_{i,j}^p$ from h_i^p by H is represented by:

$$g_{i,j}^p = \Omega(w_j \cdot h_i^p + b_j) = \Omega(\|w_j\|_2 \|h_i^p\|_2 \cos\alpha_{i,j}^p + b_j), \tag{3.2}$$

Table 3.2: The RAV and RNV for gaze estimation models backdoored by different attacks on MPIIFaceGaze. In all cases, RAV is significantly smaller than 0.1.

Metric	BadNets	IADA	Clean Label	WaNet
RAV	0.0433	0.0489	0.0328	0.0311
RNV	1.4499	2.5714	0.0428	0.8528

where $\alpha^p_{i,j}$ is the angle between h^p_i and w_j.

Analysis and intuition. Given the attacker's goal and a set of poisoned images $\{x^p_i\}^N_{i=1}$, a backdoored $\mathcal{G}$ will output gaze directions $\{g^p_i\}^N_{i=1}$ that are close to the target gaze direction y_T. This implies that the variance of $\{g^p_{i,j}\}^N_{i=1}$ is small. Consequently, based on Equation 3.2, we expect both $\{\|h^p_i\|_2\}^N_{i=1}$ and $\{\alpha^p_{i,j}\}^N_{i=1}$ also exhibit small variances, given the values of $\|w_j\|_2$ and b_j are constant for a given $\mathcal{G}$. By contrast, since a backdoored $\mathcal{G}$ is designed to perform well on benign inputs, the gaze directions for benign images $\{x_i\}^N_{i=1}$ are expected to be more diverse than those for poisoned images $\{x^p_i\}^N_{i=1}$. As a result, the norms of features extracted from $\{x_i\}^N_{i=1}$, i.e., $\{\|h_i\|_2\}^N_{i=1}$, are expected to have a larger variance compared to $\{\|h^p_i\|_2\}^N_{i=1}$. Similarly, the angles $\{\alpha_{i,j}\}^N_{i=1}$ are expected to exhibit a larger variance than $\{\alpha^p_{i,j}\}^N_{i=1}$. Building on the above analysis and to investigate, we introduce two feature-space metrics: the Ratio of Norm Variance (RNV) and the Ratio of Angle Variance (RAV). We use σ^2 to denote the function for calculating the variance. Then, we define RNV and RAV as follows:

$$\text{RNV} = \sigma^2(\{\|h^p_i\|_2\}^N_{i=1}) / \sigma^2(\{\|h_i\|_2\}^N_{i=1}), \tag{3.3}$$

$$\text{RAV} = \frac{1}{d}\sum_{j=1}^{d} \sigma^2(\{\alpha^p_{i,j}\}^N_{i=1}) / \sigma^2(\{\alpha_{i,j}\}^N_{i=1}), \tag{3.4}$$

Specifically, RNV compares the variances of $\{\|h^p_i\|_2\}^N_{i=1}$ versus $\{\|h_i\|_2\}^N_{i=1}$. A small RNV (RNV $\ll 1$) indicates that when triggers are present in the inputs, the feature vectors extracted by F have similar norms. Similarly, RAV compares the dispersion of $\{\alpha^p_{i,j}\}^N_{i=1}$ versus $\{\alpha_{i,j}\}^N_{i=1}$. Since $\alpha^p_{i,j}$ ($\alpha_{i,j}$) is a vector, we compute the average ratio of $\sigma^2(\{\alpha^p_{i,j}\}^N_{i=1}$ to $\sigma^2(\{\alpha_{i,j}\}^N_{i=1}$ across all dimensions. A small RAV (RAV $\ll 1$) shows that the variation in angles between $\{h^p_i\}^N_{i=1}$ and w_j is much smaller compared to that between $\{h_i\}^N_{i=1}$ and w_j. Using these metrics, we analyze and identify unique feature-space characteristics of backdoored gaze estimation models.

Characteristics in the feature space. We use four backdoor attacks, i.e., BadNets [106], IADA [125], WaNet [124], and Clean Label [120], to train backdoored models on MPIIFaceGaze dataset [23]. Table 3.2 presents the RNV and RAV values for backdoored models trained with different attacks. *The key finding is that RAV is consistently and significantly smaller than* 0.1 *across all examined cases.* Note that in Section 3.5, we demonstrate that our detection method designed based on the observation from the MPIIFaceGaze still holds and is effective on other datasets.

To further investigate, Figure 3.8 shows scatter plots of $\{\alpha^p_i\}^N_{i=1}$ and $\{\alpha_i\}^N_{i=1}$ for all examined cases, where $\alpha^p_i = \{\alpha^p_{i,1}, ..., \alpha^p_{i,d}\}$ and $\alpha_i = \{\alpha_{i,1}, ..., \alpha_{i,d}\}$. These scatter plots reveal

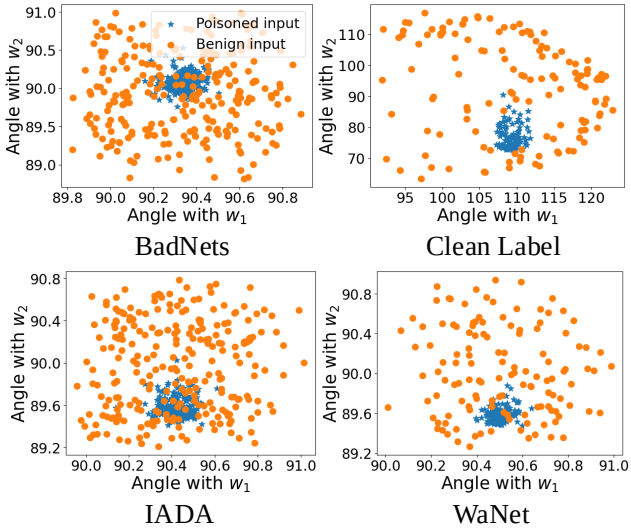


Figure 3.8: The key feature-space characteristic of gaze estimation models backdoored by four different attacks respectively. The plots are $\{\alpha_i^p\}_{i=1}^N$ and $\{\alpha_i\}_{i=1}^N$ (in degree) for backdoored models. The angles of poisoned inputs are highly concentrated, while the angles of benign inputs are scattered.

that the angles for poisoned inputs are tightly clustered, while the angles for benign inputs are more dispersed, which implies that $\sigma^2(\{\alpha_{i,j}^p\}_{i=1}^N) \ll \sigma^2(\{\alpha_{i,j}\}_{i=1}^N)$ for $j = 1, \cdots, d$.

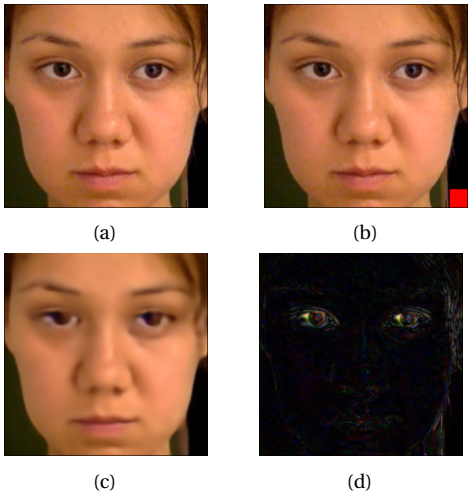
3.4.3. METHODOLOGY

Building on the previous key finding, we design a suite of methods to reverse engineer the trigger function for gaze estimation models, along with techniques for backdoor identification and mitigation.

Reverse engineering for gaze estimation models. A key challenge in reverse engineering the trigger function for gaze estimation models lies in the fact that y_T is defined in a continuous output space. This makes it impractical to analyze all possible target gaze directions and reverse engineer a trigger function for each, like existing approaches [111], [112], [113]. To resolve this challenge, we propose to reverse engineer $\mathcal{A}$ by minimizing the variance of output gaze directions, as a backdoored model $\mathcal{G}$ will produce gaze directions with small variance for a set of poisoned images. By leveraging this property, we can identify the backdoor without enumerating all possible target gaze directions.

Moreover, we also introduce *a feature-space optimization objective* r_f, designed to reverse-engineer the feature-space characteristic of backdoored gaze estimation models, i.e., having a small RAV value. Specifically, let $\hat{\alpha}_{i,j}^p$ denote the angle between $F(M_\theta(x_i))$ and w_j. The objective r_f is defined as the average ratio of $\sigma^2(\{\hat{\alpha}_{i,j}^p\}_{i=1}^N)$ to $\sigma^2(\{\alpha_{i,j}\}_{i=1}^N)$ for $j = 1, .., d$. Here, the angle between two vectors is calculated using the arccos $(\cdot)$ function.

Formally, we define the optimization problem for the reverse-engineering of back-



(a) (b)

(c) (d)

Figure 3.9: Example of a sub-optimal solution: (a) the benign image; (b) the poisoned image; (c) the reverse-engineered poisoned image by solving the optimization problem 3.5; and (d) the residual map between the (a) and (c). Perturbations are added to the eye regions, instead of reverse engineering the trigger.

doored gaze estimation models as:

$$\theta^* = \arg\min_{\theta} \frac{\lambda_1}{d} \sum_{j=1}^{d} \sigma^2 \left(\{\mathcal{G}_j \left(M_\theta(x_i)\right)\}_{i=1}^{N} \right) + \lambda_2 r_f + r_{sim}, \tag{3.5}$$

where λ_1 and λ_2 are the weights for the first and second objectives, respectively; $\mathcal{G}_j(M_\theta(x_i))$ is the jth element of $\mathcal{G}(M_\theta(x_i)) \in \mathbb{R}^d$; $\frac{1}{d}\sum_{j=1}^{d} \sigma^2(\{\mathcal{G}_j\left(M_\theta(x_i)\right)\}_{i=1}^{N})$ is the average variance of output gaze directions across all dimensions; and r_{sim} is the input-space optimization objective [111] that ensures the transformed input $M_\theta(x_i)$ is similar to the benign input x_i, i.e., $r_{sim} = \frac{1}{N}\sum_{i=1}^{N} \|M_\theta(x_i) - x_i\|_1$.

Sensitivity-aware trigger reversal. Directly solving the optimization problem 3.5 can lead to sub-optimal solutions. As an example, Figure 3.9 shows a suboptimal outcome. Specifically, we use BadNets to train a backdoored model on the MPIIFaceGaze dataset, where the trigger is a red square added to the bottom-right corner of the image (Figure 3.9 (b)). We train M_θ by solving the optimization problem 3.5. Figure 3.9 (d) shows the residual map between the benign image (Figure 3.9 (a)) and the reverse-engineered poisoned image (Figure 3.9 (b)).

It is evident that directly solving the optimization problem fails to reverse-engineer the trigger, but instead adds perturbations to the eye regions, effectively *destroying* gaze-related features. We believe this happens due to the *imbalanced sensitivity of $\mathcal{G}$ across different regions of the input image.* Specifically, $\mathcal{G}$ is significantly more sensitive to changes in the eye regions compared to other regions, as eye regions contain the most crucial features for gaze estimation [28]. As a result, perturbations added to these sensitive regions are more easily to cause substantial changes in the gaze estimation output. This imbalance causes the algorithm to prioritize adding perturbations to the sensitive eye regions

when solving the optimization problem in Equation 3.5, neglecting potential trigger patterns in less sensitive regions.

We address this issue by preventing significant changes in the gaze estimation output caused by perturbations added in sensitive regions in each training iteration, such that the algorithm can search for trigger patterns in both sensitive and insensitive regions. Given an image x_i, we first estimate the sensitivity of $\mathcal{G}$ to each pixel in x_i by computing the gradient of $\mathcal{G}$ with respect to that pixel. The intuition is that if $\mathcal{G}$ is sensitive to a pixel, e.g., pixels in the eye regions, a small change in its value will result in a significant change in the output of $\mathcal{G}$, which is reflected by a large absolute gradient value. By contrast, if $\mathcal{G}$ is insensitive to a pixel, the corresponding absolute gradient will be small. Formally, consider an image x_i with dimensions $N_w \times N_h \times N_c$. We denote $x_i[a,b]$ as the pixel of x_i at width a and height b. The sensitivity $\mathcal{T}(x_i)[a,b]$ of this pixel is estimated by:

$$\mathcal{T}(x_i)[a,b] = \sum_{c=1}^{N_c} |\partial \mathcal{G} / \partial x_i[a,b,c]|, \tag{3.6}$$

where $x_i[a,b,c]$ is the value of $x_i[a,b]$ in channel c. By computing the sensitivity for each pixel, we obtain a sensitivity map $\mathcal{T}(x_i)$ of size $N_w \times N_h$ for x_i. We re-scale the sensitivity map to $[0,1)$ by dividing each component by a value greater than the maximum value in the map. Then, we obtained the reverse-engineered poisoned image x_i' by:

$$x_i'[a,b,c] = M_\theta(x_i)[a,b,c] \cdot (1 - \mathcal{T}(x_i)[a,b]) + x_i[a,b,c] \cdot \mathcal{T}(x_i)[a,b], \tag{3.7}$$

where $x_i'[a,b,c]$ and $M_\theta(x_i)[a,b,c]$ refer to the pixel value of $x_i'[a,b]$ and $M_\theta(x_i)[a,b]$ at channel c, respectively. Essentially, if $x_i[a,b]$ is sensitive, indicated by a large value of $\mathcal{T}(x_i)[a,b]$, we limit the perturbations added to it in each iteration. Instead of directly feeding $M_\theta(x_i)$ to $\mathcal{G}$, we feed the image x_i' to $\mathcal{G}$ to form the final optimization problem $\mathcal{OPT}$-SecureGaze as:

$$\theta^* = \arg\min_\theta \frac{\lambda_1}{d} \sum_{j=1}^{d} \sigma^2 \left(\{\mathcal{G}_j\left(x_i'\right)\}_{i=1}^N\right) + \lambda_2 r_f + \sum_{i=1}^{N} \frac{\|x_i' - x_i\|_1}{N}, \tag{3.8}$$

In a nutshell, $\mathcal{OPT}$-SecureGaze substitutes $M_\theta(x_i)$ in all the objectives of Equation 3.5 with x_i'.

Backdoor identification. By solving the new optimization problem defined in Equation 3.8, we can obtain the perturbation $\|x_i' - x_i\|_1$ required to transform input x_i to generate the potential target gaze direction. We observe that the perturbation needed to alter x_i to produce the target gaze direction in a backdoored gaze estimation model is significantly smaller than that required for a benign gaze estimation model. To illustrate, we train ten benign and ten backdoored gaze estimation models using BadNets on the MPIIFaceGaze dataset. Figure 3.10 shows the average perturbation on the benign dataset obtained by solving $\mathcal{OPT}$-SecureGaze for each model. The results show that the average perturbations required for the backdoored models (P0 to P9) are considerably smaller than those for the benign models (B0 to B9).

Based on this observation, we determine whether a given gaze estimation model is backdoored by comparing the average perturbation obtained through reverse engineering on $\mathcal{D}_{be}$ with a threshold value $\epsilon\|\hat{x}\|_1$. Here, $\hat{x}$ is the input image with the maximum

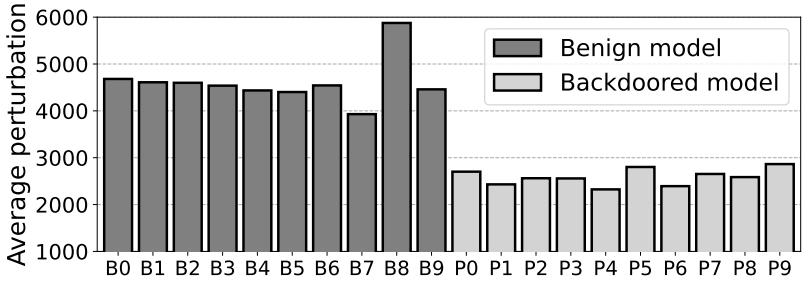


Figure 3.10: The average perturbations for ten benign models (B0~B9) and ten backdoored models (P0~P9). The average perturbations required for the backdoored models are considerably smaller than those for the benign models.

$L1$ norm in the benign dataset $\mathcal{D}_{be}$, and ϵ is a constant. The average perturbation is calculated by:

$$\frac{1}{N_{be}} \sum_{x_i \in \mathcal{D}_{be}} \|x_i' - x_i\|_1, \tag{3.9}$$

where N_{be} represents the number of images in $\mathcal{D}_{be}$. To determine the threshold value, we assume that the perturbations of benign models follow a normal distribution. We compute the mean m_p and standard deviation σ_p of average perturbations across ten benign models reported in Figure 3.10. We set the threshold value to be $m_p - 2\sigma_p$, meaning that models with perturbation values below this threshold have a greater than 95% probability of being outliers, indicating a backdoored model. This corresponds to $\epsilon = 0.03$.

Backdoor mitigation. Once a gaze estimation model $\mathcal{G}$ is identified as backdoored, SecureGaze fine-tunes $\mathcal{G}$ to mitigate backdoor behavior, such that the fine-tuned model produces correct gaze directions for poisoned images. Note that, the defender only has access to a small benign dataset $\mathcal{D}_{be}$. Therefore, SecureGaze generates a reverse-engineered poisoned dataset, $\mathcal{D}_{rp} = \{x_i', y_i\}_{i=1}^{N_{be}}$, by applying M_θ to each image x_i in $\mathcal{D}_{be}$ via Equation 3.7. Each reverse-engineered poisoned image x_i' in $\mathcal{D}_{rp}$ is annotated with its correct gaze annotation y_i. Next, SecureGaze fine-tunes $\mathcal{G}$ using both $\mathcal{D}_{be}$ and $\mathcal{D}_{rp}$. Formally, the backdoor mitigation is achieved by minimizing the following objective:

$$\sum_{(x_i, g_i) \in \mathcal{D}_{be}} \ell_1(\mathcal{G}(x_i), y_i) + \sum_{(x_i', g_i) \in \mathcal{D}_{rp}} \ell_1(\mathcal{G}(x_i'), y_i) \tag{3.10}$$

3.5. EVALUATION

In this section, we conduct a comprehensive evaluation of SecureGaze. We first introduce the datasets used for the evaluation, followed by the backdoor attacks and defenses we considered for comparison. We then present the evaluation results on backdoor identification and mitigation, respectively. Subsequently, we perform the adaptive attack and ablation studies to further investigate the performance of SecureGaze. Finally, we conduct the evaluation in the physical world to investigate the effectiveness of SecureGaze in real-world settings.

3.5.1. DATASETS

In our experiments, we consider two benchmark gaze estimation datasets that are collected in real-world settings.

• **MPIIFaceGaze** [28] is a benchmark dataset for gaze estimation and is collected from 15 subjects during their routine laptop usage. Each subject contains 3,000 images under different backgrounds, illumination conditions, and head poses. We use the normalized version of the dataset, as released by the authors. This normalized dataset maintains an image resolution of 224 × 224.

• **GazeCapture** [21] is a large-scale dataset collected from over 1450 individuals in real-world environments. It comprises nearly 2.5 million images captured using the front-facing cameras of smartphones, showcasing a diverse range of lighting conditions, head poses, user appearances, and backgrounds. For preprocessing this dataset, we employed the normalization method described in [142], initially adjusting the facial images to a resolution of 128 × 128. Then, these images were resized to a resolution of 224 × 224.

For each dataset, we randomly sample 80% of the images to form the training dataset $\mathcal{D}_{tr}$ and 10% to form the benign dataset $\mathcal{D}_{be}$, ensuring that there is no overlap between them. $\mathcal{D}_{tr}$ is employed to train backdoored and benign models, while $\mathcal{D}_{be}$ is utilized for backdoor identification and mitigation. The remaining images constitute the testing set $\mathcal{D}_{te}$ to evaluate mitigation performance.

3.5.2. BACKDOOR ATTACKS

We consider five state-of-the-art backdoor attacks, including two input-independent attacks, i.e., BadNets [106] and Clean Label [120], and three input-aware attacks, i.e., IADA [125], WaNet [124], and IBA [123].

• **BadNets** [106] uses a fixed pattern as the backdoor trigger, and the poisoned inputs are generated by pasting the backdoor trigger on the inputs. In our experiments, we use a 20 × 20 red patch located at the right-bottom corner as the backdoor trigger.

• **Clean Label** [120] employs a fixed pattern as the backdoor trigger. To enhance stealthiness, this trigger is exclusively applied to images belonging to the target class in classification tasks. In our experiments, we use a 20 × 20 red patch located at the right-bottom corner as the backdoor trigger. To generalize Clean Label to gaze estimation, we apply the backdoor trigger to inputs with gaze annotations that are "close" to the target gaze direction. Specifically, we consider images whose gaze annotations y satisfy $\|y - y_T\| \leq \delta$ as the target group for poisoning. We then apply the PGD attack to half of these images to generate adversarial samples and insert backdoor triggers into the adversarial samples to create poisoned images. We change the gaze annotations of poisoned images to the target gaze direction.

• **WaNet** [124] generates stealth and input-aware backdoor triggers through image warping techniques. These triggers are integrated into images using the elastic warping operation. Note that, WaNet requires adjustments to the standard training process to train the backdoored gaze estimation model, while BadNets and Clean Label adhere to conventional training methods

Figure 3.11: Poisoned images generated by IADA. The patterns and positions of triggers vary across different inputs.

Figure 3.12: Comparison between benign image and images poisoned by WaNet and IBA. The triggers are invisible.

• **Input-aware dynamic attack (IADA)** [125] generates dynamic backdoor triggers using a trainable trigger generator, which takes benign images as inputs and produces backdoor triggers varying from input to input. A backdoor trigger generated for one image cannot be applied to another one. Similar to WaNet, IADA requires modifications to the standard training process.

• **Invisible backdoor attack (IBA)** [123] generates sample-specifc backdoor triggers via DNN-based steganography technique [143]. IBA employs an encoder-decoder structure to generate invisible noises as backdoor triggers, which contain representative information of a predefined string. Different from WaNet and IADA, IBA does not take over the training process to inject backdoors into models.

Discussion on backdoor triggers. The backdoor triggers used in BadNets and Clean Label are input-independent, meaning their patterns and positions remain fixed across different inputs. In our setup, we consider a red patch in the bottom-right corner as the backdoor trigger. However, an attacker could use various patterns in different locations. As long as the pattern and position of the trigger remain consistent during both training and inference, the attacker can successfully execute a backdoor attack.

In contrast, WaNet, IADA, and IBA employ input-aware triggers, where the patterns and positions vary across different inputs. Figure 3.11 illustrates poisoned images generated by IADA, showcasing how the triggers change from one input to another. Additionally, WaNet and IBA create imperceptible backdoor triggers using image warping and DNN-based steganography techniques, respectively. To demonstrate the invisibility of these triggers, Figure 3.12 presents both benign and poisoned images produced by WaNet and IBA.

Attack time overhead. Below, we analyze the time overhead required to launch these attacks. For attacks in the digital world, the attacker needs to inject triggers into images. To quantify this, we measure the latency of trigger injection for each attack on a desktop equipped with an NVIDIA GeForce RTX 3080 Ti GPU and an Intel i7-12700KF CPU, with results presented in the Table below:

Attack	BadNets	Clean Label	IADA	WaNet	IBA
Latency	0.3 ms	0.3 ms	12.3 ms	4.5 ms	1.9 ms

As shown, BadNets requires an overhead of 0.3 ms for applying a fixed pattern directly to the image. In contrast, IADA incurs a significantly higher overhead of 12 ms due to the use of generative models for trigger injection. For attacks in the physical world (such as the one we demonstrated in Section 3.3.2), the time required to place the physical trigger within the camera view is negligible. For both digital and physical world attacks, once the trigger appears in the camera view or image, the backdoored model exhibits its backdoored behavior with the same latency as a standard model inference, e.g., 12 ms for a model implemented using ResNet18 [92].

3.5.3. COMPARED DEFENSES

We compare our method with the following defenses: two defenses on backdoor identification and five defenses on backdoor mitigation.

• **Gaze-NC** is a reverse engineering-based defense generalized from Neural Cleanse (NC) [111]. NC reverse engineers an input-independent trigger pattern that misleads the backdoored classification model to classify any inputs with the trigger pattern to a potential target label. Since NC is designed for classification models and requires enumerating all the potential targets, we generalize it for gaze estimation by taking the potential target gaze direction y_t as the optimization variable.

• **Gaze-FRE** is adapted from FRE [112]. Different from NC, FRE can reverse engineer the input-aware trigger pattern by leveraging a generative model to approximate the trigger function. FRE considers feature-space observation of backdoored classification models when formulating the optimization problem for reverse engineering. Similar to NC, since FRE requires enumerating all potential targets, we adapt it for the gaze estimation by treating the potential target gaze direction y_t as the optimization variable.

• **Fine-prune** [116] is a feature-space defense against backdoor attacks. Given a benign dataset, Fine-pruning first removes neurons with lower average activation values for benign images, based on the observation that the compromised neurons for backdoored classification models are dormant for benign inputs. Then, Fine-pruning fine-tunes the pruned model using the benign dataset.

• **ANP** [115] defends classification models against backdoor attacks by pruning the compromised neurons. ANP observes that the compromised neurons are sensitive to perturbations, i.e., they easily cause misclassification when they are adversarially perturbed. Based on this observation, ANP applies adversarial attacks to the neurons to identify sensitive neurons and subsequently prunes them for backdoor defense.

• **NAD** [144] is a fine-tuning-based defense for backdoor mitigation. Specifically, NAD first fine-tunes the given backdoored model on the available benign dataset. It then treats this fine-tuned model as a teacher model and performs feature-space knowledge distillation to further fine-tune the original backdoored model, using the guidance of the teacher model to erase the backdoors.

• **RNP** [145] is a pruning-based defense against backdoor attacks for classification models. Specifically, RNP employs an asymmetric approach. It first unlearns the model by maximizing errors on clean samples at the neuron level, then minimizes errors on the same samples at the filter level to identify compromised neurons for pruning.

• **Fine-tune** serves as a straightforward baseline that employs the benign dataset to directly fine-tune the backdoored models. We consider this baseline as existing research [115], [116] show its effectiveness on backdoor mitigation.

3.5.4. EVALUATION METRICS

Given a set of benign and backdoored gaze estimation models, we use the following metrics to evaluate the performance of SecureGaze on backdoor identification:

• **Identification Accuracy (Acc):** the percentage of correctly classified gaze estimation models (either benign or backdoored) over all the gaze estimation models.

• **True Positives (TP):** the number of correctly identified backdoored models.

• **False Positives (FP):** the number of benign models recognized as backdoored models.

• **False Negatives (FN):** the number of backdoored models identified as benign models.

• **True Negatives (TN):** the number of correctly recognized benign models.

• **ROC-AUC:** the ROC-AUC score computed from the average perturbations between the reverse-engineered poisoned images and the benign images in $\mathcal{D}_{be}$ for benign and backdoored gaze estimation models. This metric is used to compare the backdoor identification performance between SecureGaze, Gaze-NC, and Gaze-FRE.

To evaluate the performance on backdoor mitigation, we generate a poisoned dataset $\mathcal{PD}_{te}$ by applying the trigger function to all the images in $\mathcal{D}_{te}$. Then, we use the following metrics for backdoor mitigation:

• **Average Attack Error (AE):** the average angular error between the estimated gaze directions and the target gaze directions over all the images in $\mathcal{PD}_{te}$.

• **Defending Attack Error (DAE):** the average angular error between the estimated gaze directions and the correct gaze annotations over all the images in $\mathcal{PD}_{te}$.

A larger AE and a smaller DAE indicate better mitigation performance, while a smaller AE indicates better attack performance.

3.5.5. IMPLEMENTATION

We develop SecureGaze using TensorFlow and Adam optimizer [97]. We use a simple auto-encoder to implement M_θ, which is similar to that used in [125]. Before performing

Table 3.3: Backdoor identification performance on MPIIFaceGaze and GazeCapture for different attacks. SecureGaze can identify the backdoored gaze estimation models on two datasets with over 92% accuracy.

Attack	MPIIFaceGaze					GazeCapture				
	TP	FP	FN	TN	Acc	TP	FP	FN	TN	Acc
BadNets	20	3	0	17	92.5%	20	2	0	18	95.0%
IADA	20	3	0	17	92.5%	19	2	1	18	92.5%
Clean Label	20	3	0	17	92.5%	20	2	0	18	95.0%
WaNet	20	3	0	17	92.5%	20	2	0	18	95.0%
IBA	20	3	0	17	92.5%	20	2	0	18	95.0%

Table 3.4: The ROC-AUC scores of different backdoor identification methods when evaluating on MPIIFaceGaze and GazeCapture with different attacks. SecureGaze outperforms both Gaze-NC and Gaze-FRE significantly.

Attack	MPIIFaceGaze			GazeCaptrue		
	Gaze-NC	Gaze-FRE	SecureGaze	Gaze-NC	Gaze-FRE	SecureGaze
BadNets	0.400	0.561	**0.995**	0.417	0.528	**0.995**
IADA	0.311	0.512	**1.000**	0.605	0.531	**1.000**
Clean Label	0.002	0.444	**1.000**	0.026	0.461	**1.000**
WaNet	0.828	0.508	**0.995**	0.630	0.601	**0.967**
All	0.385	0.506	**0.998**	0.419	0.530	**0.986**

the reverse engineering, we pre-train M_θ on the benign dataset $\mathcal{D}_{be}$ for 5,000 steps with the learning rate of 0.001. We train M_θ for 2,000 steps with a batch size of 50 and the learning rate of 0.0015. We set $\lambda_1 = 20$, $\lambda_2 = 800$. For backdoor mitigation, we fine-tune the gaze estimation models using a batch of 50 benign and 50 reverse-engineered poisoned images for 300 iterations. We use ResNet18 [92] (without the dense layer) to implement F, and a dense layer without activation function to implement H.

3.5.6. BACKDOOR IDENTIFICATION PERFORMANCE

We evaluate backdoor identification performance on 200 backdoored and 40 benign gaze estimation models. Specifically, for each dataset, we first train 20 benign models and then train 20 backdoored models for each attack. It is important to note that although the 20 backdoored (or benign) models for each attack-dataset combination are trained on the same training dataset, they have different parameters and exhibit variations in performance due to two key factors: 1) Each model is randomly initialized with different parameters; 2) During training, image batches are randomly sampled in each iteration, introducing variability in the training process. This is a standard evaluation protocol used in existing works [112], [146].

Evaluation results. We report the backdoor identification results of SecureGaze in Table 3.3, which indicate that SecureGaze can identify backdoored gaze estimation models trained by both input-independent and input-aware attacks, on MPIIFaceGaze and GazeCapture, with an average accuracy of 92.5% and 94.5%, respectively. Specifically, TP and FN remain consistent across most evaluation scenarios, with TP being 20 and FN

Table 3.5: Performance of SecureGaze in backdoor mitigation. SecureGaze shows larger AE and smaller DAE on two datasets, which demonstrates good performance in backdoor mitigation for various attacks.

	MPIIFaceGaze				**GazeCapture**			
	Undefended		SecureGaze		Undefended		SecureGaze	
Attack	AE	DAE	AE	DAE	AE	DAE	AE	DAE
BadNets	3.25	14.8	17.2	3.59	1.09	20.0	17.7	3.65
IADA	3.19	14.4	15.6	3.50	1.54	10.6	10.2	3.77
Clean Label	0.72	15.4	16.4	2.51	2.45	9.85	10.9	3.20
WaNet	1.31	15.9	15.3	3.29	2.51	9.55	9.57	3.66
IBA	3.04	14.4	14.2	4.12	0.91	19.4	19.2	3.90

being 0. This indicates that SecureGaze successfully identifies all 20 backdoored gaze estimation models without any false negatives. Moreover, TN and FP are identical for each backdoor attack. This consistency arises because the set of benign gaze estimation models, which are attack-free and independent of backdoor attack, remains the same across all five scenarios, and thus, SecureGaze leads to the same identification results, in FP and TN, regardless of the attacks.

Discussion on failure cases. For FN cases, SecureGaze struggles to identify a trigger function that produces similar gaze directions, instead prioritizing the minimization of perturbations. By contrast, for FP cases, SecureGaze reverse-engineers a trigger function that maps different inputs close to the target gaze direction but neglects the magnitude of perturbations introduced to benign images. We believe this issue arises from using fixed values for λ_1 and λ_2, where FN cases require larger values, while FP cases benefit from smaller values. A potential solution is to dynamically adjust λ_1 and λ_2. For instance, we can initially increase their values to ensure the trigger function generates similar outputs across different inputs, then gradually decrease them to focus on minimizing perturbations.

Comparison with state-of-the-art defenses. Table 3.4 shows the ROC-AUC scores of SecureGaze, Gaze-NC, and Gaze-FRE for different backdoor attacks on two datasets. We also report the scores when applying the various attacks simultaneously. As shown, the score of SecureGaze is above 0.96 in all the examined cases, which is significantly higher than that of Gaze-NC and Gaze-FRE. Besides, we notice that Gaze-FRE fails to find a trigger function that enables the backdoored gaze estimation model to map different inputs to similar gaze directions. This observation confirms our analysis that the feature-space characteristics for backdoored classification models [112] do not hold for backdoored gaze estimation models.

3.5.7. BACKDOOR MITIGATION PERFORMANCE

Evaluation results. We train backdoored gaze estimation models by each considered attack on each dataset. The backdoor mitigation results of SecureGaze are shown in Table 3.5, which indicate that SecureGaze can mitigate backdoor behaviors for various attacks on two datasets. Specifically, SecureGaze can substantially increase AE, indicating that the output gaze directions for poisoned inputs deviate significantly from the target gaze

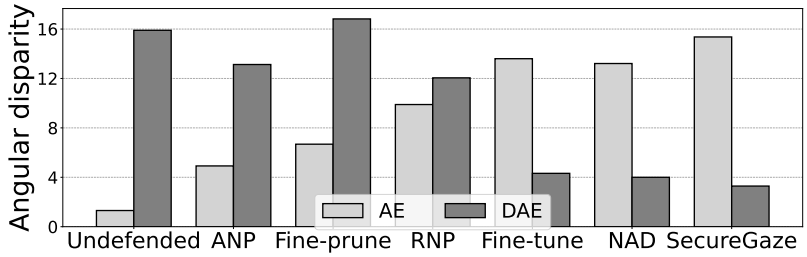


Figure 3.13: Performance of different defenses on backdoor mitigation. SecureGaze outperforms compared methods, i.e., undefended, ANP, Fine-prune, RNP, Fine-tune, and NAD.

direction after backdoor mitigation. Additionally, SecureGaze significantly reduces DAE, which means that the mitigated backdoored gaze estimation models perform normally and output gaze directions close to correct gaze annotation, even though triggers are injected into the inputs.

Comparison with state-of-the-art defenses. We compare SecureGaze with ANP, RNP, NAD, Fine-prune, and Fine-tune on backdoor mitigation. Specifically, we train a backdoored gaze estimation model by WaNet on MPIIFaceGaze and apply different methods to mitigate the backdoor behavior. The evaluation results are shown in Figure 3.13. The AE for SecureGaze is significantly larger than that for other methods, while the DAE for SecureGaze is much smaller than that for other methods, which shows the superiority of SecureGaze on backdoor mitigation. Moreover, Fine-prune, ANP, and RNP, which prune compromised neurons, perform poorly on backdoored gaze estimation models. This supports our analysis that the feature-space characteristics of backdoored gaze estimation models differ from those of backdoored classification models, making it ineffective to target specific neurons for backdoor mitigation.

3.5.8. SYSTEM PROFILING

We measure the latency and memory usage of SecureGaze during two key processes: reverse-engineering the trigger function for backdoor identification and fine-tuning the model for backdoor mitigation. These measurements are conducted on a desktop with an NVIDIA GeForce RTX 3080 Ti GPU and an Intel i7-12700KF CPU, following the implementation details outlined in Section 3.5.5.

Latancy. For reverse-engineering the trigger function, we repeat the process five times for a given gaze estimation model. The average latency for reverse engineering is 12 minutes. For backdoor mitigation, we repeat the experiments five times with a given gaze estimation model and a reverse-engineered trigger function. The average latency for backdoor mitigation is 100 seconds.

Memory usage. We measure the memory specifically allocated to the training process, i.e., training M_θ or fine-tuning the gaze estimation model. This is determined by subtracting the memory usage before training from the peak memory usage during training. Specifically, reverse-engineering the trigger function requires approximately 9,970 MB of memory, while fine-tuning the gaze estimation model consumes around 6,000 MB.

Table 3.6: The impact of FSO, λ_1, λ_2, and p on backdoor identification performance.

Metric	Different λ_1			Different λ_2			Different p			w/o FSO
	10	20	30	600	800	1000	5%	10%	15%	
TP	20	20	19	20	20	20	20	20	20	20
FP	3	3	3	11	3	3	4	3	3	20
FN	0	0	1	0	0	0	0	0	0	0
TN	17	17	17	9	17	17	16	17	17	0
Acc	92.5%	92.5%	90%	72.5%	92.5%	92.5%	90%	92.5%	92.5%	50%

Table 3.7: Backdoor identification and mitigation performance of SecureGaze on various datasets and regression tasks.SecureGaze is effective across various datasets and regression tasks.

Dataset	**Backdoor identification**					**Backdoor mitigation**			
	TP	FP	FN	TN	Acc	Undefended		SecureGaze	
						AE	DAE	AE	DAE
ETHXGaze	20	0	0	20	100%	1.24	44.5	45.4	2.97
Biwi	20	0	0	20	100%	2.48	25.2	27.1	1.10
Pandora	20	0	0	20	100%	0.48	22.8	23.0	2.71

Note that, for a given gaze estimation model, the process of backdoor identification and mitigation needs to be performed offline only once before deployment. As a result, SecureGaze does not introduce additional run-time latency to the gaze estimation model after deployment. Furthermore, since SecureGaze does not need to enumerate all potential targets, it is more efficient than existing reverse-engineering-based techniques that require scanning all labels (e.g., 140 minutes for FRE [112] on ImageNet [147]).

3.5.9. ABLATION STUDIES

We conduct ablation studies to investigate the impact of different design choices on the performance of SecureGaze. We consider WaNet as the attack and evaluate on MPIIFaceGaze.

Impact of weights and the size of the benign dataset. We vary the values of λ_1 and λ_2 in Equation 3.5 from 10 to 30 and from 600 to 800 respectively to investigate their impacts on the performance of backdoor identification. Moreover, we study the impact of the size of $\mathcal{D}_{be}$ on the identification performance by changing the ratio p of the benign dataset to the original whole dataset from 5% to 15%. We report the backdoor identification results in Table 3.6. We observe that the performance of SecureGaze is insensitive to λ_1, as the identification accuracy is almost stable with different λ_1. However, SecureGaze is sensitive to λ_2 and the identification accuracy increases with λ_2, as a larger λ_2 allows the feature-space optimization objective to have a greater contribution to the optimization problem. This observation proves that the proposed feature-space optimization objective is important for backdoor identification. Additionally, as p decreases from 10% to 5%, the identification accuracy and TN decrease, while TP remains stable. This is because, compared to a larger p, it is easier to find a small amount of perturbation that can lead to the backdoor behavior on a smaller p for benign models. However, the

Table 3.8: Results on adaptive attack with different values for β. Adaptive WaNet is less effective than WaNet.

Metric	WaNet	Adaptive WaNet	
		β =0.02	β =1.0
AE	1.50	5.41	10.8
DAE	16.0	14.9	12.8
Acc	92.5%	92.5%	67.5%

identification accuracy is still 90% even when $p = 5\%$.

Impact of feature-space optimization objective (FSO). We study the impact of FSO on the performance of backdoor identification by removing it from $\mathcal{OPT}$-SecureGaze. The results are shown in Table 3.6, which indicates that all the backdoored and benign models are classified as backdoored models. This means that SecureGaze cannot identify backdoor without the FSO. We further observe that without the FSO, SecureGaze cannot find a trigger function that can map different inputs to similar output vectors. As a result, SecureGaze solves the optimization problem by focusing on minimizing the amount of perturbations, which leads to the misclassification of backdoored models.

Generalization to various datasets. We investigate the generalization capability of SecureGaze across different regression tasks by utilizing three additional datasets. These include ETHXGaze [27], a complex gaze estimation dataset with a wider range of head poses and gaze directions, and two datasets for head pose estimation, i.e., Biwi [148] and Pandora [149]. Specifically, ETHXGaze contains more than one million images with a wider range of head poses and gaze directions. Head pose estimation seeks to determine a three-dimensional vector representing the Eular angle (yaw, pitch, roll) from a monocular image. This modality is commonly used in human-computer interaction to infer user attention [150], [151], [152] and for authentication system [153]. Biwi is collected from 24 subjects, and each subject has 400 to 900 images. We use the *cropped faces of Biwi dataset (RGB images)* released by [149]. Pandora has 100 subjects and more than 120,000 images.

For each of these datasets, i.e., ETHXGaze, Biwi Kinect, and Pandora, we randomly select 80% and 10% of the images without replacement to form the training dataset and the benign dataset, respectively. We train 20 backdoored and 20 benign models on the training dataset for each dataset. For head pose estimation, we set $\lambda_1 = 10$, $\lambda_2 = 100$, and $\epsilon = 0.05$, using average L_1 error to define AE and DAE, rather than average angular error. The evaluation results for backdoor identification and mitigation are shown in Table 3.7, which demonstrates that SecureGaze is effective across various datasets and regression tasks in human-computer interaction.

3.5.10. ADAPTIVE ATTACK

When the attacker has the full knowledge of SecureGaze, one potential adaptive attack that can bypass our method is to force RAV to be close to 1 to break the feature-space characteristics. Based on this intuition, we design an adaptive attack that adds an additional loss term L_{adp} with a weight β to the original loss function of the chosen attack to

Table 3.9: The average attack error (in degree) for subjects with physical triggers before and after backdoor mitigation.

Model	Subject 1	Subject 2	Subject 3	Subject 4
Before mitigation	1.71	1.07	0.98	1.17
After mitigation	15.9	16.6	10.3	7.6

enforce RAV to be close to one. We define L_{adp} as:

$$L_{adp} = \left|1 - \frac{1}{d}\sum_{j=1}^{d} \frac{\sigma^2\left(\{\mathcal{B}(F(\mathcal{A}(x_i)), w_j)\}_{i=1}^{N_p}\right)}{\sigma^2\left(\{\mathcal{B}(F((x_i), w_j))\}_{i=1}^{N_b}\right)}\right|, \tag{3.11}$$

where N_p and N_b are the numbers of poisoned and benign inputs in a minibatch. We consider two values for β, i.e., 0.02 and 1.0. We train 20 backdoored models by incorporating L_{adp} into WaNet for each considered value of β.

Table 3.8 shows the identification accuracy and the averaged AE and DAE over 20 backdoored models. As shown, the AE of the adaptive attack is much higher than that of WaNet and it increases as β raises. This proves that the feature-space characteristics we observed of backdoored gaze estimation models are vital to result in the backdoor behavior. Moreover, the adaptive attack with $\beta = 0.02$ cannot reduce the identification accuracy as the feature-space characteristics are not totally broken. When increasing β to 1.0, the identification accuracy drops to 67.5%. However, the AE is higher than 10.0 with $\beta = 1.0$, in which we believe the attack is ineffective.

3.5.11. PHYSICAL-WORLD BACKDOOR DEFENSE

Below, we apply SecureGaze to mitigate the backdoor behavior of the backdoored gaze estimation model we considered in Section 3.3.2, in which a physical item, i.e., a piece of white tape on the face, can effectively trigger the backdoored behaviors. We record the estimated gaze from the mitigated model for each subject by the gaze estimation pipeline in Figure 3.4 when the physical trigger is present on the face. Figure 3.14 visualizes the estimated gaze directions, while Table 3.9 quantifies the average attack error, comparing results before and after mitigation. Specifically, before mitigation, the estimated gaze directions (green dots) are concentrated around the attacker-chosen target (presented by the red star), exhibiting a small average attack error. By contrast, after backdoor mitigation using SecureGaze, the gaze estimations (blue dots) form four clusters corresponding to the four corners where the stimulus appeared, resulting in a significantly higher average attack error than before mitigation. Moreover, we also plot the gaze directions estimated by the backdoored model from subjects without triggers (yellow dots) in Figure 3.14, which overlap with the estimations for subjects with physical triggers after mitigation (blue dots), indicating that SecureGaze can effectively mitigate the backdoor behavior. A video demo that compares behaviors of the backdoored and backdoor-mitigated models can be found in our GitHub repository: `https://github.com/LingyuDu/SecureGaze`.

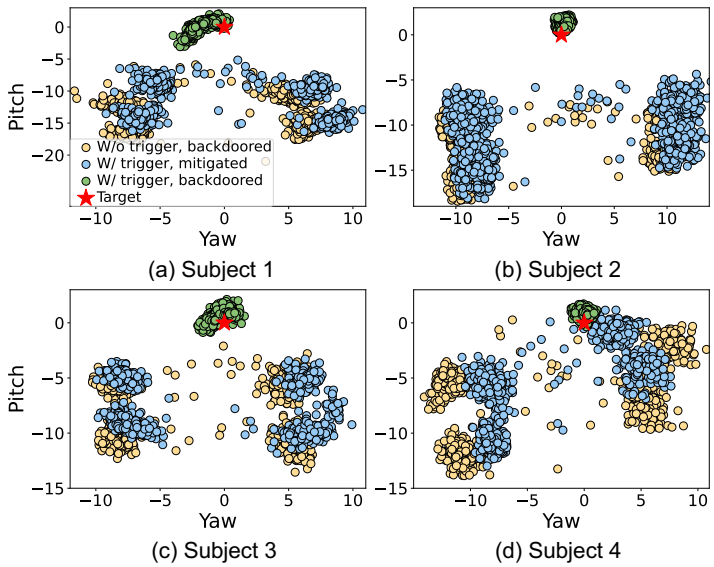


Figure 3.14: Gaze directions estimated by the backdoored gaze estimation model before (green dots) and after (blue dots) backdoor mitigation using SecureGaze.

3.5.12. LIMITATIONS AND FUTURE WORK

Limitation. Similar to existing reverse-engineering-based methods [112], [146], the current design of SecureGaze adopts a fixed threshold for backdoor identification, which may be less effective if the attacker employs a large trigger. Moreover, while we have investigated the impact of different hyperparameter values on the performance of SecureGaze, our analysis is limited to a few scenarios rather than an exhaustive search across a broader range of cases.

Future research directions. A promising avenue for future research involves extending SecureGaze to a wider range of regression models with continuous output spaces that are adopted in human-computer interactions. Another interesting direction is to generalize SecureGaze to more complex threat scenarios, e.g., the gaze estimation models are backdoored by multiple trigger functions associated with multiple target gaze directions. Additionally, exploring more adaptive attacks could provide deeper insights into the robustness and limitations of SecureGaze, enabling a more comprehensive investigation into backdoor defenses tailored specifically for gaze estimation models.

3.6. CONCLUSION

In this chapter, we present SecureGaze to answer the **Sub-Question 2** on defending gaze estimation models against backdoor attacks to improve the trustworthiness of the pretrained paradigm. We identify the unique characteristics of backdoored gaze estimation models, based on which we introduce a novel suite of techniques to reverse engineer the trigger function for backdoored gaze estimation models without the need to

enumerate all the outputs. Our comprehensive experiments in both digital and physical worlds show that SecureGaze is consistently effective in defending gaze estimation models against six backdoor attacks that are triggered by input-aware patterns, input-independent patterns, and physical objects. We also adapt seven state-of-the-art classification defenses, showing that they are ineffective for gaze estimation, while SecureGaze consistently outperforms them.

4

PROTECTING USER PRIVACY IN GAZE ESTIMATION SERVICES

In the previous two chapters, we explored scenarios where the developers deploy either self-trained or pre-trained gaze estimation models to build applications. In Chapter 2, we introduced a framework to reduce the resource cost associated with self-trained models, while in Chapter 3, we enhance the trustworthiness of pre-trained models by presenting an approach to defend against backdoor attacks. This chapter explores a different scenario: calling gaze estimation services by sharing the full-face images with the service provider, who then returns the estimated gaze directions. While this scenario allows the developer to reduce the efforts and resource consumption on gaze estimation, it raises severe privacy concerns as full-face images contain sensitive personal attributes.

In this chapter, we focus on addressing the privacy concerns in gaze estimation services. We present the first approach that can effectively preserve users' privacy in black-box gaze tracking services without compromising gaze estimation performance. Specifically, we proposed a novel framework to train a privacy preserver that converts full-face images into obfuscated counterparts, which are effective for gaze estimation while containing no privacy information. Evaluation on four datasets shows that the obfuscated image can protect users' private information, such as identity and gender, against unauthorized attribute classification. Meanwhile, when used directly by the black-box gaze estimator as inputs, the obfuscated images lead to comparable gaze estimation performance to the conventional, unprotected full-face images.

4.1. INTRODUCTION

With the increasing demand for gaze-based applications, many vendors now offer accessible gaze estimation services [34], [35], [36], [37], [38], [39]. In these services, users share their full-face images with the service provider, who then takes the images as inputs for gaze estimation. However, most of these services are operated by commercial entities, making the gaze estimation system an opaque black box to users. When querying the gaze estimation services, users do not have any knowledge of how their face images are being processed, stored, and utilized. This problem becomes even more concerning given that facial images contain rich information about users' private attributes, such as identity and gender. Thus, when a malicious service provider has access to a large collection of unprotected face images, it can easily infer sensitive user information beyond the intended purpose, posing significant privacy threats to the users [141].

The privacy implications of gaze estimation have started to attract attention from the pervasive computing and eye-tracking communities [154], [155]. A large body of work [156], [157], [158], [159] focuses on the design of obfuscation techniques. These methods aim to eliminate sensitive user information from eye images to prevent iris-based user re-identification without compromising the utility of the estimated gaze data for downstream applications. Although these works have made significant progress in designing privacy-aware gaze-based applications, the question of how to preserve users' privacy in black-box gaze estimation services remains an open challenge.

Besides the methods specifically designed for eye tracking, many privacy-preserving techniques have been developed for general-purpose image recognition and detection tasks. A popular solution is to train an encoder-based feature extractor via adversarial learning, so that it can capture task-related features from the original images while eliminating features related to the user's private attributes [160], [161], [162], [163]. However, these works assume the privacy protector and the user have the full knowledge of the deep learning models utilized by the service provider, which is impractical for the real-world black-box gaze estimation services we considered. Another line of work adds small perturbations in the face images to obstruct unauthorized deep learning models from inferring private user attributes [50], [164]. However, the obfuscated images generated by these methods still contain a substantial amount of private user information that can be visually recognized by human eyes.

In this chapter, we propose PrivateGaze, the first approach that can effectively preserve user's privacy in black-box gaze estimation services without compromising the estimation performance. An overview of PrivateGaze is shown in Figure 4.1. The core component is the proposed *privacy preserver*, which operates on the user's side to convert privacy-sensitive full-face images into privacy-enhanced obfuscated images that remain effective for gaze estimation. PrivateGaze comprises a suite of techniques we developed to tackle two major challenges.

First, the privacy preserver should eliminate features related to the user's private attributes, i.e., identity and gender, from the original full-face images. To resolve this challenge, we introduce a novel method that generates an average full-face image from a public dataset and leverages it as a template to transform images of different users, ensuring that the transformed versions exhibit a similar facial appearance akin to the average full-face image. Second, to maintain good gaze estimation performance, we need to

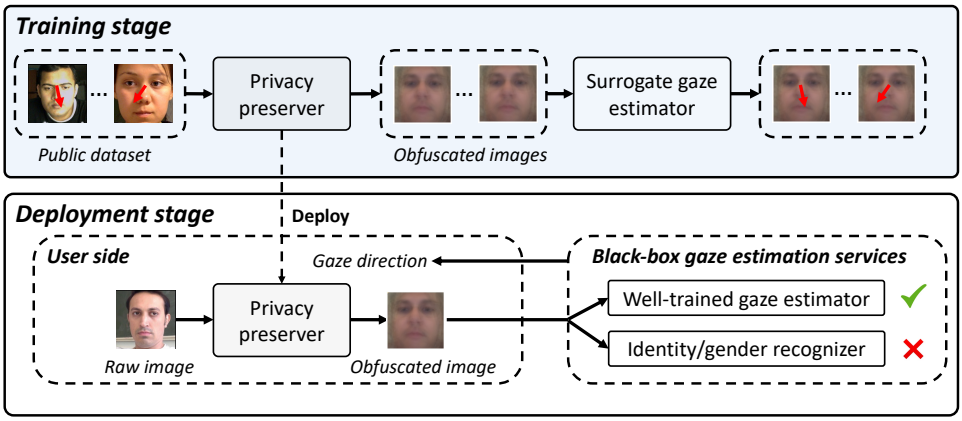


Figure 4.1: An illustration of PrivateGaze, a framework to preserve users' privacy when they are using black-box gaze estimation services. The core of PrivateGaze is the privacy preserver, which transforms the original privacy-sensitive full-face image into an obfuscated version as input for the untrusted gaze estimation services. During the training stage, we train the privacy preserver with the assistance of a pre-trained surrogate gaze estimator. After training, the privacy preserver is deployed on the user's device to generate obfuscated images that can be used by the black-box gaze estimation services. This ensures accurate gaze estimation while preventing the user's private attributes, such as gender and identity, from being inferred by the service provider.

ensure that the essential gaze-related information in the original images is preserved in the obfuscated images. Moreover, the obfuscated images should be readily compatible with the black-box gaze estimator, requiring no additional adaptation from the service provider. To achieve this goal, we train a *surrogate gaze estimator* on a public dataset. As shown in Figure 4.1, we leverage the well-trained surrogate gaze estimator to encourage the privacy preserver to generate obfuscated images that contain features leading to the same gaze direction as the original images. In summary, our major contributions are three-fold:

- We propose PrivateGaze, the first approach that can effectively preserve users' privacy when using black-box gaze tracking services without compromising the gaze estimation performance.

- We propose a novel framework to train a privacy preserver that can be deployed on the user's side to convert privacy-sensitive full-face images into privacy-enhanced obfuscated counterparts. The obfuscated images are effective for gaze estimation while containing no information about the user's private attributes.

- We conduct extensive experiments on four benchmark datasets to demonstrate the effectiveness of PrivateGaze. The results show that the obfuscated images produced by PrivateGaze can effectively protect users' private information, such as identity and gender, against unauthorized attribute inference. This protection remains robust even when the malicious attribute recognizer is trained on obfuscated images with correct attribute labels. Moreover, when used directly by the black-box gaze estimator as inputs, the obfuscated images achieve comparable tracking performance to the conventional, unprotected full-face images. Lastly, our system profiling shows that the proposed pri-

vacy preserver can be deployed on various computation platforms with low system costs in terms of latency and memory usage.

Chapter roadmap. The rest of this chapter is organized as follows. In Section 4.2, we review related work and discuss the research gaps. We then present the detailed design of PrivateGaze in Section 4.3. Subsequently, we evaluate PrivateGaze in Section 4.4, followed by the conclusion in Section 4.5. The implementation of PrivateGaze is available at `https://github.com/LingyuDu/PrivateGaze`.

4.2. RELATED WORK

Below, we review related work on privacy-preserving techniques in the image domain and in eye-tracking systems. We also highlight the gaps that this chapter aims to address.

4.2.1. PRIVACY-PRESERVING METHODS IN THE IMAGE DOMAIN

There have been several approaches to preserving user privacy in images across various application scenarios. For example, Oh et al. [164] employ adversarial image perturbations to raw images to confuse deep learning-based classifiers from recognizing the user's identity. Shan et al. [50] propose adding imperceptible perturbations to images before their release, causing identity recognizers trained on these perturbed images to misidentify normal images. However, these perturbed images still retain private attributes and can be easily identified by recognizers trained on the perturbed dataset. In our work, we pursue a stringent privacy objective where attackers cannot infer private attributes from obfuscated images, even when deep learning classifiers have been trained on these obfuscated images with correct attribute labels.

Previous works [160], [161] have explored approaches where instead of adding perturbations to raw images, they focus on training an encoder to extract features from raw images that are useful for utility tasks while excluding information related to private attributes. For instance, Liu et al. [160] utilize adversarial learning to jointly train an encoder, a task-related model, and a private attribute recognizer. The task-related model and the private attribute recognizer are built upon the features extracted by the encoder from raw images. Wu et al. [162] propose DAPter, a method aimed at preserving user privacy during the utilization of task-related inference services on cloud platforms. However, their approach requires access to the task-related model and involves modifying its parameters for effective training, which is impractical for black-box models in real-world scenarios. Our work differs from these approaches by focusing on a more practical scenario where users have no knowledge of the deep learning model employed by the service provider for gaze estimation.

Face swapping [165], [166], [167], [168], [169], [170], [171], [172], [173] and face de-identification [174], [175], [176], [177] are potential methods to preserve user privacy by altering the identities of subjects in raw images. However, these techniques may retain other user attributes in the synthesized images [168], [171], [172], [176], [177], such as facial expressions and emotions, which are also privacy-sensitive information that users may wish to protect. By contrast, our method exclusively maintains gaze features in the synthesized images, offering robust privacy protection for users when utilizing gaze estimation services. Moreover, we observe that synthesized images generated by

face swapping from target images of different subjects often exhibit distinct appearances [171], [172], [173], and similarly, de-identified images for different subjects may show visual differences [175]. We believe these inherent properties could be exploited by adversaries to classify users' identities if they can stealthily collect some synthesized images. In our approach, synthesized images from different subjects maintain similar appearances, and our experiments show that the adversaries cannot correctly classify users' identities even with access to stealthily collected synthesized images.

4.2.2. PRIVACY-PRESERVING SOLUTIONS FOR EYE-TRACKING SYSTEMS

Privacy-preserving eye tracking has emerged as a significant research topic in recent years due to growing privacy concerns associated with various stages of the eye-tracking pipeline. We categorize existing works into three groups.

The first group focuses on the data collection stage, aiming to protect users' privacy-sensitive data by preventing its transmission to a central server. For example, Elfares et al. [178] utilize federated learning to train a deep learning-based gaze estimator. This approach retains raw data locally with users to preserve privacy and sends only minimal updates necessary for gaze estimation to the central server. Steil et al. [179] propose a method to protect the privacy of users and bystanders from scene images captured by a head-mounted eye tracker. Specifically, they develop a method to disable the scene camera upon detecting privacy-sensitive situations and automatically reactivate it when eye movement patterns change.

The second group of works focuses on preserving user privacy in the gaze estimation stage, aiming to mitigate the presence of private attributes in the images used by gaze estimators. For example, John et al. [156] introduce pixel-level noise to eye images captured by eye-tracking cameras, effectively thwarting iris authentication attacks. Eskildsen et al. [157] propose various methods to obfuscate eye images, including adding noise and applying non-linear low-pass filters, to prevent identification based on iris patterns. Additionally, John et al. [158] propose a hardware-based solution to remove bio-metric information from eye images by inducing optical defocus. This is achieved by increasing the distance between the eyes and the eye-tracking cameras, thereby intentionally blurring the iris region. Bozkir et al. [180] train a support vector regression model to estimate gaze direction from synthetic eye images, thereby preserving personal information for users.

The last group focuses on preserving the private attributes of users contained in the gaze data obtained by gaze estimators. David-John et al. [181] explore various privacy mechanisms such as adding Gaussian noise and temporal downsampling to reduce user identification accuracy based on gaze data features like fixations and saccades. Steil et al. [182] apply differential privacy by adding noise to features extracted from gaze data. They demonstrate that their approach prevents attackers from accurately identifying the user's identity and gender from gaze trajectories, while still maintaining good performance in gaze-based document type recognition tasks. Li et al. [183] propose a framework that directly applies differential privacy to raw gaze data. Their method can integrate with existing eye-tracking ecosystems and operate in real time, enhancing privacy protection during data processing stage.

In this chapter, we focus on addressing privacy concerns during the gaze estimation

stage. We propose a novel method to remove private attributes from full-face images while maintaining comparable performance in gaze estimation for black-box gaze estimation services.

4.3. METHOD

In this section, we introduce a novel method, PrivateGaze, that can convert a normal full-face image into a privacy-perceived full-face image. With the privacy information removed, the privacy-perceived full-face image still contains sufficient information for a black-box gaze estimation service to perform the gaze estimation task. In the following, we first define the threat model to formulate the problem and then detail the design of PrivateGaze.

4.3.1. THREAT MODEL

Black-box gaze estimator. With recent developments in gaze estimation, it has become common to include the full-face image of the user as input to the methods [21], [28]. While existing works in privacy preservation [160], [161], [162] assume the details of the deep neural network used by the service provider are known, we consider a more practical case where the gaze estimator $\mathcal{G}_b(\cdot)$ is performed by a black-box, deep learning-based model. Specifically, the black-box gaze estimator $\mathcal{G}_b(\cdot)$ *is trained on an unknown dataset* $\mathcal{D}_b$ that contains raw full-face images and gaze annotations. Users can only query $\mathcal{G}_b(\cdot)$ for service and have no knowledge about its implementation and training details.

Privacy concerns. The end-to-end gaze estimation system, including the processing pipeline and the deep learning-based gaze estimation model, makes the gaze estimation services untrustworthy. The full-face image of the user can be illegally used for purposes beyond gaze estimation, such as classifying the user's private attributes like identity and gender. Therefore, the user would like to remove the private information contained in the full-face images before using them to call the gaze estimation services, without sacrificing gaze estimation performance.

Capabilities and goals of the malicious service provider. We assume the malicious service provider can stealthily collect a dataset $\mathcal{D}_p$, comprising images submitted by users for gaze estimation service, along with annotations of private user attributes such as identity and gender. Specifically, the service provider receives raw images when users do not adopt privacy protection techniques, and obfuscated images when they do. $\mathcal{D}_p$ is used to train classifiers aimed at discerning users' private attributes from images that do not belong to $\mathcal{D}_p$.

Our goals. In this work, we envision a trustworthy party that provides a privacy preserver $\mathcal{P}(\cdot)$ to protect the user's privacy. As shown in Figure 4.1, during the deployment stage, $\mathcal{P}(\cdot)$ converts the user's original full-face images x into obfuscated images x' that do not contain information related to the user's attributes, such as identity and gender. The user then directly calls the black-box gaze estimator $\mathcal{G}_b(\cdot)$ with the obfuscated image x'. Formally, the obfuscated image x' must fulfill the objectives of preserving the user's privacy while ensuring good gaze estimation performance:

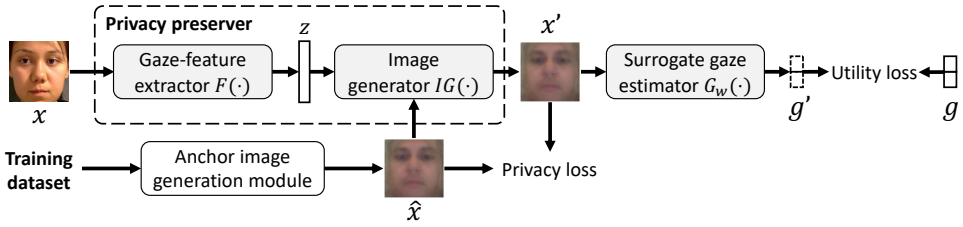


Figure 4.2: An overview of PrivateGaze, which comprises the privacy preserver, the anchor image generation module, and the surrogate gaze estimator $\mathcal{G}_w(\cdot)$ trained on the training dataset $\mathcal{D}_w$. The privacy preserver includes the gaze-feature extractor $F(\cdot)$ and the image generator $IG(\cdot)$. $F(\cdot)$ extracts gaze features z from the raw images x in the training dataset. $IG(\cdot)$ takes z and a pre-generated image $\hat{x}$ as inputs to form the obfuscated images x'. $\hat{x}$ serves as the *anchor image* and is crafted from the training dataset using the proposed anchor image generation module. Subsequently, we compute the *privacy loss* based on $\hat{x}$ and x' to train $\mathcal{P}(\cdot)$ for the privacy objective. x' is then passed to $\mathcal{G}_w(\cdot)$ to obtain the estimated gaze direction g'. Finally, we calculate the *utility loss* based on the gaze annotations g and g' to train the privacy preserver for the utility objective.

- **Privacy goal:** The obfuscated image x' cannot be used to correctly classify private attributes of the user, such as identity and gender, even if the malicious service provider trains deep learning-based classifiers on $\mathcal{D}_p$, i.e., a set of x' with accurate labels for these confidential user attributes.
- **Utility goal:** The obfuscated image x' can be directly used by $\mathcal{G}_b(\cdot)$ without any adaptation needed from the service provider's side. The gaze estimation performance of $\mathcal{G}_b(\cdot)$ with x' should be similar to the original full-face images.

Assumption. We assume a public gaze estimation dataset $\mathcal{D}_w$ is available, which contains training examples (x_i, g_i), where x_i is the full-face image and g_i is the corresponding gaze annotation. The dataset $\mathcal{D}_w$ will be used to train $\mathcal{P}(\cdot)$. Note that $\mathcal{D}_w$ is different from $\mathcal{D}_b$, as we do not know which dataset has been used by the service provider to train the black-box gaze estimator $\mathcal{G}_b(\cdot)$.

4.3.2. OVERVIEW OF PRIVATEGAZE

To achieve the design goals, we propose a novel framework PrivateGaze consisting of a privacy preserver, an anchor image generation module, and the surrogate gaze estimator as shown in Figure 4.2. The privacy preserver $\mathcal{P}(\cdot)$ converts unprotected raw images x into obfuscated images x' to protect the private information of users, such as gender and identity contained in x. To achieve this goal, the privacy preserver ensures that x', converted from different x (images from different subjects), will exhibit similar facial appearances akin to a pre-generated average full-face image called the anchor image $\hat{x}$. We devise the anchor image generation module to generate the $\hat{x}$ from the $\mathcal{D}_w$.

To achieve the utility goal, the privacy preserver $\mathcal{P}(\cdot)$ is designed to extract gaze features z from x and generate x' that maintains these features for effective gaze estimation. Specifically, $\mathcal{P}(\cdot)$ consists of the gaze-feature extractor $F(\cdot)$ and the image generator $IG(\cdot)$. $F(\cdot)$ extracts gaze features z from the input x. $IG(\cdot)$ takes z along with $\hat{x}$ as inputs to generate the privacy-preserved x'. The generated x' has a similar appearance to $\hat{x}$ while preserving the gaze-related information z from x for accurate gaze estimation.

Algorithm 1 Anchor image generation

Input: $\mathcal{D}_w$, $\mathcal{G}_b(\cdot)$, $\mathcal{G}_w(\cdot)$, $K_1 = 15$, $m = 1$, $M = 35$, $N = 500$, and a list of candidate anchor image $Anc = \{\}$.

1: Randomly sample N images from $\mathcal{D}_w$ to form $\{x_i\}_{i=1}^N$;
2: Query both $\mathcal{G}_b(\cdot)$ and $\mathcal{G}_w(\cdot)$ with $\{x_i\}_{i=1}^N$ to obtain the list $\{D_g(x_i)\}_{i=1}^N$, where $D_g(x_i) = |\mathcal{G}_w(x_i) - \mathcal{G}_b(x_i)|_1$;
3: Sort $\{D_g(x_i)\}_{i=1}^N$ in the ascending order;
4: Obtain the set of raw images after sorting $\{x_k\}_{k=1}^N$, where $D_g(x_k) \leq D_g(x_{k+1})$;
5: **while** $K_1 + m \leq M$ **do**
6: Calculate the average full-face image $Ave(m) = \frac{1}{m}\sum_{k=1}^{m} x_k$;
7: Anc.append($Ave(m)$);
8: $m \leftarrow m + 1$;
9: **end while**
10: Query $\mathcal{G}_b(\cdot)$ and $\mathcal{G}_w(\cdot)$ with $Anc = \{Ave(m)\}_{m=1}^M$ to obtain the list $\{D_g(Ave(m))\}_{m=1}^M$;
11: $\hat{x} = Ave(\hat{m})$, where $\hat{m} = \arg\min_m \{D_g(Ave(m))\}_{m=1}^M$;

Output: The anchor image $\hat{x}$.

To train $\mathcal{P}(\cdot)$, we construct a surrogate gaze estimator $\mathcal{G}_w(\cdot)$ trained on the $\mathcal{D}_w$, which performs the gaze estimation training with input x'. In this way, we are able to maximize the information in x' for the gaze estimation task.

4.3.3. ANCHOR IMAGE GENERATION MODULE

Below, we present a novel method for generating the anchor image $\hat{x}$ from a public dataset. The anchor image serves as a template for the obfuscated images x', ensuring they exhibit a facial appearance similar to $\hat{x}$. This allows us to manipulate the appearances of x' to preserve user's privacy while achieving the utility goal.

A major challenge in achieving this utility goal is training $\mathcal{P}(\cdot)$ with the surrogate gaze estimator $\mathcal{G}_w(\cdot)$, while aiming for good gaze estimation performance on the black-box gaze estimator $\mathcal{G}_b(\cdot)$. To address this challenge, we carefully generate the anchor image $\hat{x}$ to ensure that both $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ yield similar gaze estimation results on $\hat{x}$. This strategy enables $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ to achieve comparable gaze estimation performance on the obfuscated images x', as they share similar appearances with the anchor image.

In detail, $\hat{x}$ is an average full-face image created by blending facial images selected from the training dataset. The design ensures that x' synthesized from $\hat{x}$ does not closely resemble any individual subject. Then, we design the method to obtain $\hat{x}$ that can lead to similar gaze estimation results from $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$. Specifically, we first randomly sample N raw images $\{x_i\}_{i=1}^N$ from the $\mathcal{D}_w$ and use them to query both $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$, where the two gaze estimators return the corresponding gaze estimation results $\{\mathcal{G}_w(x_i)\}_{i=1}^N$ and $\{\mathcal{G}_b(x_i)\}_{i=1}^N$, respectively. We then calculate the $L1$ norm between the gaze directions estimated by $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ for each image x_i to obtain the list $\{D_g(x_i)\}_{i=1}^N$, where $D_g(x_i) = |\mathcal{G}_w(x_i) - \mathcal{G}_b(x_i)|_1$. After that, we sort the list $\{D_g(x_i)\}_{i=1}^N$ in the ascending order. We use $\{x_k\}_{k=1}^N$ to denote the set of raw images after sorting, which satisfies $D_g(x_k) \leq D_g(x_{k+1})$. We use $Ave(m)$ to denote the average full-face image calculated from the first

m images in $\{x_k\}_{k=1}^{N}$ by $Ave(m) = \frac{1}{m}\sum_{k=1}^{m} x_k$. We generate M average full-face images by varying m from K_1+1 to K_1+M, where $K_1 > 1$ and $K_1 + M \leq N$. Finally, we query $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ with the M average full-face images and select the one that leads to the minimum L_1 norm between the outputs of $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ as the anchor image $\hat{x}$. We set $K_1 = 15$, $M = 35$, and $N = 500$. The number of queries for $G_b(\cdot)$ is determined by N. The parameters K and M determine the minimum and maximum number of full-face images utilized in generating the anchor image, respectively. A higher value of K ensures the anchor image to be distinct from any single subject, whereas a larger M may include undesired full-face images, potentially resulting in significantly different gaze estimation results for $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ during the anchor image generation process. We show the $\hat{x}$ obtained from GazeCapture dataset [21] in Figure 4.3 and summarize the procedure of generating the anchor image in Algorithm 1.

4.3.4. GAZE-FEATURE EXTRACTOR

To ensure the obfuscated images x' are effective for gaze estimation, $\mathcal{P}(\cdot)$ extracts the gaze features z from x using the gaze-feature extractor $F(\cdot)$. As shown in the left part of Figure 4.3, $F(\cdot)$ comprises the gaze-aware encoder $E(\cdot)$ and the gaze projector $G(\cdot)$. Specifically, $E(\cdot)$ takes x as input and outputs a feature map z. To encourage $E(\cdot)$ in capturing the most essential and meaningful gaze-related features, z is fed into a nonlinear gaze projector $G(\cdot)$ to estimate gaze direction. We define the gaze estimation loss in training $E(\cdot)$ and $G(\cdot)$ as follow:

$$\mathcal{L}_g = \sum_{(x_i, g_i)\in\mathcal{D}_w} \ell(G(E(x_i)), g_i), \tag{4.1}$$

where $\ell(\cdot)$ is the L_1 loss function. The $G(\cdot)$ will be discarded in the deployment stage, and only the gaze features z are sent to the image generator $IG(\cdot)$.

4.3.5. IMAGE GENERATOR

The image generator $IG(\cdot)$ is designed to synthesize the obfuscated images x' from gaze features z such that x' can be effectively used for gaze estimation by $\mathcal{G}_b(\cdot)$ while not containing private attributes from z. To achieve this goal, $IG(\cdot)$ generates x' with appearances similar to the anchor image $\hat{x}$ while preserving the gaze features extracted from x. Specifically, $IG(\cdot)$ takes both z and $\hat{x}$ as inputs to generate x'.

The structure of $IG(\cdot)$ is depicted in the right part of Figure 4.3, which adopts an encoder-decoder architecture. The encoder comprises several convolutional blocks (Conv Block) and takes $\hat{x}$ as input to produce a feature map $\hat{z}$ that has the same spatial dimension as the gaze features z extracted by the gaze-aware encoder $E(\cdot)$. $\hat{z}$ and z are concatenated and then fed into the decoder. Similar to the encoder, the decoder comprises several up-convolutional blocks (Up-conv Block). Each up-convolutional block involves upsampling the feature map followed by several convolutional layers. In our current design, each convolutional block consists of two convolutional layers, and each up-convolutional block has three convolutional layers. As shown in Figure 4.3, to ensure that x' can closely resemble $\hat{x}$, we concatenate the output of each up-convolutional block with the corresponding feature map from the encoder. The resulting combined feature map is used as the input for the next up-convolutional block in the sequence.

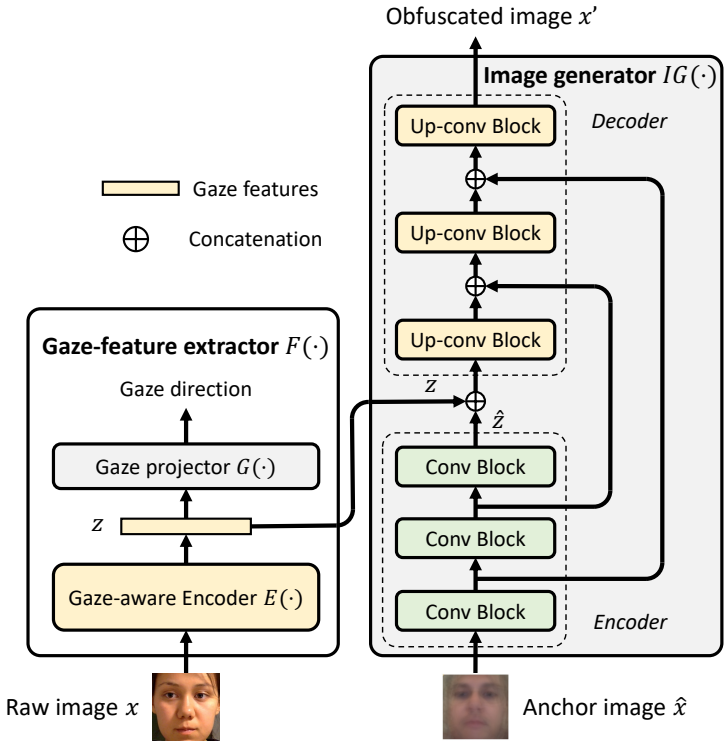


Figure 4.3: The overall design of the privacy preserver $\mathcal{P}(\cdot)$, which consists of the gaze-feature extractor $F(\cdot)$ and the image generator $IG(\cdot)$. $F(\cdot)$ extracts gaze features z from the raw image x of the user. $IG(\cdot)$ takes the extracted gaze features z and the anchor image $\hat{x}$ as inputs to generate the privacy-preserved obfuscated image x'. x' has a similar appearance to $\hat{x}$ while retaining the gaze features extracted from x. Only the components with color-coded yellow will be deployed on the user's device after training for privacy preservation.

We denote the process of generating x' from z and $\hat{x}$ as $x' = IG(\{z, \hat{x}\})$. Note that in Figure 4.3, we use three convolutional blocks and three up-convolutional blocks to illustrate the structure of $IG(\cdot)$. However, in practice, the number of these convolutional and up-convolutional blocks can vary based on design considerations and we are using four convolutional blocks and four up-convolutional blocks in our current design.

4.3.6. TRAINING OF THE PRIVACY PRESERVER

First, to achieve the privacy objective, it is crucial that x', generated from different raw images, maintains a uniform appearance similar to $\hat{x}$. Therefore, we define the privacy loss as follows:

$$\mathcal{L}_{privacy} = \sum_{(x_i, g_i) \in \mathcal{D}_w} 1 - \text{MS-SSIM}\left(\hat{x}, IG\left(\{E(x_i), \hat{x}\}\right)\right), \tag{4.2}$$

where MS-SSIM$(\cdot,\cdot)$ is a function that calculates the multi-scale structural similarity [184], measuring the similarity between two images with values in the range of $[0,1]$. A larger value of MS-SSIM indicates greater similarity between the two images.

Second, to achieve the utility objective, we utilize x' as input to $\mathcal{G}_w(\cdot)$ to obtain the estimated gaze direction g' for x', and then train $\mathcal{P}(\cdot)$ to ensure g' closely approximates g. $\mathcal{G}_w(\cdot)$ is trained on raw images, and its parameters are frozen during the training of $\mathcal{P}(\cdot)$. Besides, the gaze estimation loss $\mathcal{L}_g$, defined in Equation 4.1, encourages the extraction of gaze features from raw images, thereby contributing to the utility objective. For training the privacy preserver to achieve the utility objective, we define the following utility loss:

$$\mathcal{L}_{utility} = \sum_{(x_i, g_i) \in \mathcal{D}_w} \ell\left(\mathcal{G}_w\left(IG\left(\{E(x_i), \hat{x}\}\right)\right), g_i\right) + \mathcal{L}_g, \tag{4.3}$$

where $\ell(\cdot)$ is the L_1 loss function. The first term of $\mathcal{L}_{utility}$ aims to minimize the L_1 norm between $\mathcal{G}_w(x')$ and g. The second term encourages the extraction of gaze features to generate x'.

Putting them all together, the final optimization objective for training $\mathcal{P}(\cdot)$ is the weighted sum of $\mathcal{L}_{utility}$ and $\mathcal{L}_{privacy}$:

$$\mathcal{L} = \mathcal{L}_{utility} + \lambda \mathcal{L}_{privacy}, \tag{4.4}$$

where λ is the weight that balances the trade-off between the utility and privacy objectives. PrivateGaze optimizes $\mathcal{P}(\cdot)$ by minimizing $\mathcal{L}$. During training, PrivateGaze samples a minibatch from $\mathcal{D}_w$ to calculate $\mathcal{L}$, and trains $\mathcal{P}(\cdot)$ by gradient descent. We summarize the training procedure for $\mathcal{P}(\cdot)$ in Algorithm 2.

4.3.7. DEPLOYMENT OF THE PRIVACY PRESERVER

In the deployment stage, the raw images collected from users are initially transformed into obfuscated images using the trained privacy preserver. Users then use these obfuscated images when calling the black-box gaze estimation services to protect their privacy. To reduce the computational cost, only specific components of the privacy preserver (highlighted in yellow in Figure 4.3) need to be deployed on the user's device. Specifically, for the gaze-feature extractor, only the gaze-aware encoder needs to be deployed, as the image generator exclusively utilizes gaze features from this encoder. For

Algorithm 2 Training algorithm of the privacy preserver

Input: $\mathcal{D}_w$, $\mathcal{G}_w(\cdot)$, and $\hat{x}$ generated by Algorithm 1.

1: Randomly initialize the parameters ψ of $\mathcal{P}(\cdot)$;
2: **for** each training step **do**
3: Sample the minibatch B of N_b images from $\mathcal{D}_w$;
4: Calculate the privacy loss over B by $\mathcal{L}_{privacy} = \frac{1}{N_b}\sum_{(x_i,g_i)\in B} 1 - \text{MS-SSIM}(\hat{x}, IG(\{E(x_i),\hat{x}\}))$;
5: Calculate the utility loss over B by $\mathcal{L}_{utility} = \frac{1}{N_b}\sum_{(x_i,g_i)\in B} \ell\left(\mathcal{G}_w(IG(\{E(x_i),\hat{x}\})), g_i\right) + \mathcal{L}_g$;
6: Update ψ by gradient descent: minimize $\mathcal{L}_{utility} + \lambda\mathcal{L}_{privacy}$;

Output: $\mathcal{P}(\cdot)$ with trained parameters ψ.

the image generator, because the anchor image remains consistent for different raw images, the encoder can be omitted. Instead, the feature maps generated by each convolutional block of the encoder are preserved as *appearance features*. Consequently, deployment requires only the decoder of the image generator and the appearance features to generate obfuscated images.

By deploying only these essential components, computational resources are optimized while achieving the system goal. The evaluation section includes a latency measurement of the privacy preserver, demonstrating its suitability for deployment across various hardware platforms without introducing much computational latency.

4.4. EVALUATION

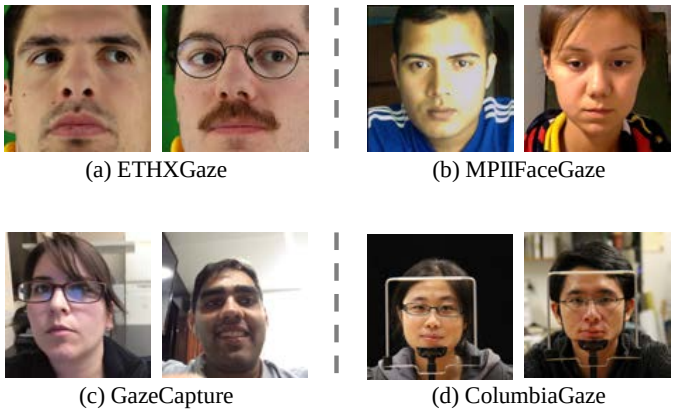
In this section, we conduct a comprehensive evaluation of PrivateGaze. We first introduce the datasets, followed by the methods for comparison and evaluation metrics. We then present the evaluation results on privacy and utility objectives. Next, we conduct ablation studies to investigate the impact of different design choices on the performance of PrivateGaze. Finally, we evaluate system performance, measuring the processing time and memory usage when deploying the proposed privacy preserver on various hardware platforms.

4.4.1. DATASETS

We consider the following four public gaze estimation datasets in our evaluation:

ETHXGaze [27] is a comprehensive dataset collected from 110 subjects in a laboratory environment, showcasing a wide range of head positions, lighting conditions, and individual appearances. It includes one training set and two testing sets. Our evaluation only utilizes the training set, which contains 80 subjects, as it is the only subset with gaze annotations. The images in this set have a resolution of 224×224.

GazeCapture [21] is a large-scale dataset collected from over 1,450 individuals in real-world environments. It includes nearly 2.5 million images taken with the front-facing cameras of smartphones, displaying a wide array of lighting conditions, head poses, user appearances, and backgrounds. In our preprocessing of this dataset, we adopted the



(a) ETHXGaze (b) MPIIFaceGaze

(c) GazeCapture (d) ColumbiaGaze

Figure 4.4: Illustration of images sampled from the four gaze estimation datasets. Our selection of datasets covers a broad spectrum of mobile gaze tracking scenarios: from smartphone usage (GazeCapture) to laptop use cases (MPIIFaceGaze), and to ubiquitous web cameras (ETHXGaze and ColumbiaGaze) that widely appear in many daily devices.

method outlined in [142] to normalize the facial images, initially bringing them to a resolution of 128 × 128. Subsequently, we resized these images to a resolution of 224 × 224.

MPIIFaceGaze [28] is a dataset of full-face images from 15 subjects, including nine males and six females. The images were captured during the participants' routine laptop usage. It includes 3,000 images per subject, featuring a diverse range of head positions, lighting environments, and backgrounds. For our purposes, we utilize the normalized version of the dataset, as released by the authors. This normalized dataset maintains an image resolution of 224 × 224.

ColumbiaGaze [185] was gathered in a controlled laboratory setting, and features data from 56 subjects, including 32 males and 24 females. It is unique for its structured capture of five distinct head poses for each subject. The dataset comprises a total of 105 images per subject, representing combinations of three vertical and seven horizontal gaze directions. In our processing of this dataset, we extract the facial region from the original images and subsequently resize these cropped patches to a uniform resolution of 224 × 224.

Figure 4.4 showcases full-face images sampled from the four datasets under discussion. These images represent a wide range of scenarios in mobile gaze tracking, from the use of front-facing smartphone cameras (GazeCapture) to laptop-based interactions (MPIIFaceGaze). Moreover, the ETHXGaze and ColumbiaGaze datasets represent scenarios where web cameras, commonly found in a variety of ubiquitous devices, are used for gaze tracking.

4.4.2. COMPARISON METHODS

We compare PrivateGaze with the following six methods:

Targeted Projected Gradient Descent Attack (TPGD): In this method, we first feed the

original image x_i from the testing dataset to the surrogate gaze estimator $\mathcal{G}_w(\cdot)$ to output the targeted gaze direction $\overline{g}_i$. We then implement the targeted projected gradient descent attack (TPGD) [186] to create perturbations $\Delta(x_i)$ for the anchor image $\hat{x}$ so that $\mathcal{G}_w(\cdot)$ outputs $\overline{g}_i$ for $\hat{x}+\Delta(x_i)$. This self-created baseline falls under the same category as the proposed PrivateGaze as the targeted attack [146], [187], [188], where the goal of PrivateGaze is to make the surrogate gaze estimator $\mathcal{G}_w(\cdot)$ output a targeted gaze annotation g_i. Finally, we take $\hat{x}+\Delta(x_i)$ as the obfuscated image and feed it to the black-box gaze estimator $\mathcal{G}_b(\cdot)$.

Gaussian Differential Privacy (GauDP): The local model of differential privacy [189] is a potential way to preserve the privacy of user when the central server is not trusted. In this baseline method, we introduce Gaussian noise to each color component $x[p,q,t]$ of every pixel in the raw image x. Specifically, we define the mechanism operating on $x[p,q,t]$ as $M(x[p,q,t]) = x[p,q,t]+\xi$, where $\xi \sim \mathcal{N}(0, Sen^2/\epsilon^2)$, with Sen representing the sensitivity of the color component's value, and ϵ denoting the privacy parameter. Since the value of each color component in the image ranges between 0 and 1, Sen is set to 1. As demonstrated in [190], M satisfies ϵ-Gaussian Differential Privacy. In our evaluation, we explore GauDP with ϵ ranging from 0.1 to 0.5.

Image Pixelization with Differential Privacy (IP-DP): IP-DP [191] is the state-of-the-art differential privacy method operating on images. Specifically, IP-DP first performs pixelization on the raw images, then adds noise sampled from a Laplace distribution to the pixelized images. The Laplace distribution has a mean of 0 and a scale of $\Delta P_b/\epsilon$, where ΔP_b is the global sensitivity of the pixelized images and ϵ is the privacy parameter. We evaluate IP-DP with different values of ϵ, including 0.3, 3.0, and 5.0.

Feature-space Differential Privacy (FS-DP): A variety of state-of-the-art differential privacy techniques [192], [193], [194], [195] add perturbations to features extracted by an encoder from raw images, rather than directly modifying the raw images themselves. We adapt these techniques for gaze estimation by training a variational auto-encoder (VAE) [65] with the surrogate gaze estimator $\mathcal{G}_w(\cdot)$. During training, the VAE takes raw images x as inputs and outputs reconstructed images $\hat{x}$, which are then processed by $\mathcal{G}_w(\cdot)$ to obtain gaze directions g'. In addition to the original VAE loss function, we introduce an extra loss term to optimize for utility, aiming to minimize the L_1 norm between g' and the gaze annotation g. In the deployment stage, the encoder of the VAE extracts d dimensional features f from x. We define the mechanism operating on f as $M(f) = f+\xi$, where ξ is sampled from a Laplace distribution $Lap(\Delta f/(d\epsilon))$. Here, Δf represents the sensitivity of the features and ϵ denotes the privacy parameter. The perturbed features are then fed into the decoder of the VAE to generate the obfuscated images. Following Xue et al. [192], we calculate the sensitivity as $\Delta f = \max_{f_i,f_j} \|f_i - f_j\|_1$, where f_i and f_j are features extracted from different raw images. We examine FS-DP with ϵ ranging from 1.0 to 3.0.

B-DAP: This method is adapted from DAPter [162], which aims to preserve user privacy in a white-box setting where the user possesses full knowledge of the deep learning model used by the service provider. The original DAPter employs a generative model-based image converter to generate obfuscated images. The image converter is trained to minimize an entropy reduction loss for privacy preservation and a task-related loss de-

fined on the outputs of the target model for the utility objective. We adapt this method by using the surrogate gaze estimator $\mathcal{G}_w(\cdot)$ as the target model during the training stage. Specifically, the adapted method B-DAP trains the image converter by minimizing both the entropy reduction loss and the L_1 loss between g_i' and the annotation g_i. During testing, we evaluate the effectiveness of B-DAP using the black-box gaze estimator $\mathcal{G}_b(\cdot)$.

MaxP: A straightforward baseline is to minimize the similarity between the obfuscated and the raw image while ensuring the surrogate gaze estimator can still perform gaze estimation with the obfuscated image. Specifically, we train an auto-encoder that takes the raw image x_i as input and outputs the obfuscated image x_i'. During training, we feed x_i' to the surrogate gaze estimator to obtain the estimated gaze direction g_i'. The auto-encoder is trained to minimize the similarity between x_i and x_i' to achieve the privacy objective, and to minimize the L_1 loss between g_i' and the annotation g_i for the utility objective. In the testing stage, we feed the obfuscated image to the black-box gaze estimator $\mathcal{G}_b(\cdot)$.

4.4.3. EVALUATION SETUP AND METRICS

In the evaluation, we designate ETHXGaze as the unknown dataset $\mathcal{D}_b$ to train the black-box gaze estimator $\mathcal{G}_b(\cdot)$, as it contains the most diverse head poses and gaze variations. Note that, $\mathcal{G}_b(\cdot)$ is trained on the raw images of $\mathcal{D}_b$. We use GazeCapture to train both the privacy preserver $\mathcal{P}(\cdot)$ and the surrogate gaze estimator $\mathcal{G}_w(\cdot)$, with 80% of the images used for training and 20% for validation. We select MPIIFaceGaze and ColumbiaGaze to evaluate the performance of $\mathcal{P}(\cdot)$. Our primary goal is to preserve the private attributes, i.e., identity and gender, of the individuals in these datasets while ensuring that gaze estimation performance remains robust when using the black-box gaze estimator.

For utility measurement, we measure the gaze estimation error, i.e., the average angular error, via the black-box gaze estimator $\mathcal{G}_b(\cdot)$. We take the obfuscated images generated by our PrivateGaze and other baselines as the inputs to assess the utility performance. We conduct this evaluation on both MPIIFaceGaze and ColumbiaGaze datasets. To evaluate the performance in privacy protection, we focus on preserving gender and identity as the two user attributes. For both the MPIIFaceGaze and ColumbiaGaze datasets, we start by randomly selecting 80% of the images that have accurate identity and gender labels. We then apply each of the seven methods under examination, i.e., PrivateGaze, GauDP, IP-DP, FS-DP, MaxP, B-DAP, and TPGD, to generate obfuscated images and form seven corresponding $\mathcal{D}_p$ sets. The $\mathcal{D}_p$ is used to train an identity recognizer and a gender recognizer. Lastly, we apply each of the seven methods on the remaining 20% of images to create corresponding testing set comprising obfuscated images. We report the recognition accuracy on this testing set for performance evaluation.

4.4.4. IMPLEMENTATION

To demonstrate the generalization of PrivateGaze, we employed various neural network architectures to construct the black-box gaze estimator $\mathcal{G}_b(\cdot)$ including ResNet18 [92], MobileNetV2 [196], ShuffleNet [197], VGG11 [198], and EfficientNet [199]. Since PrivateGaze and TPGD generate different obfuscated images for each black-box gaze estimator, the reported results are averaged across these five architectures. By contrast,

the obfuscated images generated using GauDP, IP-DP, FS-DP, MaxP, and B-DAP are independent of the black-box gaze estimator used. The surrogate gaze estimator $\mathcal{G}_w(\cdot)$ is implemented using the ResNet18 architecture [92]. The classifiers used for identity and gender recognition are implemented using ResNet18. The encoder of the image generator $IG(\cdot)$ shares the same structure as $E(\cdot)$, ensuring that features extracted from the raw image x_i and the anchor image $\hat{x}$ have identical spatial dimensions. The decoder of the image generator consists of four up-convolutional blocks, each containing an upsampling of feature maps followed by three convolutional layers.

We develop PrivateGaze using the PyTorch framework and use the Adam optimizer. The standard learning rate is set to 0.001, unless specified otherwise. The privacy preserver is trained over 12,000 steps with a mini-batch size of 25 for all evaluation scenarios. The classifiers for identity and gender recognition are trained for 20 and 5 epochs, respectively. The surrogate gaze estimators, trained on the GazeCapture dataset, undergo 5 epochs of training. The black-box gaze estimators, using different structures, are trained for 25 epochs. The learning rates for training both the surrogate and black-box gaze estimators are set to 0.0001. We fix the value of λ in Equation 4.4 at 75 and conduct ablation studies to assess the impact of λ on the performance of PrivateGaze.

4.4.5. PERFORMANCE IN THE PRIVACY GOAL

The evaluation results for the effectiveness of various privacy-preserving methods are summarized in Table 4.1. For PrivateGaze, it is important to note that different anchor images are generated when the black-box estimator follows different network architectures. This variation occurs because the image generation module queries $\mathcal{G}_b(\cdot)$ to generate the anchor image $\hat{x}$. We report the averaged identity recognition accuracy and gender recognition accuracy for PrivateGaze over these different structures of $\mathcal{G}_b(\cdot)$.

From Table 4.1, it is evident that the average identity and gender recognition accuracies on images obfuscated by PrivateGaze are notably low, with values of 6.3% and 1.1% respectively, and minimal standard deviations of 0.2 and 0.1 across different structures of $\mathcal{G}_b(\cdot)$. Additionally, the average gender recognition accuracies are 62.0% for the MPIIFaceGaze dataset and 53.9% for the ColumbiaGaze dataset, both with a standard deviation of 0.0 across different structures of $\mathcal{G}_b(\cdot)$. The results underscore the effectiveness of PrivateGaze in significantly reducing the recognizability of identity and gender attributes in the obfuscated images. The attacker cannot train an effective identity recognizer or gender recognizer on the obfuscated images generated by PrivateGaze, even when they have access to correct identity and gender labels of the obfuscated images. The minimal standard deviations indicate that PrivateGaze achieves consistent privacy performance across various architectures of $\mathcal{G}_b(\cdot)$, demonstrating its robustness and generalizability.

For GauDP, IP-DP, and FS-DP, we observe that the identity and gender recognition accuracies decrease as the values of ϵ drop, indicating that these methods can effectively preserve user privacy with smaller ϵ values. For example, the identity recognition accuracy for GauDP with $\epsilon = 0.1$ is 4.85% on the ColumbiaGaze dataset, while for FS-DP is 5.86% with $\epsilon = 1.0$. However, as we will demonstrate in Section 4.4.6, the utility performance of GauDP, IP-DP, and FS-DP with small ϵ values is significantly worse compared to PrivateGaze.

For obfuscated images obtained by MaxP and B-DAP, the identity recognition ac-

Table 4.1: Identity and gender recognition accuracies (%) on obfuscated images generated by PrivateGaze and other baseline methods evaluated on MPIIFaceGaze and ColumbiaGaze datasets. We report the results of GauDP, IP-DP, and FS-DP with different ϵ values. The proposed PrivateGaze can effectively preserve the private attributes against the attackers who train their classifiers on obfuscated images annotated with correct identity and gender labels. Lower recognition accuracy indicates better performance in privacy protection. "W/o Defense" denotes the scenario where unprotected original images are used for gaze estimation services, with attribute classifiers trained and tested on the original images.

	MPIIFaceGaze		**ColumbiaGaze**	
Method	Identity	Gender	Identity	Gender
W/o Defense	99.7	99.8	99.9	99.9
GauDP (0.1)	38.2	73.2	4.85	57.5
GauDP (0.3)	84.2	87.0	95.1	87.2
GauDP (0.5)	93.1	95.7	99.6	96.9
IP-DP (0.3)	17.9	65.9	6.20	57.0
IP-DP (3.0)	94.3	96.2	86.4	96.9
IP-DP (5.0)	96.2	98.6	97.4	99.2
FS-DP (1.0)	29.6	63.0	5.86	57.6
FS-DP (2.0)	75.6	78.4	38.9	67.7
FS-DP (3.0)	90.3	80.5	73.8	82.2
MaxP	99.3	99.6	99.7	99.5
BDAP	98.9	96.3	91.1	95.6
TPGD	6.50	62.0	2.82	56.9
PrivateGaze	6.31	62.0	1.13	56.9

curacy exceeds 90%, and the gender recognition accuracy is higher than 95% on both datasets. This outcome implies that the obfuscated images generated by MaxP and BDAP still retain discernible features, which could potentially be exploited by attackers to infer the private attributes of the users. Lastly, TPGD achieves results similar to PrivateGaze on both testing datasets, as it is designed based on our framework.

In-depth analysis and discussion. As shown in Table 4.1, the gender recognizers for PrivateGaze and TPGD have the same recognition accuracy, i.e., 62.0% and 56.9% on MPIIFaceGaze and ColumbiaGaze datasets, respectively. We observe that when obfuscated images contain no discernible gender cues, the trained recognizers tend to output the same gender label for any inputs. Essentially, without gender-related information in the obfuscated images, the best knowledge the recognizers can obtain is an approximation of the gender distribution in the training dataset $\mathcal{D}_p$. Consequently, during inference, leveraging prior probability distributions, the recognizers predict the gender of a given testing image to be the one with the highest prior probability, thereby achieving a better classification accuracy than random guessing.

For instance, in MPIIFaceGaze, the gender recognizers for both PrivateGaze and TPGD consistently classify subjects in all testing images as male. This outcome results from the imbalanced gender distribution within the training dataset $\mathcal{D}_p$, where 59.5% of images feature male subjects. Consequently, given that 62.0% of the testing images contain male subjects, this results in an equal gender recognition accuracy of 62.0%. Thus, these evaluation results demonstrate the effectiveness of PrivateGaze in preserving gender information.

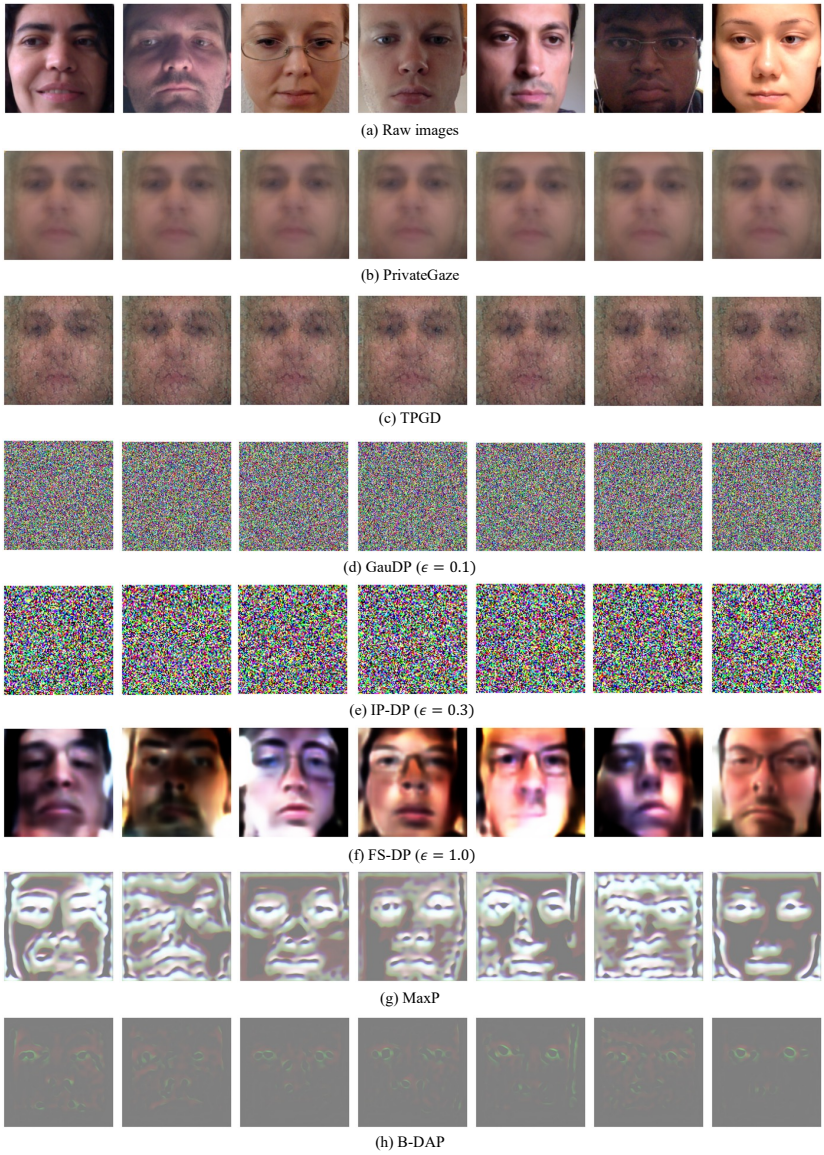


Figure 4.5: Illustration of (a) raw images of different subjects and obfuscated images generated by (b) PrivateGaze, (c) TPGD, (d) GauDP ($\epsilon = 0.1$), (e) IP-DP ($\epsilon = 0.3$), (f) FS-DP ($\epsilon = 1.0$), (g) MaxP, and (h) B-DAP. The obfuscated images obtained by PrivateGaze and TPGD have similar appearances, making it challenging for attackers to infer user identity and gender from the obfuscated images.

In contrast to the imbalanced gender distribution, the identity distribution within the training set $\mathcal{D}_p$ for both MPIIFaceGaze and ColumbiaGaze datasets is well-balanced, with each subject having an equal number of training images. Thus, when the identity recognizers cannot learn any identity-related information from the obfuscated images and are faced with equal prior probabilities, the identity recognition accuracy is similar to that of random guess, i.e., 6.31% and 1.13% for MPIIFaceGaze and ColumbiaGaze datasets, respectively. This result further demonstrates the capability of PrivateGaze in preserving user information from potential malicious service providers.

Visualization of obfuscated images. We also perform a visual comparison between the raw full-face images and the obfuscated images generated by the proposed PrivateGaze and other baseline methods. As shown in Figures 4.5 (b) and (c), the obfuscated images generated by PrivateGaze and TPGD exhibit similar visual characteristics among themselves, making it extremely challenging for an attacker to infer the user private attributes, even when correctly labeled obfuscated images are used to train the deep learning-based classifiers (as showcased in Table 4.1). In contrast, as shown in Figure 4.5 (g), the user attributes are much more discernible in the obfuscated images produced by MaxP. Moreover, although it might be difficult for human observers to identify the subjects in images obfuscated by B-DAP, the results shown in Table 4.1 indicate that an attacker can successfully train a classifier on these obfuscated images to accurately classify user private attributes.

As shown in Figures 4.5 (d) and (e), GauDP ($\epsilon = 0.1$) and IP-DP ($\epsilon = 0.3$) apply significant perturbations to the raw images, making it difficult for the malicious service provider to classify identities and genders from them. Furthermore, FS-DP ($\epsilon = 1.0$) generates obfuscated images that have significantly different appearances from raw images by perturbing features with strong random noises, effectively preserving user privacy.

Residual map. To better understand why the obfuscated images generated by PrivateGaze have similar appearances yet lead the black-box gaze estimators to output varied gaze directions, we examine the residual maps between the obfuscated image and the anchor image. As shown in Figure 4.6, the privacy preserver adds perturbations to the eye regions of the anchor image, which are the most critical parts of the facial image for gaze estimation [28]. These perturbations are learned by the privacy preserver and contain the gaze-related features of the raw images. In this way, the obfuscated images retain similar appearances while leading to different gaze directions when fed into the black-box gaze estimator.

4.4.6. PERFORMANCE IN THE UTILITY GOAL

Below, we evaluate the performance of different methods in achieving the utility goal, i.e., gaze estimation task. The results are reported in Table 4.2, which demonstrate that PrivateGaze consistently outperforms all compared methods and achieves the lowest average angular error across two datasets and with five different neural network architectures for the black-box gaze estimator. On average, PrivateGaze improves gaze estimation performance by 41.79%-78.47%, and 34.13%-64.68% on MPIIFaceGaze and ColumbiaGaze, respectively.

For differential privacy-based methods, i.e., GauDP, IP-DP, and FS-DP, while they

Table 4.2: Performance in the utility goal is measured by the average angular error (in degree) of the black-box gaze estimator across five different structures, using obfuscated images generated by different methods as inputs for gaze estimation. We present results for GauDP, IP-DP, and FS-DP with different ϵ values, respectively. PrivateGaze consistently outperforms all compared methods in all examined scenarios. Overall, PrivateGaze achieves an average performance improvement in gaze estimation of 49.86%-77.10%, and 34.13%-60.67% on MPIIFaceGaze and ColumbiaGaze, respectively.

Method	ResNet	MobileNet	ShuffleNet	VGG	EfficentNet	Average	Improvement
GauDP (0.1)	20.73	25.70	13.93	14.06	86.02	32.09	77.10%
GauDP (0.3)	20.37	25.72	13.62	14.82	44.52	23.81	69.13%
GauDP (0.5)	20.09	25.69	13.33	15.77	26.51	20.28	63.75%
IP-DP (0.3)	20.94	30.03	13.77	14.32	91.68	34.14	78.47%
IP-DP (3.0)	26.22	30.29	15.22	15.76	69.92	31.48	76.65%
IP-DP (5.0)	28.90	30.15	15.50	16.64	40.25	26.28	72.03%
FS-DP (1.0)	16.09	19.86	19.68	20.99	19.36	19.19	41.79%
FS-DP (2.0)	15.63	18.16	17.95	19.63	17.36	17.74	58.56%
FS-DP (3.0)	14.90	16.46	16.16	18.23	15.58	16.26	54.79%
MaxP	13.65	29.95	18.93	17.72	13.32	18.71	60.72%
BDAP	20.07	46.52	23.38	13.43	13.08	23.29	68.44%
TPGD	17.28	17.82	11.71	15.89	9.99	14.66	49.86%
PrivateGaze	**7.02**	**6.66**	**8.34**	**7.50**	**7.23**	**7.35**	

(a) **MPIIFaceGaze**

Method	ResNet	MobileNet	ShuffleNet	VGG	EfficentNet	Average	Improvement
GauDP (0.1)	14.46	17.14	13.58	14.60	83.52	28.66	60.67%
GauDP (0.3)	14.89	17.27	13.48	13.75	44.61	20.80	43.75%
GauDP (0.5)	15.65	17.33	13.31	13.39	25.86	17.11	34.13%
IP-DP (0.3)	19.42	20.55	12.86	18.42	86.91	31.63	64.68%
IP-DP (3.0)	26.75	21.02	12.55	15.78	63.39	27.89	59.94%
IP-DP (5.0)	30.80	21.03	12.36	15.11	36.51	23.16	51.77%
FS-DP (1.0)	19.35	19.30	19.94	24.62	21.95	21.03	46.88%
FS-DP (2.0)	18.90	18.96	20.44	23.54	21.39	20.64	45.88%
FS-DP (3.0)	18.80	18.71	20.26	22.99	20.68	20.29	44.94%
MaxP	15.42	22.69	20.96	27.24	15.72	20.40	44.75%
BDAP	15.22	41.32	24.87	18.50	12.64	22.50	49.91%
TPGD	21.45	23.17	19.35	20.81	13.67	19.69	42.76%
PrivateGaze	**9.72**	**11.45**	**11.19**	**12.19**	**11.80**	**11.17**	

(b) **ColumbiaGaze**

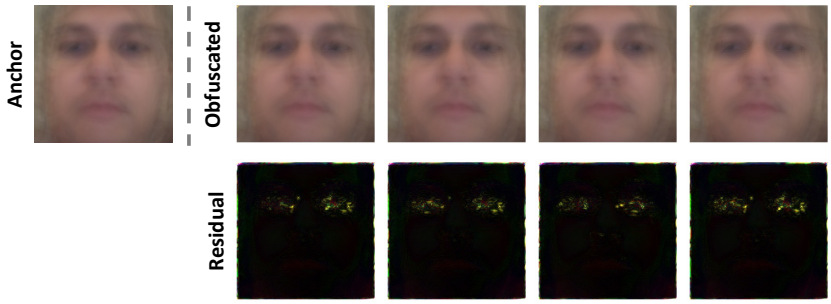


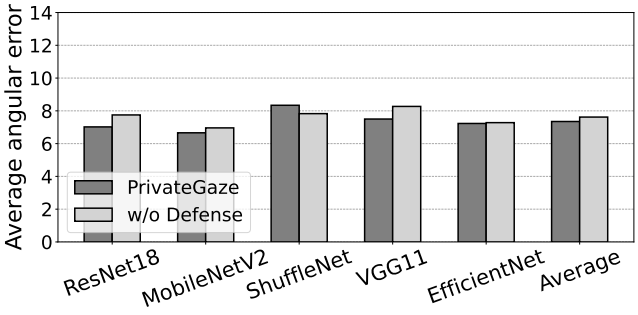
Figure 4.6: Illustration of the residual maps (scaled by a factor of 30) generated from the obfuscated images (with different gaze directions) compared to the anchor image. The privacy preserver introduces perturbations into the eye regions of the anchor image when generating obfuscated images, resulting in different gaze directions.

achieve strong privacy performance by setting ϵ to small values, their utility performance notably lags behind PrivateGaze. For instance, GauDP with $\epsilon = 0.1$ results in an average angular error over various structures of $\mathcal{G}_b(\cdot)$ of 32.09° on MPIIFaceGaze, nearly five times higher than that of PrivateGaze. Increasing the values of ϵ improves the utility performance of differential privacy-based methods by reducing the intensity of the added perturbation for privacy protection. However, as shown in Table 4.1, this improvement comes at the cost of deteriorating privacy performance.

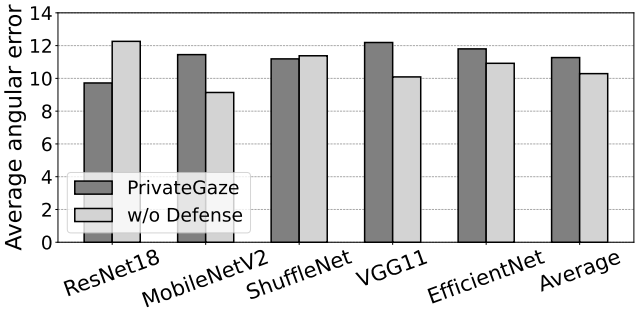
On MPIIFaceGaze dataset, the average utility performance for MaxP and B-DAP is 18.71° and 23.29° respectively. These results indicate that the autoencoder and image converter trained by MaxP and B-DAP, respectively, with the surrogate gaze estimator cannot generalize well to black-box gaze estimators. Despite TPGD achieving similar performance in privacy protection as PrivateGaze (as shown previously in Table 4.1), its average utility performance over different black-box gaze estimators is 14.66°, which is significantly worse than that for PrivateGaze, at 7.35°. Similar observations hold for the evaluation results on the ColumbiaGaze dataset.

To further investigate the performance of PrivateGaze on the utility goal, we compare the average angular error of the black-box gaze estimator when using obfuscated images generated by PrivateGaze versus raw images (*w/o Defense*) as the inputs. The results are shown in Figure 4.7. On the MPIIFaceGaze dataset, the average performance of PrivateGaze among different structures is 7.35°, which is better than that of *w/o Defense* at 7.62°. On the ColumbiaGaze dataset, PrivateGaze achieves an average utility performance of 11.27° across different structures, which is slightly higher than *w/o Defense* at 10.29°. Overall, these results indicate that PrivateGaze maintains comparable utility performance for the black-box gaze estimator even when compared to a method that does not employ any privacy protection.

Discussion on utility performance. As shown in Figure 4.7, on the MPIIFaceGaze dataset, PrivateGaze achieves superior gaze estimation performance compared to *w/o Defense* on average. To explore the reason behind this improvement, we observed a reduction in the average angular error of $\mathcal{G}_w(\cdot)$ during training on obfuscated images generated by $\mathcal{P}(\cdot)$ from the validation set. We observe a decrease from 12° to 3.52°, which is lower than the



(a) MPIIFaceGaze



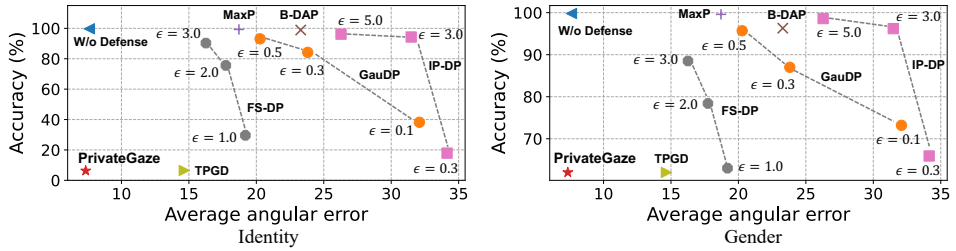
(b) ColumbiaGaze

Figure 4.7: Average angular error (in degree) of the black-box gaze estimator when using obfuscated images generated by PrivateGaze compared to raw images, i.e., w/o Defense, as inputs for gaze estimation. Our method demonstrates performance comparable to that of raw images across different structures and datasets.

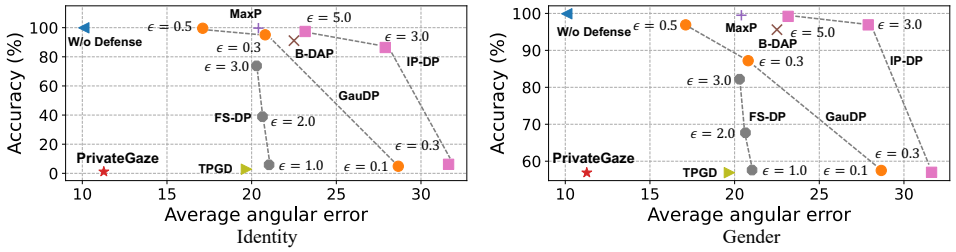
error observed on raw images, at 5.31°. This suggests that the privacy preserver acts as an *image filter* that eliminates redundant features from raw images, thereby enhancing gaze estimation performance.

Moreover, we observe that the average gaze estimation performance of PrivateGaze is better than *w/o Defense* on MPIIFaceGaze, while it shows marginally inferior performance on ColumbiaGaze. This difference can be attributed to the training data used for the privacy preserver, which was trained on the GazeCapture dataset and evaluated on both MPIIFaceGaze and ColumbiaGaze. As shown in Figure 4.4, both GazeCapture and MPIIFaceGaze are acquired under real-world conditions employing front-facing cameras in mobile devices, whereas ColumbiaGaze is collected in a more controlled laboratory environment using web cameras. The images from GazeCapture are more similar to those from MPIIFaceGaze than to those from ColumbiaGaze. Consequently, the image filter, i.e., privacy preserver, operates more effectively on MPIIFaceGaze compared to ColumbiaGaze, and leads to better gaze estimation performance on MPIIFaceGaze.

In-depth analysis on the generalization ability of PrivateGaze. PrivateGaze demonstrates good generalization ability, i.e., $\mathcal{P}(\cdot)$ is trained with $\mathcal{G}_w(\cdot)$ but yields good gaze estimation performance on $\mathcal{G}_b(\cdot)$. We attribute this capability primarily to the proposed anchor im-



(a) MPIIFaceGaze



(b) ColumbiaGaze

Figure 4.8: Overall performance comparison between PrivateGaze and the compared methods on (a) MPIIFaceGaze and (b) ColumbiaGaze datasets. The X-axis is the utility performance, i.e., the average angular error, while the Y-axis is the privacy performance, i.e., the identity recognition accuracy and gender recognition accuracy. The plots are the identity (gender) recognition accuracy and the average angular error for PrivateGaze and the compared methods. The overall performance of PrivateGaze lies on the lower left corner in all the evaluation scenarios, which indicates the superiority of PrivateGaze on the privacy-utility trade-off.

age generation module and privacy protection mechanism. Specifically, the privacy protection mechanism ensures that the obfuscated images x' closely resemble the anchor image $\hat{x}$, resulting in similar gaze estimation results for both $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$. This alignment encourages the obfuscated images to produce similar outputs for both $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$. Consequently, while PrivateGaze optimizes the x' to facilitate accurate gaze direction inference by $\mathcal{G}_w(\cdot)$, it also achieves good gaze estimation performance on $\mathcal{G}_b(\cdot)$. Our experiments validate this analysis. In contrast, TPGD utilizes the anchor image to generate obfuscated images without considering their similarity to the anchor image, leading to substantially inferior performance compared to PrivateGaze in terms of utility. Moreover, as detailed in Section 4.4.8, neglecting the outputs of $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$ during the generation of $\hat{x}$ results in a significant performance decline for PrivateGaze in terms of utility.

4.4.7. OVERALL PERFORMANCE COMPARISON

Below, we compare the overall performance, i.e., the privacy-utility trade-off, between PrivateGaze and the compared methods. The results are shown in Figure 4.8. Across all evaluation scenarios, PrivateGaze consistently occupies the lower left corner, indicating its effectiveness in preserving user privacy while maintaining good utility performance. Notably, TPGD, GauDP ($\epsilon = 0.1$), FS-DP ($\epsilon = 1.0$), and IP-DP ($\epsilon = 0.3$) demon-

Table 4.3: Impact of the proposed anchor image generation algorithm on the performance of PrivateGaze. Our method of generating anchor images is essential for maintaining the utility performance of PrivateGaze.

Task	PrivateGaze	RandomAnchor
Utility: Gaze estimation	7.23°	9.06°
Privacy: Identity recognition accuracy	6.31%	6.40%
Privacy: Gender recognition accuracy	62.0%	62.0%

strate privacy-preserving performance comparable to PrivateGaze, yet their utility performances significantly lag behind. Moreover, among the differential privacy-based methods, i.e., GauDP, IP-DP, and FS-DP, FS-DP achieves the most favorable privacy-utility trade-off. This is because FS-DP adopts state-of-the-art DP techniques and involves the gaze estimator in the training stage. Lastly, MaxP and B-DAP exhibit overall performances within the upper middle region, indicating their inability to achieve both utility and privacy goals simultaneously.

4.4.8. ABLATION STUDIES

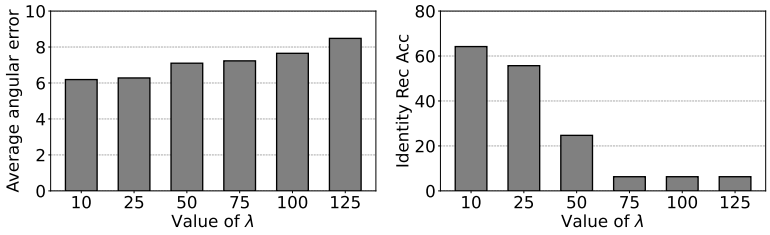
We conduct ablation studies to investigate the impact of different design choices on the system performance. We use MPIIFaceGaze as the testing set and implement a black-box gaze estimator using EfficientNet.

Impact of λ on the performance trade-off between utility and privacy. For the optimization problem described in Equation 4.4, the parameter λ trades off the utility objective and the privacy-preserving objective. To explore how changes in λ affect the performance of PrivateGaze, we experiment with varying λ within the range from 10 to 125 and present results in Figure 4.9.

Overall, reducing the value of λ improves the utility performance of PrivateGaze yet degrades its privacy performance when λ is set below 75. Particularly, there is a decrease in the average angular error from 8.48° to 6.10° when λ is reduced from 125 to 10. This indicates an increased weighting of the utility objective in the optimization problem. For privacy objective, the identity recognition accuracy is higher than 20% when the value of λ is smaller than 75, while the gender recognition accuracy is stable for all the examined values of λ. In conclusion, we have chosen to set λ to 75 in our implementation. This setting effectively balances the need to preserve user privacy in the obfuscated images while maintaining comparable utility performance to raw images.

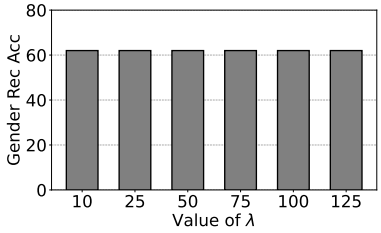
Impact of anchor image generation. We investigate the impact of the proposed anchor image generation module (described in Algorithm 1) on the performance of PrivateGaze. Specifically, instead of querying the black-box gaze estimator and the surrogate gaze estimator to find a suitable set of images, we randomly sample 50 images from the training set to form an average facial image as the anchor image. We use the term *RandomAnchor* to denote the method of generating anchor images through randomly sampled images. The results are shown in Table 4.3.

First, the anchor image does not affect the performance of PrivateGaze on the privacy objective, as the identity recognition accuracy and the gender recognition accuracy of RandomAnchor and PrivateGaze are similar. Second, the average angular error of RandomAnchor is 20% higher than that of PrivateGaze. This indicates that the pro-



(a) Utility

(b) Preserving users identities



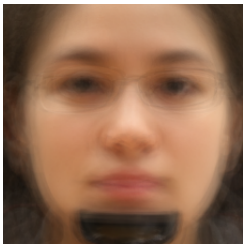
(c) Preserving users genders

Figure 4.9: Impact of λ on the performance trade-off between utility and privacy. (a) utility, (b) preserving users' identities, and (c) preserving users' genders. The value of λ trades off the performance of PrivateGaze on utility against privacy. PrivateGaze sets $\lambda = 75$, which allows PrivateGaze to consistently preserve user privacy while maintaining comparable utility performance to raw images across different datasets.

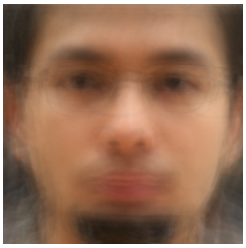
posed method for generating the anchor image can improve the generalizability of the privacy preserver. In other words, when trained with the surrogate gaze estimator, the privacy preserver can still achieve good gaze estimation performance when the obfuscated images are used by the black-box gaze estimator. Moreover, we observe that the utility performance of RandomAnchor is superior to that of the other baseline methods shown in Table 4.2. This highlights the effectiveness of our method in achieving the utility objective.

Impact of gender of the anchor image on preserving gender information. In general, the anchor image is constructed by averaging full-face images from both male and female subjects, ensuring it does not portray a specific gender. When using the anchor image as the base in generating the obfuscated images, as illustrated in Figure 4.6, the resulting obfuscated images will resemble the anchor image and also avoid portraying a specific gender. Nevertheless, there is a possibility that the images used to generate the anchor image predominantly belong to subjects of a specific gender, especially in cases where the gender distribution within the training dataset is imbalanced. In such instances, the anchor image may inadvertently exhibit characteristics of that specific gender.

To study how the gender of the anchor image affects the performance of PrivateGaze in preserving gender information, we generate an anchor image $\hat{x}_f$ using only images containing female subjects. The resulting anchor image is shown in Figure 4.10 (a). In this case, we consider the genders of $\hat{x}_f$ and the corresponding obfuscated images as "fe-



(a) Female



(b) Male

Figure 4.10: Two anchor images that are generated using images containing (a) female-only subjects and (b) male-only subjects, respectively.

male". We find that the gender classification accuracy on the obfuscated images remains at 62.0%. As discussed in Section 4.5.1, this indicates the effectiveness of PrivateGaze in preserving user's gender information. Similarly, we conduct experiments using only male images to generate the anchor image (illustrated in Figure 4.10 (b)), and we obtain the same result. These results demonstrate that the efficacy of PrivateGaze in preserving users' gender information is not compromised even when the anchor image exhibits a specific gender.

4.4.9. SYSTEM PERFORMANCE ON DIFFERENT COMPUTATION PLATFORMS

We measure the processing time and memory usage of the privacy preserver when implemented with different architectures and deployed on different computation platforms. In addition to the default structure, which consists of four convolutional blocks and up-convolutional blocks, we also evaluate a variant with such blocks. We assess the performance of PrivateGaze on three hardware platforms, including a desktop equipped with an NVIDIA GeForce RTX 3080Ti GPU, a laptop featuring an NVIDIA GeForce RTX 3060 GPU, and a laptop equipped with an NVIDIA GeForce RTX 1050Ti GPU. These platforms are chosen to represent a wide range of common computational devices used in daily scenarios.

Processing time. We measure the latency introduced by the privacy preserver in generating obfuscated images. We randomly sample one image from the MPIIFaceGaze dataset and feed it into the privacy preserver. We repeat the experiment 1000 times, and report the average processing time on different hardware platforms in Table 4.4.

Specifically, for the privacy preserver consisting of four convolutional blocks and up-convolutional blocks, the average processing time on the desktop with an NVIDIA GeForce RTX 3080Ti GPU is less than 4 ms. The processing time on the laptops with an NVIDIA GeForce RTX 3060 GPU and with an NVIDIA GeForce RTX 1050Ti GPU is 10.9 ms and 53.5 ms, respectively. When reducing the number of the convolutional and up-convolutional blocks to three, the average processing time decreases. On the desktop, it is reduced to 2.7 ms, on the laptop with an NVIDIA GeForce RTX 3060 GPU it is 9.6 ms, and the laptop with an NVIDIA GeForce RTX 1050Ti GPU it is 46.7 ms. These results indicate that the deployed privacy preserver introduces minimal processing latency.

Table 4.4: The processing time (in ms) on different hardware platforms. PrivateGaze does not introduce too much processing latency.

Platforms	3 Blocks	4 Blocks
Desktop (RTX 3080Ti)	2.7	3.8
Laptop (RTX 3060)	9.6	10.9
Laptop (RTX 1050Ti)	46.7	53.5

Table 4.5: The memory usage (in MB) on different hardware platforms. PrivateGaze consumes similar memories on different hardware platforms.

Platforms	3 Blocks	4 Blocks
Desktop (RTX 3080Ti)	2193	2267
Laptop (RTX 3060)	1964	2043
Laptop (RTX 1050Ti)	1902	2003

Memory usage. To measure memory usage, we follow the method described in [200] by reporting the memory allocated specifically to the privacy preserver. This is determined by subtracting the memory usage before loading the privacy preserver from the run-time memory usage. The results are shown in Table 4.5, indicating similar memory usage across different scenarios, approximately 2,000 MB for the privacy preserver.

Utility and privacy performance. We report the utility and privacy performance of the privacy preserver with different structures in Table 4.6. When reducing the number of convolutional and up-convolutional blocks, the utility performance of PrivateGaze is decreased to 8.12°, which is approximately 1° higher than w/o defense, while the privacy performance maintains stable. Therefore, although using three convolutional and up-convolutional blocks can slightly reduce the processing time, we opt to design the privacy preserver with four convolutional and up-convolutional blocks to ensure comparable utility performance to w/o defense.

4.4.10. DISCUSSIONS

Below, we discuss the key findings of this chapter and the impacts of PrivateGaze. We also discuss its limitations and propose future research directions for enhancing user privacy in black-box mobile services.

Key findings. This work presents three major findings. First, we demonstrate effective user privacy preservation by transforming different raw images into obfuscated images that have similar appearances with a pre-generated anchor image. Second, leveraging

Table 4.6: The utility and privacy performance of PrivateGaze in different structures.

Task	3 Blocks	4 Blocks	w/o Defense
Utility: Gaze estimation	8.12°	7.23°	7.28°
Privacy: Identity recognition accuracy	6.36%	6.31%	99.8%
Privacy: Gender recognition accuracy	62.0%	62.0%	99.4 %

the anchor image allows us to control the appearance of obfuscated images, thereby achieving our utility goal. Specifically, since the anchor image, i.e., the average full-face image, produces consistent outputs for both $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$, obfuscated images that closely resemble the anchor image also yield consistent results for gaze estimators $\mathcal{G}_w(\cdot)$ and $\mathcal{G}_b(\cdot)$. This alignment enables $\mathcal{P}(\cdot)$ trained with $\mathcal{G}_w(\cdot)$ to perform accurate gaze estimation on $\mathcal{G}_b(\cdot)$. Lastly, our well-designed $\mathcal{P}(\cdot)$ structure and training objective allow us to manipulate the behaviours of gaze estimators through imperceptible modification applied to the anchor image. This finding underscores vulnerabilities in deep learning-based gaze estimation systems.

Impacts of PrivateGaze. Compared to existing works [160], [161], [162], PrivateGaze addresses a more practical scenario where the deep learning-based model used by the service provider remains a black box to users. PrivateGaze introduces a novel framework designed to preserve user privacy while maintaining good gaze estimation performance on such black-box models. While our current evaluation focuses on preserving identity and gender as private attributes, the framework's flexibility allows for the preservation of other private attributes, such as ages, emotions, and details of the user's surroundings. Moreover, PrivateGaze can be extended to preserve user privacy in various applications, including head pose estimation [201] and emotion recognition [202], by adapting the utility goals accordingly.

Limitations. Our experiments have demonstrated that PrivateGaze outperforms DP-based methods in achieving both privacy and utility goals. However, it is important to note that unlike DP-based methods, the current design of PrivateGaze does not provide a theoretical privacy guarantee.

Future research directions. A promising avenue for future research involves extending PrivateGaze to other applications, such as hand pose estimation [201]. Moreover, while the measured processing latency suggests that PrivateGaze is promising for real-time on-device deployment, its run-time memory usage is around 2 GB in the current implementation. This may be acceptable for relatively powerful edge devices, but it can still be challenging for commodity smartphones and memory-constrained standalone AR/VR headsets. Supporting such platforms more efficiently requires future work on lightweight architecture design, memory optimization, and mobile-specific model compression. Another intriguing direction is to develop privacy-preserving solutions tailored for wearable-based gaze estimation systems that either utilize near-eye pupil images[16], [137] or event streams [203], [204], [205], [206] as tracking inputs. These systems pose unique challenges due to the sensitivity of the data captured and the wide adoption of eye tracking in head-mounted platforms such as augmented/virtual reality devices [207], [208]. Designing effective privacy-preserving solutions for such systems could significantly enhance user trust and adoption in these technologies.

4.5. CONCLUSION

In this chapter, we present PrivateGaze to address **Sub-Question 3** of the main research question to protect users' privacy information when calling black-box gaze tracking services. PrivateGaze trains a user-side privacy preserver to convert privacy-sensitive full-

face images into privacy-enhanced obfuscated versions. The obfuscated images do not contain any information about the users' private attributes yet can be directly used by the black-box gaze estimator to obtain accurate gaze directions. Our comprehensive experiments on four benchmark datasets show that PrivateGaze can effectively protect users' private attributes, e.g., identity and gender, even when the attribute recognizers are trained on obfuscated images with accurate attribute labels. Meanwhile, the obfuscated images generated by PrivateGaze can achieve comparable gaze tracking performance to conventional, unprotected full-face images.

5

CONCLUSION

The demand for diverse gaze-based applications in daily life settings and the advancement of deep learning have encouraged the development of appearance-based gaze estimation, which enables non-intrusive and cost-effective gaze estimation by using general-purpose cameras. Depending on the requirements for the development workflow and the applications, the developer can adopt one of the following paradigms to build applications upon the appearance-based gaze estimation: (1) training gaze estimation models themselves, (2) using pre-trained gaze estimation models, or (3) calling gaze estimation services. These paradigms, however, either require substantial resources for model training or raise trustworthiness concerns due to the involvement of third parties, which places limitations on the widespread adoption of gaze estimation systems. Therefore, we consider the following main research question:

How can we make gaze estimation systems both resource-efficient and trustworthy?

We addressed this main research question by considering each paradigm individually. In the following, we first summarize our contributions to each paradigm. Then, we conclude the overarching contribution of this dissertation to the main research question. We finally gaze at the future of gaze estimation systems.

5.1. CONTRIBUTIONS

This dissertation made the following contributions towards resource-efficient and trustworthy gaze estimation systems.

EfficientGaze: Improving resource-efficiency for self-trained models. Self-trained gaze estimation models offer the highest trustworthiness compared to other paradigms, as no external parties are included in developing the gaze estimation system. However, the reliance on large-scale labeled datasets and complex neural networks to train accurate gaze estimation models results in a high resource burden. In Chapter 2, we presented EfficientGaze, a resource-efficient gaze estimation framework that reduces the resource cost for using self-trained gaze estimation models. We proposed to use selected discrete

cosine transform (DCT) coefficients of RGB images as input for gaze estimation, which significantly alleviates the computational burden, enabling up to a 3.7 times reduction in FLOPs. Furthermore, we designed multi-task gaze-aware contrastive learning to learn gaze representations from unlabeled full-face images to dramatically reduce the data labeling hurdle. Our evaluation demonstrates that EfficientGaze achieves comparable gaze-estimation accuracy with the conventional approaches that require a large amount of well-labeled data, while significantly improving the efficiency in terms of computational resources.

SecureGaze: Defending pre-trained models against backdoor attacks. Deploying pre-trained models can effectively reduce the resource cost compared to self-trained models. However, this dissertation disclosed that pre-trained models are vulnerable to backdoor attacks, where an adversary model provider can inject a backdoor into the pre-trained model and manipulate the output of the backdoored gaze estimation model with a backdoor trigger. In Chapter 3, we proposed SecureGaze, the first approach to defend gaze estimation models against backdoor attacks. We identified the key characteristics of backdoored gaze estimation models, based on which we introduced a novel suite of techniques to reverse engineer the backdoor trigger. With the reverse-engineered backdoor trigger, SecureGaze determines if a pre-trained model is backdoored or not, and mitigates the backdoor behavior of the compromised model. Our comprehensive experiments in both digital and physical worlds showed that SecureGaze is consistently effective in defending gaze estimation models against various backdoor attacks that are triggered by input-aware patterns, input-independent patterns, and physical objects.

PrivateGaze: Protecting user privacy in gaze estimation services. Calling gaze estimation services offers the least resource cost among the three paradigms. However, given the rich private information contained in full-face images, gaze estimation services raise severe privacy concerns due to their black-box nature, where the users do not have knowledge of how their full-face images are processed and stored. In Chapter 4, we presented PrivateGaze, the first approach that can effectively preserve users' privacy information when calling black-box gaze estimation services. PrivateGaze trains a privacy preserver to convert privacy-sensitive full-face images into privacy-enhanced obfuscated versions that are then used to call gaze estimation services. The obfuscated images do not contain any information about the users' private attributes, yet can be directly used by the gaze estimation services to obtain accurate gaze directions. Our experiments demonstrated that PrivateGaze can effectively protect users' private attributes, e.g., identity and gender, even when the attribute recognizers are trained on obfuscated images with accurate attribute labels. Meanwhile, the obfuscated images generated by PrivateGaze can achieve comparable gaze estimation performance to conventional, unprotected full-face images.

5.2. LOOKING BACK

Resource-efficient and trustworthy gaze estimation systems are essential for enabling pervasive gaze-based applications. In this dissertation, we considered the three primary paradigms of gaze estimation systems and improved each of them in terms of resource efficiency or trustworthiness. We summarize the contributions of this dissertation in

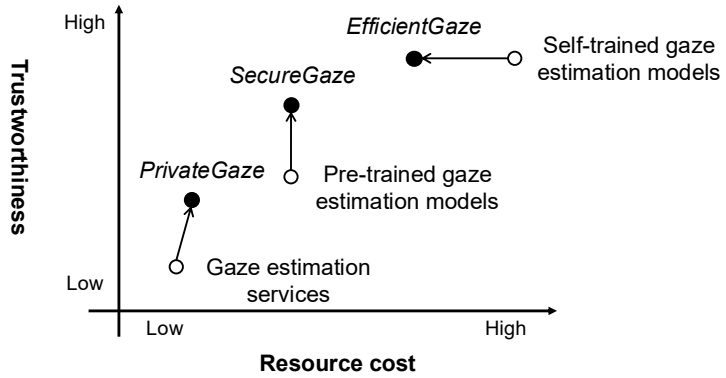


Figure 5.1: Illustration of contributions of this thesis. We improve the resource efficiency of self-trained models while enhancing the trustworthiness of pre-trained models and gaze estimation services.

Figure 5.1. Overall, we shifted the spectrum formed by resource cost and trustworthiness across different paradigms toward the upper left part, representing more desirable scenarios.

First, EfficientGaze significantly reduces the resource cost of using self-trained models by overcoming the data labeling hurdle for large-scale datasets and improving the computational efficiency of the gaze estimation model. However, EfficientGaze still requires collecting unlabeled images and labeling a handful of images in the supervised calibration stage, maintaining a higher resource cost than pre-trained models. Secondly, SecureGaze improves the trustworthiness of pre-trained gaze estimation models by detecting and mitigating the backdoor in the pre-trained models. These processes are performed only once before deployment, thereby avoiding additional resource cost for model deployment. Lastly, PrivateGaze enhances the trustworthiness of gaze estimation services by protecting user privacy, at the cost of a modest increase in computational resources required to deploy the privacy preserver.

5.3. FUTURE WORK

In this section, we outline potential directions for future work on gaze estimation. This dissertation has presented several novel techniques that advance gaze estimation systems toward greater efficiency and trustworthiness. Nonetheless, further improvements and innovations are needed to enable more pervasive and widely deployable gaze estimation systems.

Generalization to vision transformers. In Chapter 2, we present the frequency-domain gaze estimation to reduce the computational burden of gaze estimation models by processing images in the frequency domain and using specifically designed structures of neural networks. However, our analysis and experiment are limited to convolutional neural networks. Given the increasing adoption of vision transformers [209] in various computer vision tasks, including gaze estimation [210], [211], it is worthwhile to explore the possibility of adapting the proposed method to vision transformer-based solutions.

Large-scale unsupervised pre-training. Achieving accurate gaze estimation on unseen subjects remains a long-standing challenge due to the diversity in facial appearances. One promising solution is to utilize a large-scale dataset that encompasses extensive variations in facial appearance to pre-train a gaze estimation model. The proposed unsupervised gaze representation learning framework (Chapter 2) eliminates the need for gaze annotations during pre-training, which makes large-scale pre-training feasible. However, we validated the effectiveness of the proposed unsupervised learning method only on existing gaze estimation datasets, without exploring its ability on learning gaze representations from broader facial image datasets that are not specifically curated for gaze estimation, such as face recognition datasets [212]. Therefore, a promising future research direction is to construct a large-scale dataset by combining various facial image datasets and to apply the proposed method to enable large-scale unsupervised pre-training.

Advanced backdoor defense mechanisms. The topic of defending against backdoor attacks also warrants further investigation. While the proposed SecureGaze in Chapter 3 is effective against a range of backdoor attacks, it may fail when the adversary employs a large-sized trigger, such as a mask. In such cases, the reverse-engineered trigger could have a norm exceeding the predefined threshold, leading to unsuccessful detection. Therefore, advanced defense mechanisms are required to address this limitation. Moreover, another interesting research direction is to explore defense mechanisms under more complex threat scenarios, e.g., the gaze estimation models are backdoored by multiple triggers associated with different target gaze directions.

Privacy protection in near-eye gaze estimation system. While this dissertation considers protecting user privacy for full-face gaze estimation (Chapter 4), near-eye gaze estimation systems are widely adopted in head-mounted platforms such as augmented or virtual reality devices [207], [208]. These systems either utilize near-eye pupil images [16], [137] or event streams from event cameras [205], [206] as tracking inputs, posing unique privacy challenges due to the sensitivity of the captured data. Therefore, an intriguing direction for future work is to develop privacy-preserving solutions tailored for near-eye gaze estimation systems, thereby strengthening user trust and facilitating broader adoption of these technologies.

ACKNOWLEDGEMENTS

Still, writing these acknowledgements has, in its own way, become another deadline-driven moment, something that feels all too familiar from my PhD journey. I had expected to write this part in a more relaxed state of mind, carefully looking back on the past four years, on either a sunny day or a rainy one, and giving these memories the time they deserve. However, once again, things did not quite go as planned, just as they often did throughout my PhD. It may already be clear that this will not be a long acknowledgement, but it carries my sincere gratitude to everyone who has accompanied and supported me along the way.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Dr. Guohao Lan. I received my PhD offer from him on April 7, 2021, exactly five years before my defense. I actually hesitated before reaching out to him, as my background did not seem to be a good match for the requirements of the PhD advertisement. Yet he responded to my email very quickly, offering encouragement and advice on the PhD application. Throughout my PhD journey, he provided me with detailed guidance and timely feedback. I am particularly grateful that he stood up for me when my work was treated unfairly, arguing with editors on my behalf. A memorable detail of our collaboration is that he would treat me when a paper was rejected. In that sense, perhaps rejection was not always entirely a bad thing. I feel honored to be the first PhD student under his supervision to reach the defense stage, and I look forward to sharing that special moment with him. For all of this and much more, I am deeply grateful.

I am also deeply grateful to my promotor, Prof. dr. Koen Langendoen, a reliable, wise, and humorous mentor. Although I was quite nervous during my first few meetings with him and usually had to prepare my slides four or five days in advance, I soon began to enjoy our meetings. He is highly encouraging and has a way of making me feel at ease, even though the discussions themselves are always intensive. One thing I have especially appreciated about him is that he always follows through on what he promises. Whenever he says that he will send me feedback on my dissertation, paper, or proposal the next day, he always does. Koen has also been incredibly supportive and patient. Looking back on these past years, I realize that he never turned down any of my requests and always took my questions seriously. I still remember my first presentation in the group, when you took a photo of me using a long stick as a laser pointer. If that photo is still around, I would be very happy to see it again.

I am also deeply grateful to my collaborator, Dr. Xucong Zhang, who is both brilliant and warm-hearted. Xucong made valuable contributions to three of my projects and provided invaluable guidance throughout. Beyond academic discussions, I also greatly enjoyed talking with him about career planning and broader topics in life. Xucong is a direct and professional collaborator: he does not hesitate to point out issues in my work and make the possible consequences clear, yet he always does so in a calm and constructive manner while staying focused on moving the project forward. I truly value

this way of collaboration and have learned a great deal from it. To me, his role in my PhD went far beyond that of a collaborator. I would also like to express my sincere gratitude to my collaborator, Dr. Jinyuan Jia. His insightful comments and valuable guidance on AI security greatly shaped the "trustworthy" part of my dissertation. I would also like to thank Mr. Yupei Liu for his valuable suggestions during our discussions and for his contributions to the SecureGaze project.

I would also like to express my sincere gratitude to my chairperson and doctoral committee members: Prof. dr. ir. B. van Arem, Prof. dr. ir. R. L. Lagendijk, Prof. dr. H. Wen, Prof. dr. H. Gellersen, Dr. Ö. Durmaz-Incel, and Prof. dr. ir. F. A. Kuipers. I am grateful to the chairperson for introducing the defense procedure and sharing important practical advice with me. I also sincerely thank the committee members for reading my dissertation, providing valuable comments, and taking the time to attend my defense.

I would also like to express my sincere gratitude to the faculty members of our group. I am especially grateful to Dr. Qing Wang for his help throughout my PhD. I also thank Dr. Przemysław Pawełczak for taking the time to talk with me and comfort me when I felt discouraged. I am grateful to Dr. Marco Zúñiga for frequently checking in on my PhD progress. I would also like to thank my office neighbor, Dr. Arash Asadi, for his humor and for the many drawings he gave me. I am also grateful to our group's Management Assistant, Pam, for handling so many practical and administrative matters. I am very grateful to my colleagues Aaron, Adrian, Anup, Florian, Fabian, Lucan, Michail, Marco, Rangesh, Seham, and Vivian for helping me complete the user study for my final project, one of the most stressful weeks of my PhD.

I am also grateful for the company of Bo, Barry, Chenxing, Changheng, Eric, Hanting, Hao, Harry, Mingkun, Mengyuan, Miguel, Ran, Shun, Shenxiu, and Talia during these years. Mingkun and Ran, I am grateful for the many conversations we shared about research and future career paths. Thank you as well for forgiving me when I misread the train schedule while we were traveling in Italy, and we ended up stranded in La Spezia until 2 a.m. Hanting, my best food-hunting companion, thank you for taking a six-hour train ride with me at six in the morning to Düsseldorf for barbecue, and then another late-night train back home. I am also grateful that you forgave me for making you walk so much with me. Barry, my cycling companion, I guess we have ridden around 2,000 kilometers together. I am very grateful for all the rides we shared, and I also had the pleasure of witnessing your bicycle evolve from a 400-euro one into a 4,000-euro one. Bo, Chenxing, and Shun, thank you for always inviting me to meals. I only wish my cooking had been good enough to return the favor and let you try my own dishes as well.

I would also like to thank my friends outside our group: Hailong, Hongming, Jingxin, Xiaozhe, Xiangwei, Xinhan, Xue, and Yun. Hailong, as my classmate during both my bachelor's and master's studies, I never expected that we would eventually end up in the Netherlands together again, this time pursuing our PhDs. Xue and Yun, thank you for being such good friends to talk and drink with. Whenever we got together for drinks, our conversations would often last until nearly two in the morning. Xinhan, we have been planning a cycling trip to Germany for three years. It is a pity that we did not manage to make it happen before my graduation, but I hope we will still do it before yours. Jingxin, we became friends during the last months of my time at HIT, and I am very glad that we have continued this friendship during our years in Europe.

Thank you, Yuting, my girlfriend and my best friend, for accompanying me throughout my PhD journey. After more than two years of long-distance between China and the Netherlands, we began a new chapter of long-distance between Belgium and the Netherlands. Together, we have shared many wonderful memories together and left our footprints across Asia, Europe, and Oceania, and I hope we will continue exploring many more places together in the future. On that note, I would also like to thank my bicycle for carrying me more than 5,000 kilometers across the Netherlands. Do not worry, my old friend; I will not sell you after graduation.

Finally, I would like to express my deepest gratitude to my parents. None of this would have been possible without your unconditional support, both financially and emotionally. In the end, this acknowledgement grew into more than two pages and kept me writing late into the night, once again not entirely as I had originally planned.

Lingyu Du
China, March 2026

BIBLIOGRAPHY

[1] T. Toyama, T. Kieninger, F. Shafait, and A. Dengel, "Museum guide 2.0-an eye-tracking based personal assistant for museums and exhibits," in *Proceedings of International Conference on Re-Thinking Technology in Museums*, vol. 1, 2011, pp. 1–24.

[2] Z. Wang, Y. Shi, Y. Wang, Y. Yao, K. Yan, Y. Wang, L. Ji, X. Xu, and C. Yu, "G-voila: Gaze-facilitated information querying in daily scenarios," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 2, pp. 1–33, 2024.

[3] O. Namnakani, P. Sinrattanavong, Y. Abdrabou, A. Bulling, F. Alt, and M. Khamis, "Gazecast: Using mobile devices to allow gaze-based interaction on public displays," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2023, pp. 1–8.

[4] K. Pfeuffer, J. Alexander, M. K. Chong, Y. Zhang, and H. Gellersen, "Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 2015, pp. 373–383.

[5] P. Valdes-Dapena, *In a new BMW sedan, drivers can change lanes using just their eyes*, `https://edition.cnn.com/2023/05/24/business/bmw-eye-lane-change/index.html`, 2023.

[6] L. Kelion, *Caterpillar backs eye-tracker to combat driver fatigue*, `https://www.bbc.com/news/technology-22640279`, 2013.

[7] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478–500, 2009.

[8] Q. Guillon, N. Hadjikhani, S. Baduel, and B. Rogé, "Visual social attention in autism spectrum disorder: Insights from eye tracking studies," *Neuroscience & Biobehavioral Reviews*, vol. 42, pp. 279–297, 2014.

[9] E. Shishido, S. Ogawa, S. Miyata, M. Yamamoto, T. Inada, and N. Ozaki, "Application of eye trackers for understanding mental disorders: Cases for schizophrenia and autism spectrum disorder," *Neuropsychopharmacology reports*, vol. 39, no. 2, pp. 72–77, 2019.

[10] B. Pfleging, D. K. Fekety, A. Schmidt, and A. L. Kun, "A model relating pupil diameter to mental workload and lighting conditions," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2016, pp. 5776–5788.

[11] Y. Yamada and M. Kobayashi, "Detecting mental fatigue from eye-tracking data gathered while watching video: Evaluation in younger and older adults," *Artificial Intelligence in Medicine*, vol. 91, pp. 39–48, 2018.

[12] N. Srivastava, J. Newn, and E. Velloso, "Combining low and mid-level gaze features for desktop activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–27, 2018.

[13] G. Lan, B. Heit, T. Scargill, and M. Gorlatova, "GazeGraph: Graph-based few-shot cognitive context sensing from human visual behavior," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2020, pp. 422–435.

[14] Y. Cheng, H. Wang, Y. Bao, and F. Lu, "Appearance-based gaze estimation with deep learning: A review and benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 7509–7528, 2024.

[15] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1124–1133, 2006.

[16] M. Kassner, W. Patera, and A. Bulling, "Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct publication*, 2014, pp. 1151–1160.

[17] *Pupil labs core eye tracker*, `https://pupil-labs.com/products/core`.

[18] *Tobii Pro Glasses 3*, `https://www.tobii.com/products/eye-trackers/wearables/tobii-pro-glasses-3`.

[19] *Pupil labs neon eye tracker*, `https://pupil-labs.com/products/neon`.

[20] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 2013, pp. 271–280.

[21] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2176–2184.

[22] E. Wood and A. Bulling, "EyeTab: Model-based gaze estimation on unmodified tablet computers," in *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, 2014, pp. 207–210.

[23] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 162–175, 2019.

[24] Q. He, X. Hong, X. Chai, J. Holappa, G. Zhao, X. Chen, and M. Pietikäinen, "Omeg: Oulu multi-pose eye gaze dataset," in *Proceedings of Scandinavian Conference on Image Analysis*, 2015, pp. 418–427.

[25] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4511–4520.

[26] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 12, pp. 2246–2260, 2007.

[27] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 365–381.

[28] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 51–60.

[29] R. Kothari, S. De Mello, U. Iqbal, W. Byeon, S. Park, and J. Kautz, "Weakly-supervised physically unconstrained gaze estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9980–9989.

[30] Y. Wang, Y. Jiang, J. Li, B. Ni, W. Dai, C. Li, H. Xiong, and T. Li, "Contrastive regression for domain adaptation on gaze estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 376–19 385.

[31] X. Zhang, M. X. Huang, Y. Sugano, and A. Bulling, "Training person-specific gaze estimators from user interactions with multiple devices," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–12.

[32] X. Zhang, Y. Sugano, and A. Bulling, "Everyday eye contact detection using unsupervised gaze target discovery," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 193–203.

[33] SeeSo, *Seeso*, `https://seeso.io/`, accessed: 2023-11-09, 2021.

[34] RealEye, *Realeye*, `https://www.realeye.io/`, accessed: 2023-11-09, 2017.

[35] E. S. Dalmaijer, S. Mathôt, and S. Van der Stigchel, "Pygaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments," *Behavior Research Methods*, vol. 46, pp. 913–921, 2014.

[36] EyeWare, *Eyeware*, `https://eyeware.tech/`, accessed: 2023-11-09, 2016.

[37] VicarVison, *Vicarvison*, `https://vicarvision.nl/blog/eyereader-webcam-based-eye-tracking-technology/`, accessed: 2023-11-09, 2007.

[38] GazeRecorder, *Gazerecorder*, `https://gazerecorder.com/`, accessed: 2023-11-09, 2009.

[39] L. V. Doore, *Wegaze*, `https://lauravandoore.com/portfolio-item/wegaze/`, accessed: 2023-11-09, 2020.

[40] M. L. Mele and S. Federici, "Gaze and eye-tracking solutions for psychological research," *Cognitive Processing*, vol. 13, no. 1, pp. 261–265, 2012.

[41] M. Khamis, F. Alt, and A. Bulling, "The past, present, and future of gaze-enabled handheld mobile devices: Survey and lessons learned," in *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2018, pp. 1–17.

[42] Y. Lei, S. He, M. Khamis, and J. Ye, "An end-to-end review of gaze estimation and its interactive applications on handheld mobile devices," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–38, 2023.

[43] C. Katsini, Y. Abdrabou, G. E. Raptis, M. Khamis, and F. Alt, "The role of eye gaze in security and privacy applications: Survey and future HCI research directions," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–21.

[44] B. Berman, *Driver-monitoring systems to be as common as seat belts*, `https://www.sae.org/news/2020/02/smart-eye-safety-driver-monitoring`, 2020.

[45] N. T. N. D. Staff, *Eye tracker wakes sleepy drivers*, `https://www.nbcnews.com/id/wbna39668980`, 2010.

[46] L. Fridman, B. Reimer, B. Mehler, and W. T. Freeman, "Cognitive load estimation in the wild," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–9.

[47] A. Palazzi, D. Abati, F. Solera, R. Cucchiara, et al., "Predicting the driver's focus of attention: The DR(eye)VE project," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1720–1733, 2018.

[48] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, 2015.

[49] *The secretive company that might end privacy as we know it*. `https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html`.

[50] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, "Fawkes: Protecting privacy against unauthorized deep learning models," in *Proceedings of the USENIX Security Symposium*, 2020, pp. 1589–1604.

[51] K. R. Rao and P. Yip, *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[52] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[53] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of International Conference on Machine Learning*, 2020, pp. 1597–1607.

[54] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo, "DetCo: Unsupervised contrastive learning for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8392–8401.

[55] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 5–22, 2022.

[56] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6912–6921.

[57] A. Newman, B. McNamara, C. Fosco, Y. B. Zhang, P. Sukhum, M. Tancik, N. W. Kim, and Z. Bylinskii, "Turkeyes: A web-based toolbox for crowdsourcing attention data," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[58] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards foveated rendering for gaze-tracked virtual reality," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–12, 2016.

[59] R. Albert, A. Patney, D. Luebke, and J. Kim, "Latency requirements for foveated rendering in virtual reality," *ACM Transactions on Applied Perception*, vol. 14, no. 4, pp. 1–13, 2017.

[60] S. Huynh, R. K. Balan, and J. Ko, "iMon: Appearance-based gaze tracking system on mobile devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–26, 2021.

[61] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[62] X. Wang, K. M. Kitani, and M. Hebert, "Contextual visual similarity," *arXiv preprint arXiv:1612.02534*, 2016.

[63] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1735–1742.

[64] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6210–6219.

[65] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[66] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[67] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., "Bootstrap your own latent-a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, 2020, pp. 21 271–21 284.

[68] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proceedings of International Conference on Learning Representations*, 2018, pp. 1–16.

[69] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 69–84.

[70] Y. Yu and J.-M. Odobez, "Unsupervised representation learning for gaze estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7314–7324.

[71] Y. Yu, G. Liu, and J.-M. Odobez, "Improving few-shot user-specific gaze adaptation via gaze redirection synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 937–11 946.

[72] Y. Sun, J. Zeng, S. Shan, and X. Chen, "Cross-encoder for unsupervised gaze representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3702–3711.

[73] K. Delac, M. Grgic, and S. Grgic, "Face recognition in JPEG and JPEG2000 compressed domain," *Image and Vision Computing*, vol. 27, no. 8, pp. 1108–1120, 2009.

[74] M. Ehrlich and L. S. Davis, "Deep residual learning in the JPEG transform domain," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3484–3493.

[75] A. Ghosh and R. Chellappa, "Deep feature extraction in the DCT domain," in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2016, pp. 3536–3541.

[76] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1740–1749.

[77] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," in *Advances in Neural Information Processing Systems*, 2018, pp. 1–12.

[78] S. Ghosh, M. Hayat, A. Dhall, and J. Knibbe, "MTGLS: Multi-task gaze estimation with limited supervision," in *Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 3223–3234.

[79] Y. Hao, R. Zheng, and B. Wang, "Invariant feature learning for sensor-based human activity recognition," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 4013–4024, 2022.

[80] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.

[81] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, pp. 1–16, 2018.

[82] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.

[83] P. C. Neto, F. Boutros, J. R. Pinto, N. Damer, A. F. Sequeira, and J. S. Cardoso, "Focusface: Multi-task contrastive learning for masked face recognition," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2021, pp. 1–8.

[84] J. Li, G. Zhao, Y. Tao, P. Zhai, H. Chen, H. He, and T. Cai, "Multi-task contrastive learning for automatic ct and x-ray diagnosis of covid-19," *Pattern Recognition*, vol. 114, p. 107 848, 2021.

[85] L. Lin, S. Song, W. Yang, and J. Liu, "Ms2l: Multi-task self-supervised learning for skeleton based action recognition," in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 2490–2498.

[86] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. Abdelzaher, "FastDeepIoT: Towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 278–291.

[87] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5353–5360.

[88] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[89] W.-H. Chen and W. Pratt, "Scene adaptive coder," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 225–232, 1984.

[90] F. Zhang and D. R. Bull, *Intelligent image and video compression: communicating pictures*. Academic Press, 2021.

[91] N. Ahmed, B. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[92] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[93] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[94] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[95] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling, "Rendering of eyes for eye-shape registration and gaze estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3756–3764.

[96] Y. Bao, Y. Liu, H. Wang, and F. Lu, "Generalizing gaze estimation with rotation consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4207–4216.

[97] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[98] L. Boccardo, "Viewing distance of smartphones in presbyopic and non-presbyopic age," *Journal of Optometry*, vol. 14, no. 2, pp. 120–126, 2021.

[99] J. Long, R. Cheung, S. Duong, R. Paynter, and L. Asper, "Viewing distance and eyestrain symptoms with prolonged viewing of smartphones," *Clinical and Experimental Optometry*, vol. 100, no. 2, pp. 133–137, 2017.

[100] D. Rempel, K. Willms, J. Anshel, W. Jaschinski, and J. Sheedy, "The effects of visual display distance on eye accommodation, head posture, and vision and neck symptoms," *Human Factors*, vol. 49, no. 5, pp. 830–838, 2007.

[101] C. Johnson, "HCI and requirements engineering-people, places and interfaces," *SIGCHI Bulletin*, vol. 29, no. 1, 1997.

[102] J. Greeson, "International standards organization ergonomic standards for displays," in *Seminar Lecture Notes-Society for Information Display*, SID Society for Information Display, vol. 2, 1996, F–6.

[103] L. Du and G. Lan, "Freegaze: Resource-efficient gaze estimation via frequency-domain contrastive learning," in *Proceedings of the International Conference on Embedded Wireless Systems and Networks*, 2023, pp. 60–71.

[104] I. Bello, W. Fedus, X. Du, E. D. Cubuk, A. Srinivas, T.-Y. Lin, J. Shlens, and B. Zoph, "Revisiting ResNets: Improved training and scaling strategies," in *Advances in Neural Information Processing Systems*, 2021, pp. 22 614–22 627.

[105] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Mądry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1563–1580, 2022.

[106] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[107] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proceedings of Network and Distributed System Security Symposium*, 2018, pp. 1–15.

[108] S. Jaiswal, S. Virmani, V. Sethi, K. De, and P. P. Roy, "An intelligent recommendation system using gaze and emotion detection," *Multimedia Tools and Applications*, vol. 78, pp. 14 231–14 250, 2019.

[109] Y. Li, P. Xu, D. Lagun, and V. Navalpakkam, "Towards measuring and inferring user interest from gaze," in *Proceedings of the ACM International Conference on World Wide Web Companion*, 2017, pp. 525–533.

[110] A. Bulling and M. Wedel, "Pervasive eye-tracking for real-world consumer behavior analysis," in *A handbook of process tracing methods*, Routledge, 2019, pp. 27–44.

[111] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2019, pp. 707–723.

[112] Z. Wang, K. Mei, H. Ding, J. Zhai, and S. Ma, "Rethinking the reverse-engineering of trojan triggers," in *Proceedings of Conference on Advances in Neural Information Processing Systems*, 2022, pp. 9738–9753.

[113] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: Scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1265–1282.

[114] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proceedings of the Annual Computer Security Applications Conference*, 2019, pp. 113–125.

[115] D. Wu and Y. Wang, "Adversarial neuron pruning purifies backdoored deep models," in *Advances in neural information processing systems*, 2021, pp. 16 913–16 925.

[116] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*, 2018, pp. 273–294.

[117] X. Xu, K. Huang, Y. Li, Z. Qin, and K. Ren, "Towards reliable and efficient backdoor trigger inversion via decoupling benign features," in *Proceedings of the International Conference on Learning Representations*, 2024.

[118] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6206–6215.

[119] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[120] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[121] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2041–2055.

[122] S. Koffas, S. Picek, and M. Conti, "Dynamic backdoors with global average pooling," in *Proceedings of IEEE International Conference on Artificial Intelligence Circuits and Systems*, 2022, pp. 320–323.

[123] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 463–16 472.

[124] A. Nguyen and A. Tran, "Wanet–imperceptible warping-based backdoor attack," in *Proceedings of International Conference on Learning Representations*, 2021, pp. 1–16.

[125] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *Proceedings of Conference on Advances in Neural Information Processing Systems*, 2020, pp. 3454–3464.

[126] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *Proceedings of IEEE European Symposium on Security and Privacy*, 2022, pp. 703–718.

[127] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *Proceedings of the Annual Computer Security Applications Conference*, 2020, pp. 897–912.

[128] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The "beatrix" resurrections: Robust backdoor detection via gram matrices," in *Proceedings of Network and Distributed System Security Symposium*, 2023, pp. 1–18.

[129] Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang, and J. Ma, "Backdoor defense with machine unlearning," in *Proceedings of the IEEE International Conference on Computer Communications*, 2022, pp. 280–289.

[130] Z. Xiang, D. J. Miller, and G. Kesidis, "Detection of backdoors in trained classifiers without access to the training set," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1177–1191, 2022.

[131] Y. Zeng, S. Chen, W. Park, Z. M. Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," in *Proceedings of the International Conference on Learning Representations*, 2022, pp. 1–28.

[132] R. Zheng, R. Tang, J. Li, and L. Liu, "Data-free backdoor removal based on channel lipschitzness," in *Proceedings of European Conference on Computer Vision*, 2022, pp. 174–191.

[133] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," *arXiv preprint arXiv:1811.03728*, 2018.

[134] W. Chen, B. Wu, and H. Wang, "Effective backdoor defense by exploiting sensitivity of poisoned samples," in *Advances in Neural Information Processing Systems*, 2022, pp. 9727–9737.

[135] J. Guan, Z. Tu, R. He, and D. Tao, "Few-shot backdoor defense using shapley estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 358–13 367.

[136] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," in *Proceedings of the Conference on Advances in neural information processing systems*, 2019, pp. 1–10.

[137] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "NVGaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[138] M. Tonsen, J. Steil, Y. Sugano, and A. Bulling, "InvisibleEye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–21, 2017.

[139] Y. Sugano, X. Zhang, and A. Bulling, "AggreGaze: Collective estimation of audience attention on public displays," in *Proceedings of the ACM Annual Symposium on User Interface Software and Technology*, 2016, pp. 821–831.

[140] X. Zhang, Y. Sugano, and A. Bulling, "Evaluation of appearance-based methods and implications for gaze-based applications," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.

[141] J. L. Kröger, O. H.-M. Lutz, and F. Müller, "What does your gaze reveal about you? on the privacy implications of eye tracking," in *IFIP International Summer School on Privacy and Identity Management*, 2020, pp. 226–241.

[142] Y. Zheng, S. Park, X. Zhang, S. D. Mello, and O. Hilliges, "Self-learning transformations for improving gaze and head redirection," in *Proceedings of the Neural Information Processing Systems*, 2020, pp. 13 127–13 138.

[143] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.

[144] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *Proceedings of the International Conference on Learning Representations*, 2021, pp. 1–19.

[145] Y. Li, X. Lyu, X. Ma, N. Koren, L. Lyu, B. Li, and Y.-G. Jiang, "Reconstructive neuron pruning for backdoor defense," in *Proceedings of theInternational Conference on Machine Learning*, 2023, pp. 19 837–19 854.

[146] W. Feng, N. Xu, T. Zhang, and Y. Zhang, "Dynamic generative targeted attacks with pattern injection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 404–16 414.

[147] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[148] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.

[149] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4664–4670.

[150] S. Hueber, C. Cherek, P. Wacker, J. Borchers, and S. Voelker, "Headbang: Using head gestures to trigger discrete actions on mobile devices," in *Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2020, pp. 1–10.

[151] A. Crossan, M. McGill, S. Brewster, and R. Murray-Smith, "Head tilting for interaction in mobile contexts," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2009, pp. 1–10.

[152] Y. Wang, J. Ding, I. Chatterjee, F. Salemi Parizi, Y. Zhuang, Y. Yan, S. Patel, and Y. Shi, "Faceori: Tracking head position and orientation using ultrasonic ranging on earphones," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–12.

[153] C. Liang, C. Yu, X. Wei, X. Xu, Y. Hu, Y. Wang, and Y. Shi, "Auth+ track: Enabling authentication free interaction on smartphone by continuous user tracking," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–16.

[154] D. J. Liebling and S. Preibusch, "Privacy considerations for a pervasive eye tracking world," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 1169–1177.

[155] C. Gressel, R. Overdorf, I. Hagenstedt, M. Karaboga, H. Lurtz, M. Raschke, and A. Bulling, "Privacy-aware eye tracking: Challenges and future directions," *IEEE Pervasive Computing*, vol. 22, no. 1, pp. 95–102, 2023.

[156] B. John, A. Liu, L. Xia, S. Koppal, and E. Jain, "Let it snow: Adding pixel noise to protect the user's identity," in *Proceedings of ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1–3.

[157] A. M. Eskildsen and D. Witzner Hansen, "Analysis of iris obfuscation: Generalising eye information processes for privacy studies in eye tracking," in *Proceedings of ACM Symposium on Eye Tracking Research and Applications*, 2021, pp. 1–10.

[158] B. John, S. Jörg, S. Koppal, and E. Jain, "The security-utility trade-off for iris authentication and eye animation for social virtual avatars," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, pp. 1880–1890, 2020.

[159] B. David-John, K. Butler, and E. Jain, "For your eyes only: Privacy-preserving eye-tracking datasets," in *Proceedings of ACM Symposium on Eye Tracking Research and Applications*, 2022, pp. 1–6.

[160] S. Liu, J. Du, A. Shrivastava, and L. Zhong, "Privacy adversarial network: Representation learning for mobile data privacy," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–18, 2019.

[161] F. Pittaluga, S. Koppal, and A. Chakrabarti, "Learning privacy preserving encodings through adversarial training," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 791–799.

[162] H. Wu, X. Tian, Y. Gong, X. Su, M. Li, and F. Xu, "DAPter: Preventing user data abuse in deep learning inference services," in *Proceedings of the Web Conference*, 2021, pp. 1017–1028.

[163] T. Xiao, Y.-H. Tsai, K. Sohn, M. Chandraker, and M.-H. Yang, "Adversarial learning of privacy-preserving and task-oriented representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 12 434–12 441.

[164] S. J. Oh, M. Fritz, and B. Schiele, "Adversarial image perturbation for privacy protection a game theory perspective," in *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 1491–1500.

[165] I. Perov, D. Gao, N. Chervoniy, K. Liu, S. Marangonda, C. Umé, M. Dpfks, C. S. Facenheim, L. RP, J. Jiang, et al., "Deepfacelab: Integrated, flexible and extensible face-swapping framework," *arXiv preprint arXiv:2005.05535*, 2020.

[166] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7184–7193.

[167] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4834–4844.

[168] E. Wilson, F. Shic, and E. Jain, "Introducing explicit gaze constraints to face swapping," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2023, pp. 1–7.

[169] Y. Nirkin, I. Masi, A. Tran Tuan, T. Hassner, and G. Medioni, "On face segmentation, face swapping, and face perception," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2018, pp. 98–105.

[170] K. Cui, R. Wu, F. Zhan, and S. Lu, "Face transformer: Towards high fidelity and accurate face swapping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023, pp. 668–677.

[171] R. Chen, X. Chen, B. Ni, and Y. Ge, "Simswap: An efficient framework for high fidelity face swapping," in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 2003–2011.

[172] J. Naruniec, L. Helminger, C. Schroers, and R. M. Weber, "High-resolution neural face swapping for visual effects," in *Computer Graphics Forum*, Wiley Online Library, vol. 39, 2020, pp. 173–184.

[173] J. Kim, J. Lee, and B.-T. Zhang, "Smooth-swap: A simple enhancement for face-swapping with smoothness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 779–10 788.

[174] B. Meden, Z. Emersic, V. Struc, and P. Peer, "K-same-net: Neural network based face deidentification," in *Proceedings of the 2017 International Conference and Workshop on Bioinspired Intelligence*, 2017, pp. 1–7.

[175] Y. Jeong, J. Choi, S. Kim, Y. Ro, T.-H. Oh, D. Kim, H. Ha, and S. Yoon, "Ficgan: Facial identity controllable gan for de-identification," *arXiv preprint arXiv:2110.00740*, 2021.

[176] Z. Kuang, H. Liu, J. Yu, A. Tian, L. Wang, J. Fan, and N. Babaguchi, "Effective de-identification generative adversarial network for face anonymization," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 3182–3191.

[177] H. Xue, B. Liu, X. Yuan, M. Ding, and T. Zhu, "Face image de-identification by feature space adversarial perturbation," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 5, e7554, 2023.

[178] M. Elfares, Z. Hu, P. Reisert, A. Bulling, and R. Küsters, "Federated learning for appearance-based gaze estimation in the wild," in *Proceedings of The 1st Gaze Meets ML workshop*, PMLR, 2023, pp. 20–36.

[179] J. Steil, M. Koelle, W. Heuten, S. Boll, and A. Bulling, "Privaceye: Privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features," in *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, 2019, pp. 1–10.

[180] E. Bozkir, A. B. Ünal, M. Akgün, E. Kasneci, and N. Pfeifer, "Privacy preserving gaze estimation using synthetic images via a randomized encoding based framework," in *Proceedings of ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1–5.

[181] B. David-John, D. Hosfelt, K. Butler, and E. Jain, "A privacy-preserving approach to streaming eye-tracking data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2555–2565, 2021.

[182] J. Steil, I. Hagestedt, M. X. Huang, and A. Bulling, "Privacy-aware eye tracking using differential privacy," in *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, 2019, pp. 1–9.

[183] J. Li, A. R. Chowdhury, K. Fawaz, and Y. Kim, "Kaleido: Real-Time privacy control for Eye-Tracking systems," in *Proceedings of 30th USENIX Security Symposium*, 2021, pp. 1793–1810.

[184] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, vol. 2, 2003, pp. 1398–1402.

[185] B. Smith, Q. Yin, S. Feiner, and S. Nayar, "Gaze locking: Passive eye contact detection for human object interaction," in *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2013, pp. 271–280.

[186] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[187] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[188] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proceedings of the International Conference on Machine Learning*, 2018, pp. 2137–2146.

[189] C. Dwork, A. Roth, et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[190] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 1, pp. 3–37, 2022.

[191] L. Fan, "Image pixelization with differential privacy," in *32th IFIP Annual Conference on Data and Applications Security and Privacy*, 2018, pp. 148–162.

[192] H. Xue, B. Liu, M. Ding, T. Zhu, D. Ye, L. Song, and W. Zhou, "Dp-image: Differential privacy for image data in feature space," *arXiv preprint arXiv:2103.07073*, 2021.

[193] W. L. Croft, J.-R. Sack, and W. Shi, "Differentially private facial obfuscation via generative adversarial networks," *Future Generation Computer Systems*, vol. 129, pp. 358–379, 2022.

[194] Y. Wen, B. Liu, M. Ding, R. Xie, and L. Song, "Identitydp: Differential private identification protection for face images," *Neurocomputing*, vol. 501, pp. 197–211, 2022.

[195] T. Li and C. Clifton, "Differentially private imaging via latent space manipulation," *arXiv preprint arXiv:2103.05472*, 2021.

[196] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[197] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[198] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[199] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning*, 2019, pp. 6105–6114.

[200] X. Zeng, K. Cao, and M. Zhang, "Mobiledeeppill: A small-footprint mobile deep learning system for recognizing unconstrained pill images," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, pp. 56–67.

[201] A. Schulz and R. Stiefelhagen, "Video-based pedestrian head pose estimation for risk assessment," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1771–1776.

[202] H. Wu, J. Feng, X. Tian, E. Sun, Y. Liu, B. Dong, F. Xu, and S. Zhong, "Emo: Real-time emotion recognition from single-eye images for resource-constrained eyewear devices," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 448–461.

[203] A. N. Angelopoulos, J. N. Martel, A. P. Kohli, J. Conradt, and G. Wetzstein, "Event-based near-eye gaze tracking beyond 10,000 hz," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2577–2586, 2021.

[204] G. Zhao, Y. Yang, J. Liu, N. Chen, Y. Shen, H. Wen, and G. Lan, "Ev-eye: Rethinking high-frequency eye tracking through the lenses of event cameras," in *Advances in Neural Information Processing Systems*, 2024, pp. 62 169–62 182.

[205] T. Zhang, Y. Shen, G. Zhao, L. Wang, X. Chen, L. Bai, and Y. Zhou, "Swift-eye: Towards anti-blink pupil tracking for precise and robust high-frequency near-eye movement analysis with event cameras," *IEEE Transactions on Visualization and Computer Graphics*, pp. 2077–2086, 2024.

[206] P. Bonazzi, S. Bian, G. Lippolis, Y. Li, S. Sheik, and M. Magno, "Retina: Low-power eye tracking with event camera and spiking hardware," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5684–5692.

[207] V. Clay, P. König, and S. Koenig, "Eye tracking in virtual reality," *Journal of Eye Movement Research*, vol. 12, no. 1, 2019.

[208] A. Plopski, T. Hirzle, N. Norouzi, L. Qian, G. Bruder, and T. Langlotz, "The eye in extended reality: A survey on gaze interaction and eye tracking in head-worn extended reality," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1–39, 2022.

[209] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020, pp. 1–22.

[210] Y. Cheng and F. Lu, "Gaze estimation using transformer," in *International Conference on Pattern Recognition*, 2022, pp. 3341–3347.

[211] Q. Yu, Y. Xia, Y. Bai, Y. Lu, A. L. Yuille, and W. Shen, "Glance-and-gaze vision transformer," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, 2021, pp. 12 992–13 003.

[212] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3730–3738.

LIST OF PUBLICATIONS

The list of publications accomplished during PhD.

1. **Lingyu Du**, Xucong Zhang, Guohao Lan, *"Talk to Me, Not the Slides: A Real-Time Wearable Assistant for Improving Eye Contact in Presentations"*, **(Under Submission)**, 2025.

2. **Lingyu Du**, Yupei Liu, Jinyuan Jia, Guohao Lan, *"SecureGaze: Defending Gaze Estimation Against Backdoor Attacks"*, In Proceedings of the ACM Conference on Embedded Networked Sensor Systems **(SenSys)**, 2025.

3. **Lingyu Du**, Xucong Zhang, Guohao Lan, *"Resource-efficient Gaze Estimation via Frequency-domain Multi-task Contrastive Learning"*, ACM Transactions on Sensor Networks **(TOSN)**, 2025.

4. **Lingyu Du**, Jinyuan Jia, Xucong Zhang, Guohao Lan, *"PrivateGaze: Preserving User Privacy in Black-box Mobile Gaze Tracking Services"*, In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, **(IMWUT)**, 2024.

5. **Lingyu Du**, Guohao Lan, *"FreeGaze: Resource-efficient Gaze Estimation via Frequency-domain Contrastive Learning"*, In Proceedings of International Conference on Embedded Wireless Systems and Networks **(EWSN)**, 2023.

6. Tongyun Yang, Bishwas Regmi, **Lingyu Du**, Andreas Bulling, Xucong Zhang, Guohao Lan, *"Through the Eyes of Emotion: A Multi-faceted Eye Tracking Dataset for Emotion Recognition in Virtual Reality"*, In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, **(IMWUT)**, 2025.


TU Delft