UNIVERSITY OF CALIFORNIA
SANTA CRUZ

DATA-EFFICIENT REPRESENTATION LEARNING FOR GAZE ESTIMATION

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Swati

March 2024

The Dissertation of Swati
is approved:

Prof. Roberto Manduchi, Chair

Prof. James Davis

Prof. Xin Eric Wang

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Data-Efficient Representation Learning for Gaze Estimation

by

Swati

The human gaze serves as a potential non-verbal cue that enhances human-computer interfaces, enabling users to engage with devices through eye movements. The ability to accurately measure and interpret gaze direction plays a critical role in various domains, including social interactions, assistive technologies, augmented reality, and psychological research to examine cognitive state.

Over the past decade, gaze estimation has emerged as a prominent area of interest within the research community. Conventional gaze estimation methods rely on specialized hardware, including high-resolution cameras, infrared light sources, and image processing units, to detect eye features like the pupil center and iris boundary. While these devices offer greater accuracy and precision, their practical use is limited by factors such as high costs, restricted head movements, and limited range of allowable distances between user and device. As an alternative to dedicated gaze-tracking hardware, several techniques have been developed to infer gaze direction directly from eye images captured by standard cameras on personal devices such as laptops, tablets, and phones.

The recent emergence of deep learning techniques has enhanced learning-based gaze estimation approaches. These appearance-based gaze estimation methods directly map eye images to gaze targets without the need for explicit detection of eye features and, therefore, have a strong capability to work in unconstrained environments. However, the effectiveness of these approaches greatly depends on having access to extensive training datasets that include a variety of eye appearances, gaze directions, head poses, lighting conditions, and other variables. In this thesis, we focus on improving the adaptability and effectiveness of webcam-based gaze estimation techniques through the application of generative modeling and representation learning.

First, we propose an easy approach for calibrating a laptop camera with a commercial gaze tracker, streamlining the process of collecting labeled gaze data to make it readily accessible for all users. This dataset can then be utilized to enhance the accuracy of appearance-based gaze estimation methods for new users and different domains.

Second, we introduce a generative redirection framework designed to manipulate gaze direction and head pose orientation in synthesized images. This framework is used to generate augmented, gaze-labeled datasets, thereby enhancing the performance of gaze estimation methods.

Third, we explore self-supervised contrastive learning to acquire equivariant

gaze representations through an unlabeled multiview dataset. These gaze-specific representations are utilized for few-shot gaze estimation, enhancing the efficacy of user-specific models.

Finally, we present a spatiotemporal model for video-based gaze estimation, incorporating attention modules to enhance understanding of both local spatial and global temporal dynamics. Furthermore, we improve the performance of this model using person-specific few-shot learning through Gaussian processes.

List of Publications

- **Swati Jindal**, Mohit Yadav, Roberto Manduchi, "Spatio-Temporal Attention and Gaussian Processes for Personalized Video Gaze Estimation", *in submission.*

- **Swati Jindal**, and Roberto Manduchi, "Contrastive representation learning for gaze estimation", *in NeurIPS workshops 2023.* (Best Paper Award)

- **Swati Jindal**, and Xin Eric Wang, "Cuda-ghr: Controllable unsupervised domain adaptation for gaze and head redirection" *in WACV 2023.*

- **Swati Jindal**, Harsimran Kaur, and Roberto Manduchi, "Tracker/Camera Calibration for Accurate Automatic Gaze Annotation of Images and Videos" *in ETRA 2022.*

- Harsimran Kaur, **Swati Jindal**, and Roberto Manduchi, "Rethinking model-based gaze estimation", *in ETRA 2022.*

Dedicated to my parents for their constant love and support.

Acknowledgments

This thesis could not have been accomplished without the assistance and mentorship of numerous individuals. I am delighted to express my thanks to those who have made this achievement possible.

First and foremost, my heartfelt appreciation goes to my advisor, Prof. Roberto Manduchi, for his unwavering support and kindness throughout my PhD journey. Roberto's insightful feedback and recommendations have been instrumental at various stages of my research. Additionally, I wish to extend my gratitude to my thesis committee members, Prof. James Davis and Prof. Xin Eric Wang, for dedicating their efforts and time in reviewing my work and provide valuable feedback. I had the pleasure of collaborating with Prof. Xin Eric Wang whose vast research experience and guidance taught me necessary skills to effectively present my project.

I am grateful to my friend and collaborator, Dr. Mohit Yadav, with whom I engaged in numerous insightful research discussions. Mohit's encouragement was a constant source of strength throughout my PhD journey, helping me to stay positive even during challenging periods. Furthermore, I extend my thanks to my former colleague, Dr. Harsimran Kaur, for our successful collaborations on numerous projects. My gratitude also goes to my friend and colleague, Dr. Fatemeh Elyasi, who has been a companion through both the bitter and sweet

moments of my PhD journey, sharing in the experiences and challenges along the way. Also, thank you to my colleagues Brigit, Younsoo Park and Seongsil for their friendship, which was crucial in maintaining my sanity during the COVID-19 pandemic. I owe a debt of gratitude to my dear friend, Dr. Swati Jindal, for always being there for me and providing me emotional support during my PhD journey.

Embarking on this journey would have been unimaginable without the encouragement from my parents and family, who have always believed in me. Their unconditional love and support were helpful in pushing my boundaries and thriving in a foreign country. I cannot express the depth of my gratitude I have for you in a few words. From the bottom of my heart, thank you for everything.

Chapter 1

Introduction

Human eyes are vital organs for vision and allow us to perceive and make sense of the world around us. The eyes are important for sensing, and their movement serves as a key indicator in non-verbal communication and social interactions. Consequently, in recent years, there has been a growing trend of interactive systems, including virtual reality (VR) headsets, mobile or wearable devices, desktops, and robots, that utilize gaze as either a primary or supplementary mode of interaction.

Driver-assistance systems utilize gaze tracking to monitor the driver's attention level, helping to identify signs of distraction or fatigue [9, 10, 11]. Clinical practitioners use gaze behavior analysis to gain insights into mental health and assist in the diagnosis of autism [12, 13]. Additionally, gaze has shown great potential in human-computer interfaces, enhancing communication and mobility for individuals with physical impairments [14, 15, 16]. For instance, gaze can enable control of a screen's mouse pointer without physical contact [17], or it can be integrated into a system that detects the user's gaze direction to steer or move a wheelchair towards a targeted point [14]. Gaze tracking is also used as a proxy for human

visual attention, facilitating the study of cognitive and behavioral analysis [18, 19] and also has practical applications in the commercial sector [20, 21, 22].

Early gaze estimation techniques were invasive, employing electrooculography to track eye movements such as saccades, smooth pursuits, and fixations [23]. These methods involved attaching sensors around the eye and measuring potential differences to assess eye movement [24, 25]. As computer vision technology advanced, gaze estimation methods evolved to incorporate dedicated hardware, such as high-resolution cameras and infrared light sources [26, 27]. These methods are readily available in commercial eye trackers and require explicit eye feature detection, such as the pupil, iris, eye corners, and corneal-reflection detection. Commercial eye trackers can achieve high accuracy with an angular error of less than one degree in optimal conditions including indoor settings, restricted head movements, a suitable distance between the camera and the user, and person-specific calibration [28]. Despite their accuracy, the widespread adoption of commercial eye trackers in various applications remains challenging due to their high cost and the need for expert knowledge to set up and operate them. These limitations have led many research communities to explore gaze tracking methods that only require easily available off-the-shelf RGB cameras. These cameras can effortlessly be integrated with various personal devices or ambient displays, making gaze tracking more accessible and versatile. This thesis primarily concentrates on non-intrusive

eye tracking methods that utilize images captured by RGB cameras, like those embedded in laptops or smartphones.

In recent decades, the advancement of deep learning technologies and improvements in optical sensors has led to the emergence of automatic appearance-based gaze analysis from images [29]. The appearance-based methods directly learn a mapping from input eye images to gaze direction. They do not depend on explicit feature detection, which allows them to effectively process low-resolution images captured by standard RGB cameras and work well in unconstrained environments. In controlled laboratory settings, where factors like fixed head pose and good illumination is maintained, appearance-based gaze estimation methods can achieve a reasonable accuracy, typically around one to two degrees [30]. However, in real-world conditions, where users have the freedom to move their heads and operate in varying illumination environments, the accuracy of appearance-based gaze estimation methods tends to decrease. This decline in performance is also attributed to the lack of data that covers a wide range of variations.

Deep convolutional neural networks have been utilized in almost every appearance-based gaze estimation approach due to their ability to learn complex non-linear mappings. Nonetheless, these deep learning approaches require extensive annotated datasets to deliver accurate results in real-world scenarios. Generally, the collection of gaze-labeled datasets takes place in controlled labora-

tory environments, where variables like illumination, camera angles, and head pose are regulated. Therefore, acquiring large and accurately annotated datasets in unstructured real-world settings presents challenges, often requiring complex and costly hardware setups. As a result, cross-domain and few-shot person-specific gaze estimation models have seen significant advancements. These models enable the adaptation of trained gaze models to new domains and allow for tuning with just a few labeled samples from unseen users. However, fine-tuning over-parameterized deep neural networks with limited data can lead to overfitting. This issue can hinder their performance in more generalized settings.

Consequently, the main challenges in estimating gaze from webcam images include (a) acquiring large annotated datasets to enhance out-of-domain generalizability, and (b) adapting gaze estimation models to new users using as few labeled samples as possible, while still achieving performance improvements. In this thesis, we propose solutions to address these challenges, aiming to enhance the performance of webcam-based gaze estimation for both image and video inputs.

1.1 Thesis Contributions

Broadly speaking, the contributions presented in this thesis fall into two categories: developing efficient techniques to acquire annotated gaze datasets for appearance-based methods, and enhancing representation learning for the gaze

estimation task.

For the first category of solutions, we introduce a simplified method for calibrating commercial trackers to a user's laptop webcam. This approach facilitates the quick collection of large, gaze-labeled datasets in unconstrained settings. Additionally, we propose a controllable generative framework designed to create augmented gaze datasets in various domains. This augmented data is then utilized to enhance the performance of cross-domain appearance-based gaze estimation.

In the second category, our solutions concentrate on learning effective gaze representations that encompass an understanding of the gaze estimation task. We introduce a self-supervised representation learning framework that utilizes unlabeled image datasets. This framework enables few-shot adaptation of these representations, enhancing performance in both cross-domain and person-specific gaze learning. Additionally, we propose a spatio-temporal representation learning framework specifically for video gaze estimation. We further refine this framework by implementing few-shot personalization using Gaussian processes for unseen users, thereby improving its effectiveness.

In the following, we briefly describe the contributions presented in this thesis.

Camera-Tracker Calibration Modern appearance-based gaze tracking algorithms require vast amounts of training data, with images of a viewer annotated with "ground truth" gaze direction. The standard approach to obtain gaze an-

notations is to ask subjects to fixate at specific known locations, and then use a head model to determine the location of "origin of gaze". We propose using an IR gaze tracker to generate gaze annotations in natural settings that do not require the fixation of target points. This requires prior geometric calibration of the IR gaze tracker with the camera, such that the data produced by the IR tracker can be expressed in the camera's reference frame. This contribution introduces a simple camera-tracker calibration procedure based on the PnP algorithm and demonstrates its use to obtain a full characterization of gaze direction that can be used for ground truth annotation.

Gaze Redirection Generative modeling has shown excellent results in generating photo-realistic images, which can alleviate the need for annotations. However, adopting such generative models to new domains while maintaining their ability to provide fine-grained control over different image attributes, e.g., gaze and head pose directions, has been a challenging problem. We proposes an unsupervised domain adaptation framework that enables fine-grained control over gaze and head pose directions while preserving the appearance-related factors of the person. Our framework simultaneously learns to adapt to new domains and disentangle visual attributes such as appearance, gaze direction, and head orientation by utilizing a label-rich source domain and an unlabeled target domain. We empirically show that the proposed method can outperform state-of-the-art techniques on both

quantitative and qualitative evaluations. Further, we demonstrate the effectiveness of the generated augmented image-label pair dataset in the target domain for improving the performance of the downstream task of gaze estimation.

Gaze Representation Learning Self-supervised learning exploits contrastive learning to encourage visual representations to be invariant under various image transformations. However, the gaze estimation task demands not just invariance to various appearances but also equivariance to the geometric transformations. We propose a simple contrastive representation learning framework for gaze estimation, which exploits multi-view data to promote equivariance and relies on selected data augmentation techniques that do not alter gaze directions for invariance learning. Our experiments demonstrate the effectiveness of our method on both cross-domain and few-shot settings of the gaze estimation task.

Spatio-Temporal Personalized Video Gaze Estimation Video gaze estimation faces significant challenges, such as understanding the dynamic evolution of gaze in video sequences, dealing with static backgrounds, and adapting to variations in illumination. To address these challenges, we propose a simple and novel deep learning model designed to estimate gaze from videos, incorporating a specialized attention module. Our method employs a spatial attention mechanism that tracks spatial dynamics within videos. This technique enables accurate gaze

direction prediction through a temporal sequence model, adeptly transforming spatial observations into temporal insights, thereby significantly improving gaze estimation accuracy. Additionally, our approach integrates Gaussian processes to include individual-specific traits, facilitating the personalization of our model with just a few labeled samples. Experimental results confirm the efficacy of the proposed approach, demonstrating its success in both within-dataset and cross-dataset settings. Our method achieves state-of-the-art performance on the Gaze360 dataset, with or without personalization.

1.2 Structure of the Thesis

This section summarizes each chapter of the thesis along with the contributions.

Chapter 2 provides an overview of related research in gaze estimation methods and datasets. This chapter also delves into current state-of-the-art deep learning techniques, offering comparisons and contrasts with the methods we propose.

Chapter 3 details an algorithm for calibrating a commercial screen-based gaze tracker with a laptop's web camera. This chapter showcases a more efficient and accurate approach to gathering extensive gaze datasets and includes a comparative analysis with previous methods.

Chapter 4 presents a method for controllable gaze and head redirection within

an unsupervised domain adaptation framework. This chapter also highlights how the use of generated data through this model enhances performance in downstream tasks related to gaze and head pose estimation.

Chapter 5 introduces a self-supervised method for learning gaze representations that are invariant to visual appearances and equivariant to geometric transformations. This approach successfully enhances performance in both cross-domain and user-specific gaze estimation tasks, with a limited number of labeled samples.

Chapter 6 introduces a video-based gaze estimation method that utilizes spatial and temporal attention modules to track the dynamic evolution of eye movements. Additionally, this chapter discusses a few-shot personalization approach for new users, implemented through Gaussian processes.

Chapter 7 summarizes the thesis and discusses possible future directions in webcam-based gaze estimation.

Chapter 2

Background

Gaze estimation has been extensively studied over a long period, with numerous methods proposed over time. In this chapter, we begin by exploring the anatomy of the human eye and then move on to discuss various gaze estimation techniques. Subsequently, we will provide a concise overview of deep learning techniques utilized in gaze estimation.

2.1 The Human Eye

The eyes, as the primary organs of the visual system, are adept at receiving visual images, which are subsequently transmitted to the brain. Their functioning is analogous to that of a camera. Light reflected from an object enters the eyes through the pupil and is focused onto a plane to create an image. The cornea, a transparent outer layer, plays a key role in transmitting and focusing light into the eye. Some portion of this light enters through a circular aperture in the middle of the eye, known as the pupil. The size of the pupil is controlled by the iris, which is

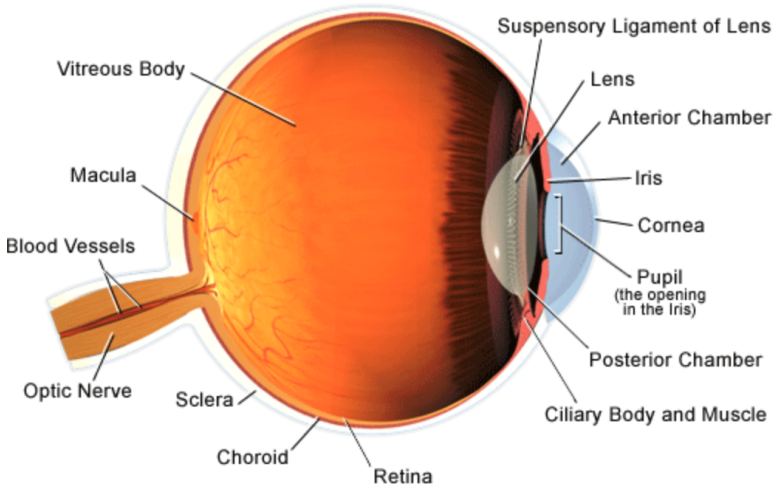


Figure 2.1: Internal anatomy of the eye[1].

the colored part of the eye. In bright light conditions, the iris contracts, reducing the size of the pupil to let in less light. Conversely, in low light conditions, the iris expands, enlarging the pupil to allow more light to enter. Subsequently, the light passes through the lens, which collaborates with the cornea to focus light rays onto the nerve layer lining the back of the eye called the retina. The retina senses light and creates electrical impulses that are sent through the optic nerve to the brain to produce vision. The internal anatomy of the eye is shown in Figure 2.1.

Figure 2.2 illustrates the external anatomy of the eye, comprising the sclera, limbus, iris, pupil, and eyelids. The eyelid is a flexible tissue comprised of skin and muscles that serve to protect the eyeball by enabling blinking. It is adorned with hundreds of eyelashes, which function to cover and shield the eyes from foreign

[1]https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-eye

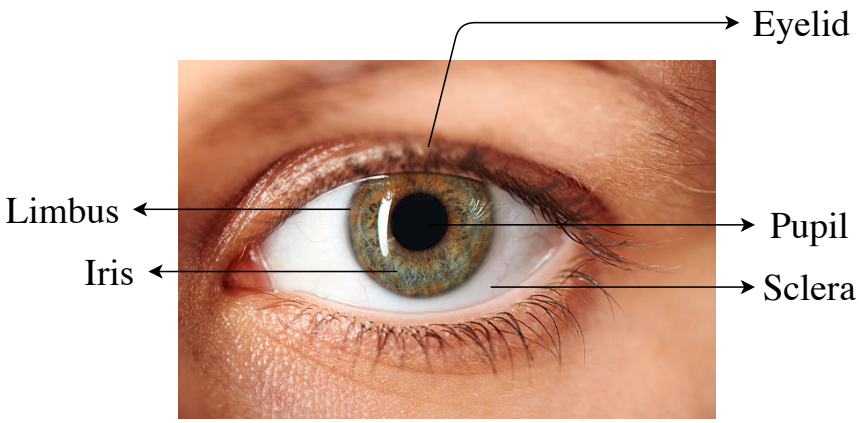


Figure 2.2: External anatomy of the eye

particles. The sclera is a white, opaque layer that envelops most of the eyeball's exterior. The boundary between the sclera and iris is called the limbus.

Eyes are not perfectly spherical; instead, they consist of a smaller anterior (front) segment joined to a larger posterior (back) segment. The anterior segment houses the cornea, iris, and pupil, whereas the posterior segment includes the vitreous body, retina, choroid, and sclera, which is the outer white shell of the eye. In the context of gaze estimation, the eyes are typically assumed to be spherical in shape, with a radius of approximately 12 – 13 mm. There are two axes that model the gaze direction: the optical axis and visual axis, as depicted in Figure 2.3[2].

The optical axis, also referred to as the Line of Gaze (LoG), is the line that passes through the pupil, cornea, and the center of the eyeball. The visual

[2]The *optic axis* of the eye is typically defined as a "line of best fit" through the centers of curvature of each refracting surface within the eye [31, 32]. It is different from the *pupillary axis*, which is the line passing through the center of the pupil and perpendicular to the corneal surface. The angle between the pupillary and the visual axis is known as the κ angle [33].

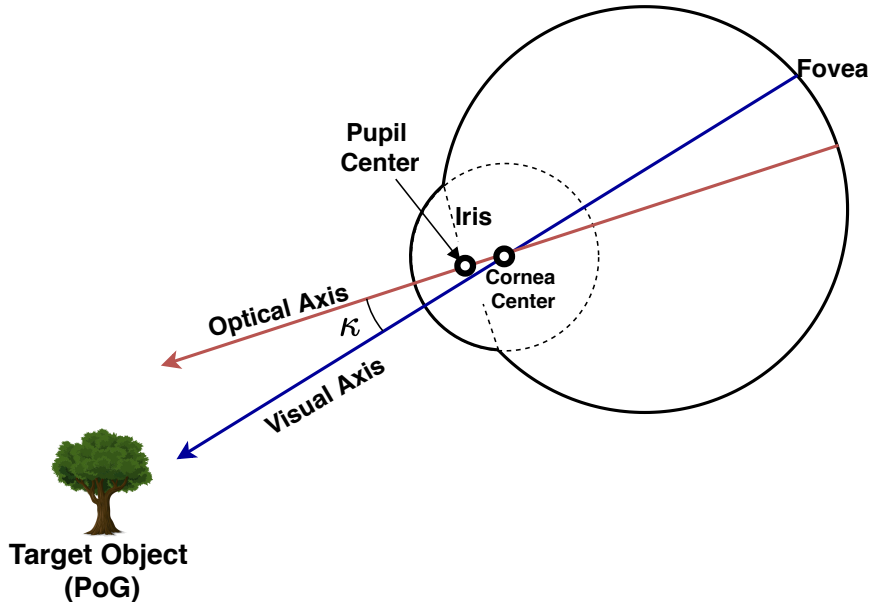


Figure 2.3: Visualization of the visual and optical axis on the eyeball model.

axis, also known as the Line of Sight (LoS), is the line that connects the fovea and the center of the cornea. This axis is regarded as the *true gaze direction*. Both the optical axis and the visual axis intersect at the *nodal point of the eye*, which is often approximated by the center of the cornea since they are in close proximity. There is a user-dependent angular offset between these two axes. This angular offset is typically determined through subject-dependent calibration in gaze tracking devices. The fovea, situated on the retina, is slightly off-center, positioned approximately $4 - 5°$ horizontally and $1.5°$ vertically below the optical axis [1, 34]. Consequently, the angular offset, also known as the kappa angle (κ), can vary among subjects, sometimes up to $3°$ [1]. Moreover, the head pose is crucial in determining the direction of the gaze. People typically align their head

movements with eye movements to effectively scan their environment. Therefore, the combined orientation of the head pose and the LoS offers insights into where the person is looking.

2.2 Gaze Estimation Methods

Early methods for gaze estimation involved identifying eye movement patterns like fixation, saccade, and smooth pursuit. This was achieved by attaching sensors around the eye and measuring potential differences [24, 25]. Contemporary gaze-tracking methods employ computer vision technologies and utilize images of the eye and face to estimate gaze directions.

Gaze tracking techniques are broadly categorized into two types: PCCR-based and Vision-based. PCCR-based (**P**upil **C**enter **C**orneal **R**eflection) methods utilize detection of eye features like corneal reflections for interpreting gaze direction. Conversely, vision-based methods rely on 2D images captured by standard cameras, employing machine learning to regress the 3D gaze direction directly from these images. The following subsections provide a brief overview of these techniques.

2.2.1 PC-CR Gaze Estimation

PC-CR gaze estimation techniques are commonly employed in commercial gaze trackers. These methods utilize infrared light sources to illuminate the eye. This

illumination leads to multiple reflections at the boundary between the eye lens and the cornea, resulting in the formation of *Purkinje images* [35]. In infrared-based gaze tracking, the first Purkinje image, commonly referred to as the glint, is predominantly used. PCCR-based methods focus on estimating the pupil center and corneal reflections (glint) to determine gaze direction. When a person looks directly at the light source, the glint and the pupil center align. However, as the person's gaze shifts away from the light source, the distance between these two points increases [36].

The infrared-based gaze tracking produces bright and dark pupil effects, which is useful for pupil detection [37]. The difference between dark and bright pupils is based on the location of the illumination source with respect to the optics. When the light source is near the camera's optical axis, it results in a bright pupil image, as most of the light is reflected back into the camera. Conversely, if the light source is positioned away from the camera's optical axis, the pupil appears dark in the image. Figure 2.4 shows the bright and dark pupil effect due to IR illumination.

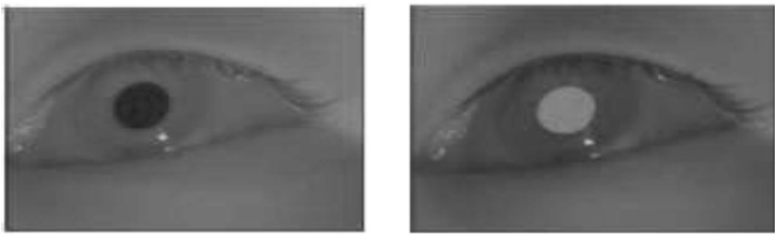


Figure 2.4: Dark (left) and bright (right) pupil effect under IR illumination.

Figure 2.5 provides an example of PCCR-based gaze estimation. In this method, a source of invisible near-infrared or infrared light is used to illuminate the pupil. This illumination leads to observable reflections in both the pupil and the cornea. These reflections are captured by an infrared camera, and various vision techniques [38, 39] have been proposed to robustly and accurately extract the centers of the pupil and the glint. The 2D vector, known as the PC-CR vector, which is formed between the pupil center and the glint image, is mapped to a 2D point of gaze or a 3D gaze direction. This mapping is achieved by fitting a polynomial function [40, 41], which is learned during a calibration procedure performed by the individual prior to using the eye tracker [37]. Figure 2.6 illustrates an example of corneal reflections along with the PC-CR vector.

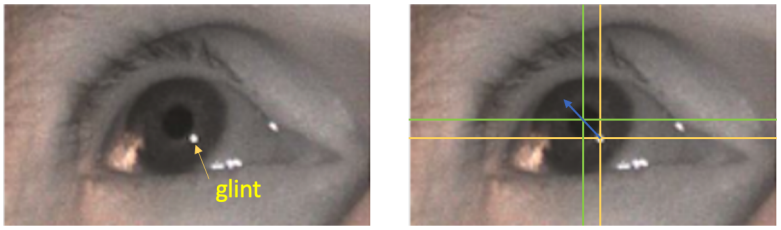


Figure 2.6: An example of a glint image (on the left) is shown alongside the PC-CR vector, which extends from the glint to the pupil center (on the right). This vector is used to map to either a 2D Point of Gaze (PoG) or a 3D gaze direction.

Additionally, several methods [1, 42, 43, 44, 45, 46] employ a 3D geometric model to estimate gaze direction. These approaches depend on various eye parameters, such as cornea radii, angles between visual and optical axes, refraction parameters,

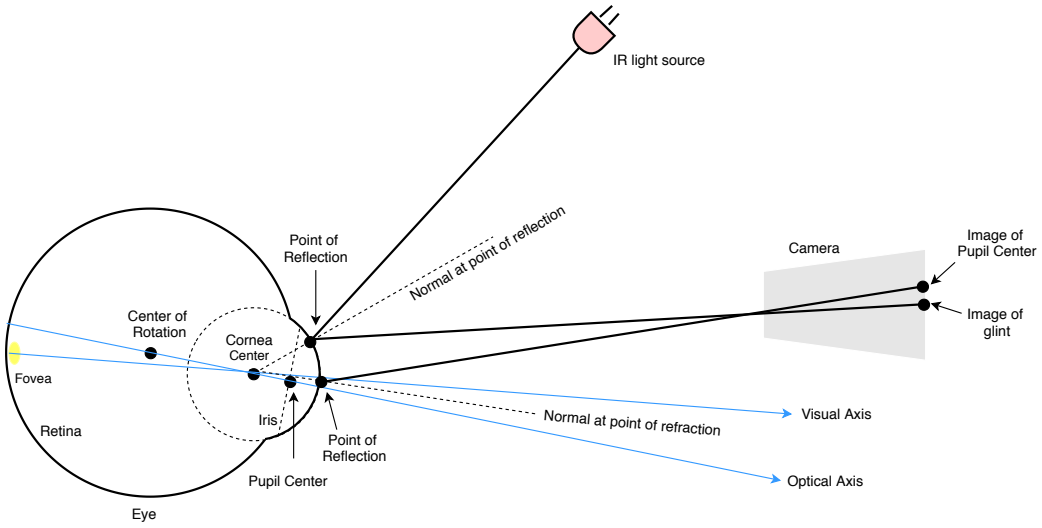


Figure 2.5: General idea of PCCR-based eye tracking showing eye model, infrared light source and camera capturing images of glint and pupil center. Reproduced from Guestrin and Eizenman [1].

iris radius, and the distance between the pupil and cornea centers. These techniques additionally necessitate camera calibration and a geometric model that includes the positions of the light sources, camera, and monitor. The fundamental strategy involves estimating the 3D locations of the cornea center and pupil center, which are used to estimate the 3D optical axis. Guestrin and Eizenman [1] demonstrate that with the use of a single camera and multiple light sources (≥ 2), it is feasible to easily estimate these locations even with free head movements. Since the visual axis cannot be directly measured from the image, the offset, known as the kappa angles, between the optical and visual axes is estimated by showing at least one

Figure 2.7: Example of Screen-Based Eye Tracker - Tobii Pro X2 eye tracker[3]

point on the screen. In fully calibrated setups, the point of regard is determined by intersecting the visual axis with the screen.

PCCR-based gaze tracking is primarily employed in two main types of eye trackers: screen-based and wearable trackers.

Screen-Based Eye Tracker Screen-based eye trackers, also known as remote eye trackers, operate remotely at a distance in a controlled environment. These devices are typically long, black rectangles with IR cameras and illuminators packed together, along with the processing unit. This unit is equipped with image detection capabilities, 3D eye modeling, and gaze mapping algorithms. Screen-based eye tracking devices are typically mounted directly below a screen, such as a monitor or laptop. These devices are used to record eye movements when users interact with stimuli displayed on the screen. They operate effectively within a certain distance range, approximately 50 – 90 cm from the user, which provides sufficient freedom for head movements.

[3]https://www.tobiipro.com/product-listing/tobii-pro-x2-30/

Wearable Eye Tracker Wearable eye trackers are designed to be positioned close to the eye or mounted on eyeglass frames, thereby allowing users to move freely. These trackers are particularly useful in real-world scenarios where users need to perform tasks while being monitored. They contain the same eye-tracking components as screen-based eye trackers. Wearable eye trackers are commonly recommended for behavioral studies and are frequently used in virtual reality.

Figure 2.8: Example of Wearable Eye Tracker - Tobii Pro Glasses 3[4]

2.2.2 Vision-based Gaze Estimation

In recent decades, due to advancements in computer vision, a wide array of vision-based gaze estimation methods has been developed. These methods directly utilize images of the eye or face to interpret the 3D gaze direction or 2D Point of Gaze (PoG). According to Hansen and Ji [37], vision-based methods are classified into three categories: 2D regression-based, 3D model-based, and appearance-based. Following this categorization, we briefly outline these gaze estimation approaches and discuss recent developments in each area.

[4]https://www.tobiipro.com/product-listing/tobii-pro-glasses-3/

2.2.2.1 2D regression-based

2D regression-based gaze estimation methods focus on identifying key points within eye images and use regression techniques to map these points to the gaze direction or Point of Gaze (PoG). Many of these methods involve infrared (IR) cameras to detect geometric features of the eye, like the pupil center and glints. With IR illumination, the PoG can be directly regressed from the pupil center-glint vector. As a result, methods that use IR do not necessitate geometric calibration for converting gaze directions into PoG. Mimica and Morimoto [47] apply least squares to map pupil-glint vectors to calibration markers using overdetermined linear equations formed by 2nd-order polynomials. Cherif et al. [48] presents an adaptive calibration technique that incorporates a secondary calibration for error correction and utilizes a higher degree polynomial to establish the mapping function. The findings from a single calibration indicate that increasing the order of the polynomial results in improved accuracy of gaze estimation. However, according to Cerrolaza et al. [49], enhancing the order of the polynomial does not lead to increased accuracy in gaze estimation due to variables like head motion, the number of calibration markers, the approach used for calculating the pupil-glint vector, etc.

Non-linear methods using artificial neural networks are also employed to learn the mapping function between calibration markers and corresponding pupil-glint

vectors. Demjén et al. [50] conducts a comparison between linear and neural network regression methods for gaze estimation, demonstrating that neural networks yield greater accuracy. Gneo et al. [51] utilizes two distinct multilayer feedforward networks, both using the same eye features as inputs, to compute the X and Y coordinates of the POG. Wu et al. [52] represents eye image features with an Active Appearance Model, which combines the shape and texture information in the eye region. Afterward, a support vector machine is used to classify 36 2D eye feature points (eye contour, iris, pupil parameters, etc.) into gazing directions. Wang et al. [53] introduces an enhanced DLSR-ANN (Direct Least Squares Regression-Artificial Neural Network) method specifically tailored for 2D gaze estimation.

Sesma et al. [54] employs the inner eye corner as a feature point, in contrast to using corneal reflections, and estimates gaze based on the PC-EC (Pupil Center-Eye Corner) vector. This method is simpler to implement and has been adopted in various works [55, 56, 57, 58]. However, since eye corners tend to shift when a person looks in different directions, they are not considered reliable key points for gaze estimation. Furthermore, Funes-Mora and Odobez [59] have utilized depth sensors in conjunction with RGB cameras for 3D rectification of eye images into a standard head pose viewpoint and scale. This technique aids in achieving head pose invariance but necessitates learning a person-specific 3D mesh model. Huang et al. [60] evaluates various combinations of feature extractors such as multi-level

Histogram of Oriented Gradients (mHoG) and examines four regression methods: k-nearest neighbors, random forest, gaussian process regression, and support vector regression. While feature-based methods are generally effective, their accuracy substantially diminishes in situations involving extreme head movements or when used in outdoor environments.

2.2.2.2 3D Model-based

3D model-based gaze estimation approaches use a physical model designed around the human eye's structure. These models are tailored to each subject and include geometric representations of the eye and parameters unique to each individual, like corneal radius and kappa angles. As discussed in Section 2.1, the human eye can be modeled with 3D geometry consisting of two spheroids, assuming the diameter of the average human eyeball as $\sim 24mm$ [61] and an average human iris as $\sim 12mm$ [62].

Some methods employ traditional computer vision techniques to identify key eye features, which are then used to predict gaze. Meyer et al. [63] and Hennessey et al. [26] are model-based methods that utilize edge detection techniques, such as the Canny edge detector [64], coupled with an ellipse fitting technique to accurately extract the most precise pupil contours. Once the pupil boundary is detected, a calibration process is undertaken to establish the relationship between the ellipse feature and screen coordinates. This is typically achieved using a homography

matrix. In a similar vein, Wang and Sung [65] focuses on observing the iris-sclera boundary and fitting an ellipse to this boundary. These ellipses are then matched to rotated and projected circles modeled on the surface of a sphere. To facilitate this approach, they utilize a camera equipped with a zoom lens to capture high-resolution images of the eye. Further, Takahashi et al. [66] suggests a method for estimating gaze direction by extracting the iris boundary. Their approach involves using a lookup table that links the shape of the iris with the corresponding gaze direction, designed to function in natural light conditions without requiring any extra light sources. Wood and Bulling [67] uses a commodity tablet device and employs elliptical model fitting to determine the iris boundary, followed by 3D back-projections to determine the optical axis and point-of-gaze.

Numerous gaze estimation techniques utilize depth camera technology to create a 3D model of the eyeball for estimating gaze direction. Jianfeng and Shigang [68] use an RGB-D camera, specifically the Kinect, to construct a head model and obtain 3D information about the pupil center. Similarly, Xiong et al. [69] focuses on tracking the iris center and facial landmarks. These landmarks are then projected onto a predefined face model, with their 3D locations determined using the Kinect camera. Wang and Ji [70] employs a hybrid approach that integrates facial features with eye models to robustly estimate the gaze point.

In most cases, methods that directly utilize single-camera images of the eye or

face are employed to model the 3D geometry of the eye and estimate gaze direction. Yamazoe et al. [71] construct a 3D face model over time by tracking keypoints in images and then apply this model to estimate head pose and determine the iris centers. Wang and Ji [72] utilize an offline 3D deformable eye-face model to estimate 3D eye gaze from observed 2D facial landmarks. In their approach, they conduct a joint optimization of person-specific eye position and visual axis offset parameters. Wood et al. [73] derived eye posture parameters by directly fitting a morphable model to the key features of the eye region and illumination, subsequently using these parameters to estimate gaze direction. Recently, deep learning techniques have been explored by Park et al. [74] for the estimation of the iris, eyeball center, and limbus landmarks from an eye image. These features are then used to estimate gaze direction. Although model-based gaze estimation methods offer higher precision, they typically require a time-consuming personal calibration process to accurately estimate parameters specific to each individual.

2.2.2.3 Appearance-based

Appearance-based gaze estimation methods do not necessitate specialized devices and can make use of standard web cameras to capture images of the eye or face. These methods directly process the pixel data from these images to produce either a 3D gaze direction or a 2D point-of-gaze. An early attempt at using CNNs for gaze estimation was presented in Zhang et al. [75]. The authors introduced

the concept of employing neural networks for appearance-based gaze estimation. They trained a LeNet-based architecture to map grayscale eye patches to 3D gaze vectors. Additionally, head pose information was incorporated by concatenating it to the fully connected layers preceding the final layer. Zhang et al. [76] expanded upon this approach by employing the VGG network [77], which resulted in a significant improvement in performance. Park et al. [74] employ DenseNet-based CNNs to map intermediate pictorial gazemap representations to 3D gaze, using these representations as a crucial step in their gaze estimation process. Later, Zhang et al. [78] employ CNNs to encode full-face images into feature maps and then learn spatial weights to either suppress or enhance information in various facial regions. This study demonstrates that full-face images yield better results than eye images for both 2D and 3D gaze estimation.

Several works [78, 79, 80, 81] employ separate CNN networks to compute features from both eyes and face and concatenate these features to estimate gaze direction. Krafka et al. [79] introduced a CNN model that processes fused features from the face, eye, and facial grid to deduce gaze direction in real time on a smartphone or tablet screen. This research highlighted that regions of the face other than the eyes also hold significant information for gaze inference. Chen and Shi [82] utilizes multi-input dilated-convolutional neural networks, which take as input both full-face images and two separate eye patches to infer gaze direction.

Cheng et al. [83] introduced a coarse-to-fine framework for gaze estimation, which initially estimates a gaze direction from the face image and then refines it using the corresponding residual predicted from eye images. Cheng et al. [80] proposes a face-based asymmetric regression-evaluation network (FARE-Net), which considers the asymmetry between the two eyes to enhance gaze estimation performance. Zhu and Deng [84] employs a gaze transform layer to integrate head pose and gaze direction for more accurate gaze estimation. Murthy and Biswas [85] demonstrate enhanced performance in gaze estimation by incorporating a difference layer, which removes common features between the left and right eye images. Subsequently, they apply an attention mechanism to assign weights to the features of each eye.

2.3 Deep Learning for Gaze Estimation

Recently, deep learning has achieved impressive results in the field of gaze estimation. This advancement is partly attributed to the development of comprehensive gaze datasets. In this section, we will provide a concise overview of these gaze datasets and explore a selection of deep learning techniques for gaze estimation that are relevant to this thesis.

2.3.1 Gaze Datasets

Gaze datasets are collected using specialized acquisition setups where individuals look at various targets or stimuli. These datasets are made available in the form of images or videos, along with the 3D gaze direction or 2D point-of-gaze. Some datasets are collected in controlled environments, while others are gathered in more natural, uncontrolled, or 'in the wild' settings.

Columbia [86] dataset was initially released for eye-contact detection in human-object interactions but was subsequently adopted by researchers in the field of gaze estimation as well. Columbia dataset contains 5,880 high-resolution images collected from 56 participants (32 male, 24 female) using a DSLR camera and provides images of resolution 5184×3456 pixels. The age of participants varies between 18 and 36 and shows a high range of diversity. Out of 56 people, 21 wore prescription glasses. It is collected in a controlled laboratory setting where subjects were asked to stabilize their heads using a chin rest and fixate on a grid of dots attached in front of them on a wall. For each subject, a combination of five horizontal head poses $(0°, \pm 15°, \pm 30°)$, seven horizontal gaze directions $(0°, \pm 5°, \pm 10°, \pm 15°)$, and three vertical gaze directions $(0°, \pm 10°)$ is acquired, giving a total of 105 images per subject. Some examples are demonstrated in Figure 2.9 and 2.10.

MPIIGaze [75] is a challenging gaze dataset containing 213,659 images collected

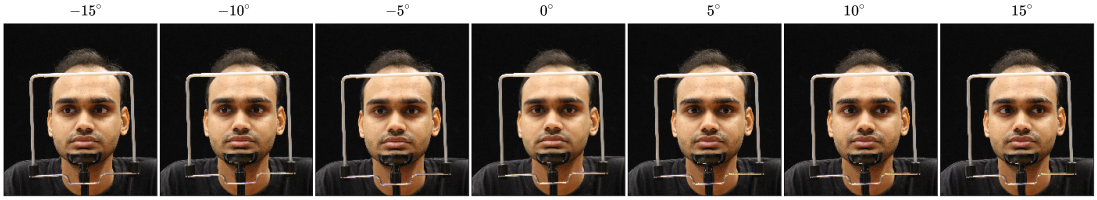


Figure 2.9: Columbia Gaze dataset contains 21 gaze directions for each head pose - three vertical and seven horizontal. The figure shows seven horizontal gazes for a particular vertical angle.

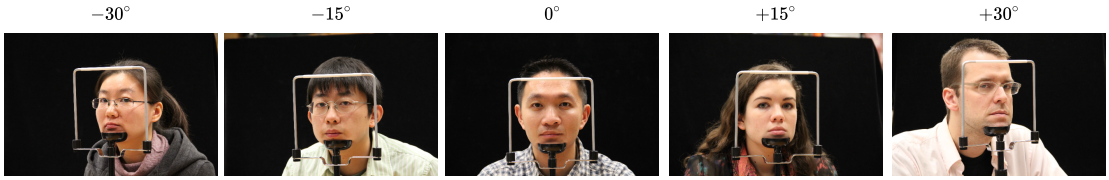


Figure 2.10: Columbia Gaze dataset contains five discrete horizontal head poses varying from $-30°$ to $30°$.

from 15 subjects during natural everyday events in front of the laptop over the course of three months. The number of images collected by each participant varies in the range between 34,745 and 1,498. It is collected by installing a data collection software on their laptops, which shows a random sequence of 20 on-screen markers where users are asked to fixate. As MPIIGaze is collected in the real world, it shows higher within-subject variations in appearance, such as illumination, make-up, and facial hair. Furthermore, Zhang *et al.* [78] provides a subset called *MPIIFaceGaze* containing 37667 face images with the hypothesis that the entire face provides more accurate information about the gaze. A few examples are shown in Figure 2.11.

GazeCapture [79] is a large-scale dataset collected through the Amazon Me-

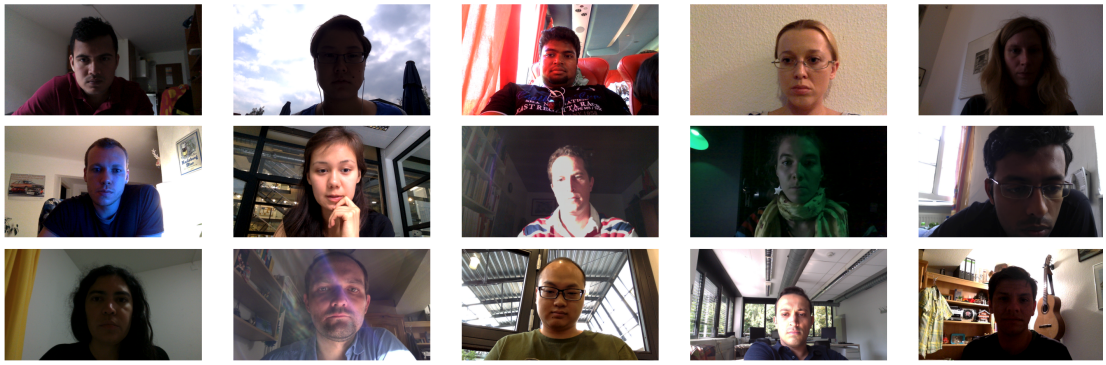


Figure 2.11: Example of images from MPIIGaze dataset.

chanical Turk (AMT) platform where workers were provided detailed instructions, from downloading the application from Apple's App Store to collecting data. 1249 subjects used iPhones while 225 used iPads, giving a total of ∼2.1M and ∼360k frames from each of the devices, respectively. Figure 2.12 shows examples of this dataset. GazeCapture contains around 2,445,504 frames from 1474 subjects collected when participants are asked to fixate on 13 fixed screen gaze markers for a total duration of ∼10 minutes. The authors also provide a predefined split of the train, validation, and test consisting of 1271, 50, and 150 subjects, respectively. This results in $1,251,983$, $59,480$, and $179,496$ frames, respectively, in train, validation, and test split.

ETH XGaze [87] dataset consists of over one million high-resolution images of varying gaze under extreme head poses. The dataset is collected from 110 participants (47 female and 63 male), aged between 19 and 41 years. For each gaze point, a total of 18 images (of resolution 6000 × 4000 pixels) were collected by the

Figure 2.12: Samples from GazeCapture dataset taken using iPhones or iPads.

18 different digital SLR cameras. A custom hardware setup is used with adjustable illumination conditions and a calibrated system to record ground-truth gaze targets. The participants were asked to focus on a randomly appearing shrinking circle and click the mouse when the circle became a dot, providing the gaze point. A few examples are provided in Figure 2.13.

Figure 2.13: Example of images from ETH-XGaze dataset captured under different head poses and lighting conditions.

Figure 2.14: Samples of EyeDiap dataset recorded from RGB-D camera (shown in left two images) and HD-camera (right two images). Reproduced from [2].

EyeDiap [2] dataset is collected using RGB-D (Microsoft Kinect) and HD cameras with an ensemble of five LEDs. The data is recorded in a total of 94 sessions from 16 participants (12 males, 4 females). In each session, participants were seated in front of the setup, within the cameras' field of view, and asked to observe various visual targets. These targets included a 3D ball within the scene and a circle displayed on a computer screen, either in a discrete manner (appearing at random locations) or in a continuous manner (following a random trajectory). Some examples from the EyeDiap dataset are illustrated in Figure 2.14, taken from the original paper.

Gaze360 [88] is a large-scale gaze-tracking dataset designed for robust 3D gaze estimation in uncontrolled environments. It includes data from 238 subjects, collected in 5 indoor (53 subjects) and 2 outdoor (185 subjects) settings, covering a broad range of head poses and distances. The ground truth for the dataset was established using a Ladybug5 360° panoramic camera mounted on a tripod at the center of the scene. Alongside it, a large, movable rigid target board marked with

an AprilTag[5] and a cross was used. Subjects participating in the data collection were instructed to fixate on this cross continuously. Overall, the Gaze360 dataset comprises 129K images for training, 17K images for validation, and 26K images for testing, each annotated with gaze information. Figure 2.15 shows some samples of the Gaze360 dataset, reproduced from the original paper.

Figure 2.15: Gaze360 samples with ground truth gaze directions (yellow arrow).

EVE [89] is an end-to-end video-based eye-tracking dataset collected in a constrained indoor setting using Tobii Spectrum Eye Tracker. It contains around 12 million frames (1920 × 1080 pixels) collected from 54 participants (30 male, 23 female, 1 unspecified) and consists of 4 camera views. The dataset was collected when participants were shown 1327 unique visual stimuli (1004 images, 161 videos, and 162 Wikipedia pages) on a 25-inch screen display, adding up to approximately

[5]https://april.eecs.umich.edu/software/apriltag

105 hours of video data. It provides 2D PoG and 3D gaze direction labels along with pupil size annotations and screen content videos. The authors also provide pre-processed images of eye patches and face images. Samples for this dataset are shown in Figure 2.16.

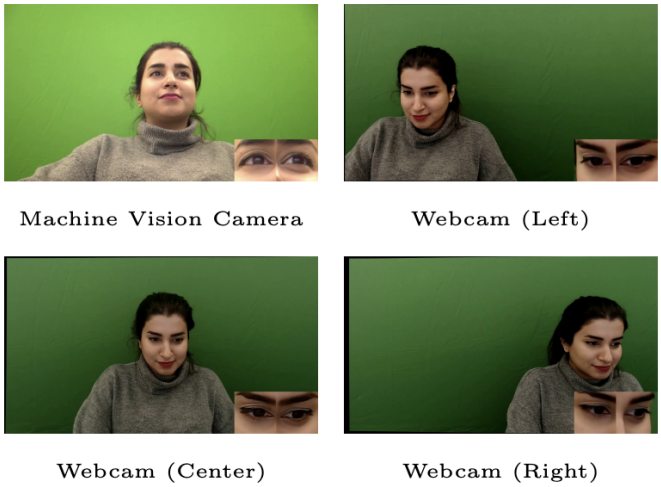


Figure 2.16: Example of frames collected from the 4 camera views with example eye patches shown as insets from the EVE dataset.

2.3.2 Redirection Methods

A variety of techniques have been developed for gaze redirection, focusing on synthesizing realistic images that not only maintain a high degree of realism but also effectively alter the perceived direction of gaze. Such approaches focusing on redirecting the gaze in real images have also emerged as an alternative method for acquiring training data for gaze estimation. Kononenko and Lempitsky [90]

introduces a method for gaze redirection that involves pixel-wise replacement using an eye flow tree, enabling the synthesis of realistic images with gaze directions adjusted upwards by $10 - 15°$. To achieve this, they use a deep warping network to modify the eye flow tree. This network is trained on eye image pair, consisting of the appearance of the eyes before and after the redirection. Ganin et al. [91] apply a deep convolutional network that incorporates coarse-to-fine warping and pixel correction to produce images with the redirected gaze. Similarly, Yu et al. [92] utilizes a deep neural network to learn the warping flow field between images, accompanied by a correction term, for the purpose of gaze redirection. Furthermore, Wood et al. [93] employs a graphics pipeline for redirecting eye images and fitting a multi-part eye region morphable model using an analysis-by-synthesis approach. This method can simultaneously recover the shape, texture, pose, and gaze of the eye region from a given image. Subsequently, it manipulates the eyes by warping the eyelids and rendering the eyeballs in the output image. This approach is particularly effective for large redirection angles, achieving superior results.

Generative Adversarial Networks (GANs), known for their efficacy in image generation tasks, have also been adopted by researchers for the purpose of gaze redirection. Chen et al. [94] developed a coarse-to-fine eye gaze redirection model that merges warping flow field techniques with adversarial learning to create high-quality redirected images. Additionally, they introduced a numerical and pictorial

guidance module, which serves to enhance the accuracy and precision of the gaze redirection process. He et al. [95] propose a GAN-based framework that utilizes a cycle consistency loss to learn gaze redirection and generate images with a high resolution. Kaur and Manduchi [96] propose a style-based approach for eye image redirection. In their method, they utilize mask-based generator networks [97] to manipulate and control the gaze direction vector.

Gaze redirection has also been employed as an auxiliary task for gaze representation learning. Park et al. [3] employ a transforming encoder-decoder based network [98, 99] to learn the disentanglement of gaze, head pose, and appearance in the latent space. The authors demonstrate how the gaze latent can be applied to enhance the performance of personalized gaze estimation tasks. Similarly, Zheng et al. [4] utilizes a transforming encoder-decoder network for gaze redirection and is designed to control both labeled factors (such as gaze and head pose) and pseudo-labeled factors (like appearance and lighting).

Xia et al. [100] proposes a framework for controllable gaze redirection that achieves both precise redirection and continuous interpolation through conditional image-to-image translation. Jin et al. [101] have developed a latent-to-latent framework that projects latent vectors into an embedding space. This approach allows for an interpretable redirection focused on specific desired attributes, like gaze and head pose, while preserving other attributes, such as appearance. The

framework is designed to ensure that there is no loss of information throughout this redirection process. Recently, neural-radiance fields (NeRFs) [102] have been utilized by Ruzzi et al. [103] for gaze redirection. In their approach, the authors focus on disentangling the features of the face and eye regions. This disentanglement enables the rigid transformation of the eyeballs to a specified gaze direction.

In Chapter 4, we present a gaze redirection framework that emphasizes controllability, learned through a combination of feature disentanglement and an unsupervised domain adaptation method. Our approach demonstrates a high level of photo-realism in the generated images and proves to be effective in enhancing the performance of gaze and head pose estimation tasks.

2.3.3 Representation Learning

Recent advancements in appearance-based gaze estimation methods have been significantly driven by deep learning. These methods require a substantial amount of labeled data to realize their full potential in terms of accuracy. Due to the significant reliance on labels by appearance-based methods, efforts have been made towards unsupervised and self-supervised gaze representation learning techniques. Yu and Odobez [7] is a pioneer work in learning low-dimensional gaze representations without requiring any gaze annotations. Their approach hinges on a gaze redirection network and utilizes the difference in gaze representation between the

input and target images as the redirection variable. They employ a redirection loss in the image domain, which facilitates the joint training of both the gaze redirection network and the gaze representation network. Subsequently, Sun et al. [8] introduces the cross-encoder network, which implements a latent code swapping strategy on image pairs. These pairs are presumed to be consistent in either gaze (for example, the left and right eyes of the same face) or eye appearance (such as the left eyes of the same face at different times). In this approach, each eye image is encoded into two distinct features: gaze and appearance. The cross-encoder then reconstructs images in the eye-consistent pair using their own gaze features combined with the appearance features of the other image, and vice versa for the gaze-similar pair, where each image is reconstructed using its own eye features and the gaze features of the other. Building upon this, Gideon et al. [104] further extend this concept by leveraging synchronized multi-view gaze video datasets, like EVE [89]. They employ the cross-encoder to encode images taken from various camera viewpoints into four distinct features: head pose, eye appearance, gaze relative to head pose, and common features shared across views. They implement a similar latent swapping mechanism, designed to enforce consistency among these features. This consistency is dependent on whether the images are sampled across different camera viewpoints, between the left or right eye, or across different points in time. These methods all make use of an encoder-decoder type of framework,

which demands a considerable number of parameters for effective representation learning.

In Chapter 5, we present a self-supervised approach based on contrastive learning for learning gaze representations. This method is inspired by the computational efficiency of contrastive self-supervised learning methods compared to generative approaches [105]. Similar to Gideon et al. [104], our work utilizes multi-view data and focuses on enforcing equivariance and invariance within the learned gaze representations.

2.3.4 Temporal Gaze Modeling

Following the release of video gaze datasets [2, 88], several temporal gaze estimation models have emerged. These models are designed to predict the direction of eye gaze from a sequence of images. The initial work of Palmero et al. [106] employs a recurrent CNN framework in which the static features of the face, eye region, and facial landmarks, extracted from each frame, are concatenated. These combined features are then input into a recurrent module, which is responsible for predicting the 3D gaze direction of the final frame in the sequence. Similarly, Kellnhofer et al. [88] proposes a video gaze tracking model that employs a bidirectional LSTM [107]. This model processes input from both past and future frames and outputs a single element. The authors use sequences

of 7 frames to predict the gaze direction of the central frame in the sequence. Wang et al. [108] released a dataset that captures human eye images and the corresponding ground-truth gaze positions on a screen while subjects engage in activities like browsing websites or watching videos. They proposed a dynamic gaze transition network to detect the transitions of eye movements over time and refine static gaze predictions using the dynamics learned from these transitions. Recently, Park et al. [89] collected a large-scale video-based eye-tracking dataset with ground-truth Point of Gaze (PoG) on a screen. The authors propose to jointly consider the spatio-temporal evolution of visual stimuli, as represented by screen content videos, to enhance the accuracy of PoG estimates on the video data.

In Chapter 6, we introduce a spatio-temporal model specifically designed for video gaze estimation. This model incorporates attention networks to capture the intricate spatial and temporal dynamics involved. Our empirical findings demonstrate that the application of both spatial and temporal attention mechanisms can significantly enhance the performance of video gaze tracking.

2.3.5 Few-shot Personalization

As discussed in Section 2.3.3, appearance-based methods for gaze estimation are highly dependent on extensive annotated datasets to achieve high accuracy. However, this reliance poses a significant challenge in real-world scenarios, particu-

larly when it comes to generalizing these models to unseen users. Various methods have been proposed to adapt pre-trained models to new users by employing only a small number of labeled samples for each individual. Liu et al. [109] utilize a two-branch differential network that is capable of predicting the differences in gaze direction for the same subject. As a result, during the inference stage, the gaze direction of a new sample can be predicted using just a few subject-specific calibration samples. Park et al. [3] leverages the learned gaze representations and applies meta-learning [110] to develop person-specific gaze networks using only a few examples. Chen and Shi [5] introduces a gaze decomposition method that involves learning a person-dependent bias during training. If no labeled samples are available for a new user during inference, this bias is set to zero. Otherwise, the bias is estimated using these few labeled samples.

In Chapter 6, we utilize Gaussian Processes to personalize the video gaze estimation model. This involves learning an additive bias correction model specific to each individual using only a few labeled samples. We demonstrate that personalization notably improves performance. Moreover, a key advantage of our approach is the ability to generate uncertainty estimates for each predicted gaze direction. This feature is particularly beneficial in identifying and eliminating erroneous predictions.

Chapter 3

Camera-tracker calibration for accurate gaze annotation of images and videos

3.1 Introduction

There has been substantial recent interest in appearance-based eye gaze tracking technology. The most successful such systems are based on machine learning algorithms that require an extensive amount of annotated image data sets for training. Compared to other applications (e.g., image classification), image annotation for gaze tracking has a fundamental difficulty because it is hard to measure one's gaze direction precisely by observing a picture. The standard method for annotating images with gaze direction is to ask subjects to fixate at a known location on a screen, typically identified by a specific marker. This procedure directly yields the *gaze point*, in screen coordinates, or, if the camera is geometrically calibrated with the screen, in camera coordinates. Estimation of the actual gaze direction requires

identification of the 3-D location of another point along the visual axis (referred to as *gaze origin*), which usually implies computing head pose. Several popular data sets have been built this way (see Section 3.2).

This standard procedure, however, has two main drawbacks. First, it inherently generates only sparse samples. Second, the accuracy of gaze direction annotations may be impaired by various factors. For example, it is well known that, during fixation, the gaze is not entirely stable. A study with 5 subjects [111] showed horizontal and vertical fluctuations during visual fixation with a standard deviation of approximately 0.1° in both directions, while a later study with 3 subjects [112] found the average fixational area (defined as the solid angle in which gaze remains for 95% of the time during fixation) to be of 1.2°. Larger deviations were measured for subjects who were myopic [113] or had other forms of visual loss [114]. The errors in pose estimation also contribute to gaze direction errors. For example, for a viewer located at a distance of 50 cm from the camera, a 2 cm depth estimation error results in a 1° gaze reconstruction error when the visual axis is at 30° from the camera's optical axis. In order to enable the acquisition of large, accurately annotated image data sets, some researchers have resorted to synthetic eye images based on carefully designed 3-D models [115, 116]. However, these synthetic images may not represent real-world conditions, as they may fail to model the diversity of morphological characteristics of human faces or the complex photometry of

illumination and reflection.

In this chapter, we propose the use of infrared (IR)-based gaze tracking devices to annotate image data acquired by a camera (such as the webcam in a laptop computer). IR-based gaze tracking is a mature technology [1, 117], with a number of commercial devices available for different market sectors such as video games [118], virtual reality [119], user interface [120], marketing research [121], and optometry [122]. We are considering here desktop-based trackers, which are typically attached to the bottom of a computer screen, and in particular, head-pose free trackers that let the user move their head within a certain range of locations and orientations.

An IR-based tracker is capable of producing high-rate, time-stamped measurements, including those of the visual axis, synchronized with images captured by the camera. This setup provides the necessary annotations. Unfortunately, this data is expressed in reference to the tracker frame, not the camera frame. Our main contribution is introducing an easy-to-use procedure to compute the rigid transformation between the two systems. This knowledge allows us to express all geometric-based annotations with respect to the camera frame. This approach is particularly useful, for instance, in training appearance-based gaze tracking algorithms. Our method requires a user to directly look at the camera for a few seconds while moving their head to different locations in front of the screen. Note

that the camera-tracker calibration is user-independent; that is, it will also work for any other users, provided that the relative geometry of the camera and tracker is not modified. In practice, this means that the tracker should remain rigidly attached to the screen after calibration, or another calibration would be called for. After calibration, large data sets with images and desired annotations can be collected without requiring the subject to fixate on specific points on the screen.

In general, we may expect to obtain reliable gaze data annotation by using specialized IR-based tracking devices. For example, the Tobii Pro Nano tracker used in our experiments has a nominal accuracy (average error or bias) of 0.3°, and a nominal precision (RMS error across samples) of 0.1° in optimal conditions (also see [28, 123] for in-depth performance analysis of a lower quality IR tracker, the Tobii EyeX). However, any residual error in the proposed gaze camera-tracker calibration procedure will contribute to errors in the measured visual axis. Note that our approach can only produce annotations when the user is within the operating range of distances from the tracker (45–85 cm for the Tobii Pro Nano).

3.2 Ground-Truth Gaze Annotation

3.2.1 Fixation Method

Several gaze-annotated image data sets have been created and made available for training and testing appearance-based gaze algorithms [79, 86, 87, 124, 125, 126]. To create these data sets, participants were asked to move their heads while fixating at specific known locations (known as *gaze points*). The visual axis for each eye in both images is estimated by determining a "gaze origin", a generic term for a point on the visual axis located within the eyeball. This is normally obtained using a face model. Note that face models are commonly employed for image normalization [124, 127], with the purpose of canceling out most of the head pose variability. Typically, the normalization procedure begins with face detection [128], followed by the detection of facial landmarks [129]. These landmarks are matched with a reference 3-D face model, such as the Surrey Face Model [6]. For example, Gross et al. [130] selects 4 eye corners and 9 nose landmarks to estimate the head pose using the Perspective-n-Point (PnP) method [131]. The 3-D gaze origin is commonly identified as the midpoint of the line connecting the eye corners. The visual axis is subsequently derived by joining the gaze origin with the gaze point.

3.2.2 Using an IR Gaze Tracker

An IR tracker computes the *pupillary axis*, that is, the line through the center of corneal curvature and the center of the pupil [31, 32]. The orientation of the visual axis with respect to the pupillary axis is described by the two *kappa* angles, which are estimated via per-individual calibration that involves fixation on a number of target points on the screen. Note that this calibration procedure is different from the proposed camera-tracker calibration. High-end two-cameras, two-illuminators IR trackers can produce measurements with high accuracy while allowing for free head motion (within a certain range of distance and head orientations).

IR gaze trackers normally provide access through their API (e.g., the Tobii Pro SDK) to two relevant measurements: (1) the *gaze point*, or point of regard, which is the intersection of the visual axis with the screen, expressed in the screen's reference frame (in pixels units); and (2) the *gaze origin*, which is a point on the visual axis contained within the eyeball, expressed in mm in the tracker's reference frame. While it is reasonable to think that the returned location of gaze origin may be at the corneal center of curvature [1], the Tobii documentation does not give any detail on its actual location besides it being within the eyeball.

Employing an IR tracker for automatic gaze annotation, instead of depending on discrete fixation targets, could facilitate data collection in more 'natural' scenarios, like reading text. This approach also has the potential to simplify the process of

gathering larger datasets. Prior work [89] utilized gaze point information from an IR gaze tracker as an alternative to the location of a fixation pattern. Adhering to the standard procedure outlined above, the visual axis is then estimated by connecting the transformed location of the gaze point, now in the camera's frame of reference, with the gaze origin point, which is estimated using a face model. In this work, we propose using the tracker to provide not only the gaze point but also the gaze origin. It is reasonable to expect that the sophisticated procedure an IR tracker uses to estimate the gaze origin location, which involves corneal reflection from a system with two projectors calibrated with two cameras [1], would yield more reliable results than a purely image-based algorithm relying on a general 3-D model. However, to utilize the gaze origin position estimated by the tracker, it is first necessary to find the relative pose of the camera with respect to the tracker such that the gaze origin can be expressed in the camera's reference frame. In the next section, we propose a simple calibration procedure that accomplishes that.

3.3 Camera-Tracker Calibration Algorithm

In the subsequent discussion, we will assume that the camera and the tracker are rigidly connected to each other. A common setup involves a tracker attached to the bottom of a screen of a laptop or desktop computer, with the camera embedded in the screen. Practically, if a gaze tracker is re-positioned each time it is used

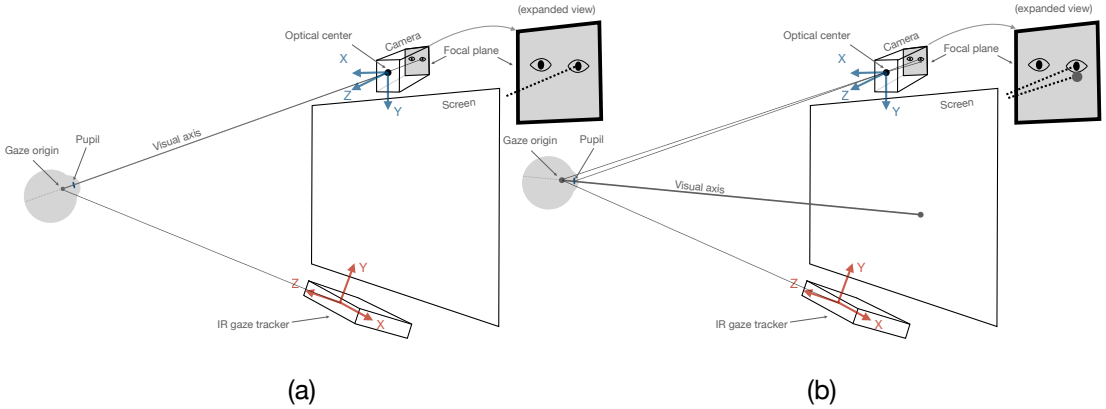


Figure 3.1: (a) When the user looks directly at the camera, the visual axis of either eye intersects with the camera's optical center. As a result, the gaze origin of that eye projects within the image of the pupil. (b) When the user looks away from the camera, the gaze origin might project outside the pupil's image.

(for instance, at the start of a session), it would necessitate a new calibration. Additionally, we assume that the intrinsic camera parameter matrix K and the radial distortion parameters have already been estimated.

Our goal is to estimate the relative pose $(R_t^c, \mathbf{T}^c)$ of the IR tracker with respect to the camera, such that a 3-D point $\mathbf{p}^t$ in the tracker's reference frame can be expressed in the camera's frame as $\mathbf{p}^c = R_t^c \mathbf{p}^t + \mathbf{T}^c$. We propose a calibration procedure that uses the 3-D location of the gaze origin for either eye, as estimated by the tracker and hence expressed in the tracker's reference frame. If we can determine the location of the projection of these points onto the camera's focal or image plane and create multiple pairs (3-D location – 2-D projection) as the user moves their head to different locations, we can employ the Perspective-n-

Point (PnP) algorithm [132] to calibrate camera and tracker. PnP determines the camera's pose based on the images of 3-D points in space with known locations, e.g., 3D gaze origin and their corresponding 2D pupil center images, as shown in Figure 3.1. While PnP is commonly used with a single image containing multiple known 3-D points in space, our proposed procedure, which involves multiple images, each with two known points in space (the gaze origin of the left and right eye), is also valid.

The challenge at hand is to determine the projection of the gaze origin onto the camera's focal plane. An eye's gaze origin is not directly observable, so there is no simple way to identify it in an image of the user, except for one specific situation: *when the user is looking directly at the camera.* In this case, it can be assumed that the image of the gaze origin for either eye is located within the image of that eye's pupil. This assumption is based on the fact that the visual axis is approximately crossing the camera's optical center when the user is looking directly at the camera. Hence, all points within the visual axis project onto the same pixel in the camera's focal plane. Given that the visual axis contains the gaze origin and that the visual axis can be expected to go through the pupil, it follows that the image (projection) of the gaze origin should be contained in the pupil's image. Note that the same should not be expected when the user gazes at a point that is away from the camera, e.g., at the opposite end of the screen from

where the camera is located (see Figure 3.1). For simplicity and due to the lack of precise information, we will assume that the image of the gaze origin when the user is looking at the camera is positioned at the center of the pupil image. This assumption allows us to use PnP with the gaze origin as a 3-D point and the pupil center as its projection on the image.

The calibration process is as follows: the user is instructed to move their head to various positions within the gaze tracker's coverage range. At each position, the user is prompted to look at the camera, and one or more images are captured. For each image, the locations of the pupils in both eyes are determined. With this information, PnP can be applied to the pairs (3-D gaze origin – 2-D pupil center) to compute the calibration parameters $(R_t^c, \mathbf{T}^c)$. In the next section, we outline some specifics of our implementation of this calibration procedure.

3.3.1 Implementation Details

For our experiments, we attached a small brightly colored paper ring around the camera of the laptop used for data acquisition (MacBook Pro). This colored ring served as a visual guide to help users clearly identify the camera's position for effective data collection. Participants were instructed to position their heads at various distances from the camera and at multiple vertical and horizontal locations within an imaginary cube of approximately $300 \times 300 \times 300$ mm in size. At each

location, users were then prompted to either gaze at the colored ring or directly at the camera and press a key. Following this, images and gaze data were collected for approximately 3 seconds, which equated to roughly 10 frames of data acquisition.

The automatic detection of the pupil center location was achieved using the algorithm described in [74]. This algorithm computes a set of visual landmarks from the image, which are detected from heatmaps generated by a stacked-hourglass network [133] trained on synthetic eye images (UnityEyes [115]). We take the pupil center to coincide with the midpoint of the "iris boundary" landmarks. From the 3-D locations of gaze origin (computed by the gaze tracker) and 2-D location of the pupil center detected in each image, we compute the calibration parameters $(R_t^c, \mathbf{T}^c)$ using PnP. We use the `solvePnPRansac`[1] implementation of PnP from OpenCV. This algorithm can take an arbitrary number of points and is robust to the presence of outliers. Outliers may occur, for example, when the user's gaze unintentionally moves away from the camera.

3.4 Experiments

We conducted a study with five participants (four female, one male), with two main goals: (1) evaluate the accuracy of the proposed calibration algorithm; (2) compare the location of the gaze origin estimated by the IR tracker with that

[1]https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga50620f0e26e02caa2e9adc07b5fbf24e

estimated using a face model, and assess the discrepancy between the visual axes computed with the two methods. All participants did not wear eyeglasses, and images were taken in a well-lit environment. During the calibration process, it is important to ensure optimal imaging conditions. However, once calibrated, the system can be used for any users and under various lighting conditions without the need for recalibration.

Participants were seated in front of a 13-inch Apple MacBook Pro, and a Tobii Pro Nano tracker was attached to the bottom of the screen using a magnet. It is worth noting that alternative configurations, such as placing the tracker in different locations, are possible. Additionally, the tracker was repositioned for each new participant, each time requiring a new calibration procedure. The laptop's camera captured images at a resolution of 1280×720 pixels. Each image was timestamped and matched with the closest measurement provided by the tracker. If the tracker didn't return gaze values for either eye, for example, due to blinking, the corresponding image was discarded.

A standard tracker calibration procedure was first conducted for each participant, using the Tobii Pro Eye Tracker Manager utility with a 9-point fixation pattern. The calibration was then evaluated by asking the participant to again fixate on the markers of the same 9-point fixation marker and measuring the average angular error (we used the validation code provided by Tobii[2]). We verified

[2]https://github.com/tobiipro/prosdk-addons-matlab

that, for each participant, the average angular error for both eyes was less than 1°.

After IR tracker calibration, each participant performed the camera-tracker calibration procedure detailed in the prior section. In addition, we conducted a second data collection exercise which is mainly used for the final evaluation of the quality of camera-tracker calibration. Participants were instructed to focus their gaze on a marker that sequentially appeared at 9 different positions on a regular calibration pattern. The purpose of this was to obtain images and associated gaze data for a representative range of gaze directions. The actual location of the marker on the screen (which is critical for fixation-type calibration procedures) was irrelevant to this study. For this data acquisition, participants were asked to find a comfortable position to fixate the points of the pattern. The distance between the participant's head and camera varied between 55 cm and 70 cm, and the measured gaze varied within a range of approximately 25° (pitch) and 30° (yaw). Approximately 150 images were acquired for each participant during this second data collection phase.

3.5 Results

3.5.1 Calibration Accuracy Evaluation

We considered two metrics for evaluating the accuracy of camera-tracker calibration. The first metric is *reprojection error* e_R. In this metric, each gaze origin location for either eye, which was computed by the tracker for the images used in calibration while the user looking at the camera, is transformed to the camera's reference frame using the parameters estimated by PnP. After this transformation, these gaze origin points are projected onto the camera's focal plane through the intrinsic parameter matrix K. If PnP was successful, the reprojection error (distance between this projected point and the pupil center location for that eye) should be small. To evaluate e_R, we consider the same data points used to compute $(R_t^c, \mathbf{T}^c)$, with users looking at the camera. We computed the reprojection error (in pixels) for each participant for all gaze origin points within the inlier set determined by the PnP algorithm. Note that the proportion of inliers across participants varied between 41% and 70%. The average reprojection error per participant is shown in Table 3.1. Note that this is consistently less than 1 pixel. Sample images from the inlier set are shown in Figure 3.2, showing good localization of the gaze origin projection within the corresponding eye's pupil image.

The second evaluation metric, known as *pupil inconsistency* e_P, was measured

Participant	e_R (pixels)		e_P (pixels)		e_G (mm)		e_V (degs)	
	Left	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**
P1	0.62	0.59	0.40	0.36	48.9	45.4	1.91	1.86
P2	0.57	0.69	0.49	0.16	82.2	81.2	1.81	2.01
P3	0.58	0.54	0.97	0.91	62.0	60.9	1.64	1.53
P4	0.72	0.75	0.42	0.94	77.6	76.0	2.68	2.62
P5	0.56	0.70	0.84	0.45	49.3	45.6	1.28	1.63

Table 3.1: Experimental results for the left and right eyes of our participants. e_R: reprojection error. e_P: pupil inconsistency error. e_G: distance between gaze point location computed with a 3-D face model [6] and with the IR tracker. e_V: the angular difference between the associated visual axes. Note that e_R values were computed on the images used for camera-tracker calibration (with the participants looking at the camera), while the other measurements are for the second data set, with the participants looking at different locations on the screen.

using the second data set where participants were looking at different points on the screen, away from the camera. For this evaluation, we initially computed the screen-camera calibration using the algorithm proposed by Rodrigues et al. [134]. Subsequently, for each measurement, we transformed both the gaze point and gaze origin obtained from the tracker into the camera's reference frame, using the obtained parameters $(R_t^c, \mathbf{T}^c)$ from PnP. The line connecting these two points represents the estimated visual axis, which is subsequently projected onto the

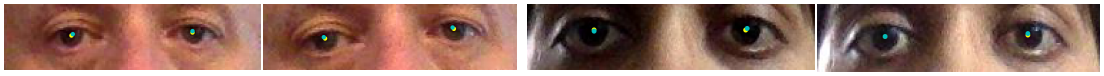


Figure 3.2: Sample images collected for camera-tracker calibration. Note that the participants are looking at the camera, with their heads moving in different locations between images. The pupil center location is shown as a yellow dot, and the projection of the gaze origin (computed by the IR tracker) is shown in aqua. These images belong to the set of inliers as determined by the PnP algorithm.

camera's focal plane using the intrinsic camera matrix K. Since the visual axis can be assumed to go through the eye's pupil, its projection should cross the pupil image. Accordingly, we define a measure of inconsistency as the distance between the projected visual axis and the pupil image. We measure this quantity as follows. First, we determine the radius r of the pupil image (assumed circular for simplicity's sake) by computing the foreshortening of the actual pupil radius R, which is a measurement provided by the Tobii Pro SDK: $r = f \cdot R/Z$, where f is the camera's focal length and Z is the distance between the pupil and the camera. Then, we measure the distance d between the projected visual axis and the pupil center (where the latter is computed using the algorithm of [74]) and define $e_P = \max(0, d - r)$. Note that $e_P = 0$ when the visual axis projection crosses the pupil image. The values of e_P averaged over all images in the second data set for each participant are shown in Table 3.1. Figure 3.3 shows examples with $e_P = 0$ (first three columns) and with $e_P > 0$ (fourth column).

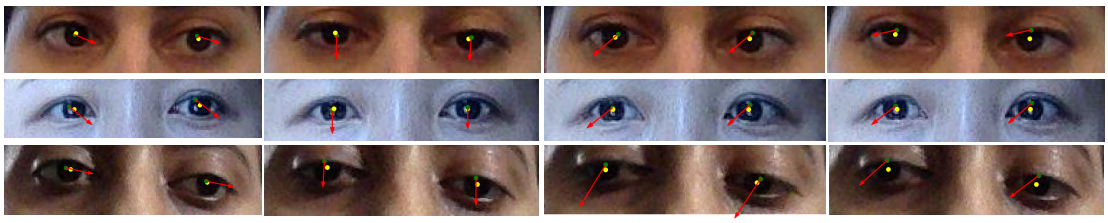


Figure 3.3: Sample images of participants looking at different locations on the screen. The pupil center is shown as a yellow dot, while the projection of the gaze origin (as computed by the IR tracker) is shown colored in green. The red arrow shows the projection of a 40 mm long segment, starting from the gaze origin and aligned along the visual axis. Note that in the first three columns, the projection of the visual axis crosses the pupil image ($e_P = 0$). For the images in the fourth column, $e_P > 0$.

3.5.2 Gaze Origin Computation: IR Tracker *vs.* Face Model

The proposed camera-tracker calibration algorithm enables the use of IR trackers to accurately measure the gaze origin (expressed in the camera's reference frame), which can be used to compute the visual axis as the line joining the gaze origin with the gaze point. It is interesting to compare the location of the gaze origin from the IR tracker with that computed using a face model. We used the 3-D face model from [6], as described in Section 3.2.1, to estimate the gaze origin for all images in our second data set.

Table 3.1 shows the average distance e_G between the gaze origin locations produced by the two procedures. The large discrepancy (up to 82 mm of distance) can be explained in large part by depth (Z) errors in the data from the face

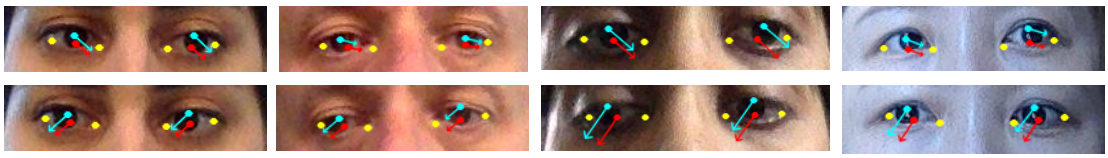


Figure 3.4: Sample images of participants looking at different locations on the screen, with a 40 mm segment of visual axis shown starting from the gaze origin, computed using a 3-D face model (shown with red color) and from the IR tracker (shown in aqua color), and joining the same gaze point (as computed by the IR tracker). The 2-D image projections of the eye corners are shown in yellow color.

model, possibly due to the imperfect fit of the 3-D model. The sample images in Figure 3.4 display, for each eye, the projection of the gaze origin estimated by the two methods, as well as the detected eye corners, which are employed to calculate the gaze origin using the face model (Section 3.2.1). It is worth noting that, at least for these examples, the gaze origin from the face model (red dots) appears to be clearly incorrect, as it seems to be located below the pupil. This would suggest an upward gaze, which is inconsistent with the fact that in these images, the participants were looking at a point below the camera. Figure 3.4 also shows the projection of the visual axes, obtained by joining the two different gaze origins with the same gaze point (from the IR tracker). The average angle between these two estimated visual axes, e_V, is shown for each participant in Table 3.1.

3.6 Summary

IR gaze trackers have the potential to be highly valuable for collecting extensive image datasets annotated with gaze information. Unlike traditional modalities requiring fixation of specific locations, IR trackers make it possible to measure gaze in dynamic settings, e.g., while reading text on the screen. In order to leverage the 3-D data produced by the IR tracker (and not just the gaze point on the screen), it is necessary to first find the relative pose of the tracker with respect to the camera. We have proposed a simple calibration procedure that asks the user to simply look at the camera from various head positions.

Our camera-tracker calibration enables the determination of the visual axis in the camera's reference frame, obtained by joining the "gaze origin" produced by the IR tracker with the gaze point on the screen, also computed by the tracker. We compare this quantity with that obtained by joining the gaze point with a different gaze origin location, computed through a 3-D face model, which is the standard procedure for obtaining gaze direction from fixation. Our results show that the average angular difference between these two axes can reach values as large 2°, suggesting that using a 3-D face model to estimate the gaze origin may introduce non-negligible errors.

There are clear limitations to the use of IR trackers for gaze annotation of images. The range of head locations and orientations from which gaze can be

computed accurately is constrained. While appropriate for interaction with a laptop or desktop computer, an IR tracker may not be used for applications that call for larger viewing distances or angles [123]. In addition, tracking accuracy is critical if this is to be used for ground-truth measurements, which means that only high-quality (and thus expensive) models (such as the Tobii Pro Nano used in this study) should be used for this purpose.

Chapter 4

Unsupervised domain adaptation for controllable gaze and head redirection

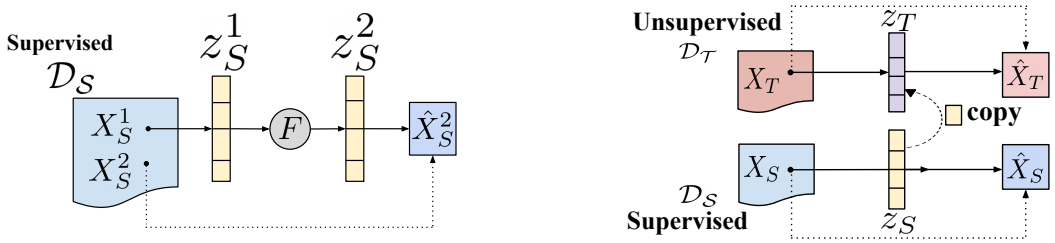
4.1 Introduction

Gaze behavior plays a pivotal role in the analysis of non-verbal cues and can provide support to various applications such as virtual reality [135, 136], human-computer interaction [137, 138], cognition [139, 140], and social sciences [141, 142]. Recent gaze estimation models rely on learning robust representations, requiring a time-consuming and expensive step of collecting a large amount of training data, especially when labels are continuous. Although various methods [115, 116, 143] have been proposed to circumvent the data need, to generalize in-the-wild real-world scenarios remains a challenge and is an open research problem.

Different gaze redirection methods [4, 96, 100] have been explored as an alternate solution for generating more labeled training data using generative adversarial

networks (GANs) [144] based frameworks. These generative methods require a pair of labeled images across both source and target domains to learn image-to-image translation; thus, these methods fail to generalize faithfully to new domains. Furthermore, various visual attributes are entangled during the generation process and cannot be manipulated independently to provide fine-grained control. Consequently, these methods have limited applicability, as in order for the generated data to be useful on downstream tasks, the variability of these visual attributes across the generated data plays a key role in their success. Few works [145, 146] on neural image generation attempt to manipulate individual visual attributes in-the-wild real-world scenarios; however, they are constrained by the availability of simulated data with pre-defined labeled attributes. The recent work [147] proposes contrastive regression loss and utilizes unsupervised domain adaption to improve gaze estimation performance.

In this chapter, we propose a novel domain adaptation framework for the task of controllable generation of eye gaze direction and head pose orientation in the target domain while not requiring any label information in the target domain. Our method learns to render such control by disentangling explicit factors (e.g., gaze and head orientations) from various implicit factors (e.g., appearance, illumination, shadows, etc.) using a labeled-rich source domain and an unlabeled target domain. Both disentanglement and domain adaptation are performed jointly, thus enabling



(a) Previous approaches [3, 4]

(b) Proposed method

Figure 4.1: **Comparison of existing and proposed method**. In (a), previous approaches [3, 4] assume conditional image-to-image translation ($X_S^1 \rightarrow X_S^2$) using a pair of labeled samples from a single domain $\mathcal{D}_{\mathcal{S}}$ and use a transforming function F in the latent space to ensure disentanglement. Here, $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$ represent the source and target domains. In (b), our method auto-encodes the images X_S, X_T from both domains into a common disentangled space using labels only from source, and transfers latent factors via a simple copy operation.

the transfer of learned knowledge from the source to the target domain. Since we use only unlabeled target-domain data to train our framework, we call it as *unsupervised domain adaptation* [148, 149].

Figure 4.1 illustrates the differences between the proposed method and previous approaches [3, 4]. Previous approaches use a pair of labeled samples (X_S^1, X_S^2) from the source domain $\mathcal{D}_{\mathcal{S}}$ to learn the conditional image-to-image translation while disentangling visual attributes using a transforming function F. In particular, Park et al. [3] provides control over only explicit factors while Zheng et al. [4] manipulate both explicit and implicit visual attributes. In contrast, our method can perform controllable generation without any input-output paired samples

and apply auto-encoding of images X_S and X_T from source $\mathcal{D}_\mathcal{S}$ and target $\mathcal{D}_\mathcal{T}$ domains into a common disentangled latent space. Concurrently, we adapt the latent representations from the two domains, thereby allowing the transfer of learned knowledge from the labeled source to the unlabeled target domain. Unlike previous approaches, the proposed method is less constrained by label information and can be seamlessly applied to a broader set of datasets and applications.

We train our method on GazeCapture [79] dataset and demonstrate its efficacy on two target domains: MPIIGaze [75] and Columbia [86] and obtain improved qualitative and quantitative results over state-of-the-art methods [3, 4]. Our experimental results exhibit a higher quality in preserving photo-realism of the generated images while faithfully rendering the desired gaze direction and head pose orientation.

The main contributions of this chapter can be summarized as follows:

1. We propose a domain adaptation framework for jointly learning disentanglement and domain adaptation in latent space, using labels only from the source domain.

2. Our method utilizes auto-encoding behavior to maintain implicit factors and enable fine-grained control over gaze and head pose directions and outperforms the baseline methods on various evaluation metrics.

3. We demonstrate the effectiveness of generated redirected images in improving

the downstream task performance on gaze and head pose estimation.

4.2 Related Work

For a comprehensive overview of gaze redirection methods, we refer the reader to Section 2.3.2. In this section, we offer a brief review of related work on disentangling representations, which is relevant to the discussions in this chapter.

The goal of learning disentangled representations is to model the variability of implicit and explicit factors prevalent in the data-generating process [150]. Fully supervised methods [151, 152, 153] exploit the semantic knowledge gained from the available annotations to learn these disentangled representations. On the other hand, unsupervised methods [154, 155] aim to learn the same behavior without relying on any labeled information. However, these methods offer limited flexibility in selecting a specific factor of variation and primarily focus on single-domain representation learning problems [156]. Unsupervised cross-domain disentangled representation learning methods [157, 158] exploit the advantage of domain-shared and domain-specific attributes in order to provide fine-grained control on the appearance and content of the image. For instance, synthetic data is utilized by a few recent works [145, 146] to control various visual attributes while relying on the pre-defined label information associated with the rendered image obtained through a graphics pipeline. On the other hand, Liu et al. [159] provide control over

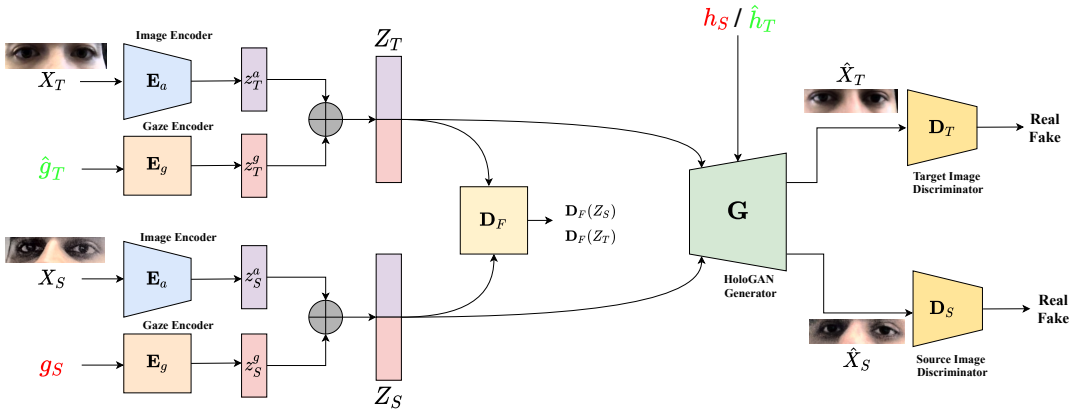


Figure 4.2: **Overview of CUDA-GHR.** $\mathbf{E}_a$ encodes the target domain image X_T to z_T^a, and the source domain image X_S to z_S^a while $\mathbf{E}_g$ encodes the target pseudo gaze label $\hat{g}_T$ and ground-truth source gaze label g_S to z_T^g and z_S^g, respectively. The overall image representations are formed as $Z_S = z_S^a \oplus z_S^g$ and $Z_T = z_T^a \oplus z_T^g$ (where, $\oplus$ is concatenate operation). These domain-specific encoded embeddings Z_T and Z_S are passed through a shared generator network $\mathbf{G}$ along with the corresponding head poses (pseudo head pose label $\hat{h}_T$ for the target domain, and ground-truth head pose label h_S for source domain). These embeddings are also passed through a feature domain discriminator $\mathbf{D}_F$. $\mathbf{D}_T$ and $\mathbf{D}_S$ represent two domain-specific image discriminators. The labels in red color are the ground-truth labels, while in green color are the generated pseudo-labels.

different image attributes using the images from both source and target domains and is trained in a semi-supervised setting. However, their approach only considers categorical labels and thus has limited applicability. In contrast, the method described in this chapter allows controllable manipulation of continuous-valued image attributes (such as gaze and head pose) in the cross-domain setting.

4.3 Proposed Method

Our goal is to learn a controller network $\mathbf{C}$ such that given an input image X_T and subset of explicit factors $\{e_i\}$ (e.g., gaze and head pose directions), it generates an image X_O satisfying the attributes described by $\{e_i\}$, i.e., $\mathbf{C} : (X_T, e_i) \rightarrow X_O$. To achieve this, we design a framework that learns to disentangle the latent space and manipulate each explicit factor independently. We start with the assumption that there are three factors of variations: 1) appearance-related such as illumination, shadows, person-specific, etc., which might or might not be explicitly labeled with the dataset, 2) eye gaze direction, and 3) head pose orientation. We train our network in an unsupervised domain adaptation setting by utilizing a fully labeled source domain and an unlabeled target domain considering distribution shift across datasets into account. Recall that we have the gaze and head pose labels only for the source domain. Therefore, we aim to disentangle and control these three factors of variations in the latent space and simultaneously transfer the learned behavior to the unsupervised target domain. We named our framework as CUDA-GHR.

4.3.1 Model

The overall architecture of the CUDA-GHR is shown in Figure 4.2. We denote S as the source domain and T as the target domain. Further, following the notations used in [3], we represent the appearance-related latent factor as z^a and

gaze latent factor as z^g.

The initial stage of our network consists of two encoders: (a) an image encoder $\mathbf{E}_a$ encodes the implicit (appearance-related) factors of an image X_i and outputs z_i^a such that $i \in \{S, T\}$, and (b) a separate MLP-based gaze encoder $\mathbf{E}_g$ encodes the input gaze g_i corresponding to the image X_i to a latent factor z_i^g. For the source domain, we use ground-truth gaze label g_S as input to $\mathbf{E}_g$ while for the unlabeled target domain, we input pseudo gaze labels $\hat{g}_T$ obtained from a pre-trained task network $\mathcal{T}$ that predicts gaze and head pose of an image. Note that $\mathcal{T}$ is trained only on source domain data. Thus, the overall embedding Z_i related to an image X_i can be formed by concatenating these two latent factors, i.e., $Z_i = z_i^a \oplus z_i^g$ (here $\oplus$ denotes concatenation). Further, Z_i and head pose label h_i are given as input to a decoder $\mathbf{G}$ based on the generator used in HoloGAN [160] as it allows the head pose to be separately controlled without any encoder. This generator $\mathbf{G}$ decodes the latent Z_i and head pose h_i to an output image given by $\hat{X}_i$ and is trained in an adversarial manner with the discriminator network $\mathbf{D}_i$. Note again that for labeled source images, we use ground-truth head pose label h_S while we take pseudo head pose label $\hat{h}_T$ produced by task network $\mathcal{T}$ for unlabeled target domain inputs. In addition, we use a feature domain discriminator $\mathbf{D}_F$ to ensure that the latent distributions of Z_S and Z_T are similar.

At inference time, the gaze and head pose directions are controlled by passing

an image from the target domain X_T through the encoder $\mathbf{E}_a$ and desired gaze direction g through $\mathbf{E}_g$, giving us $\mathbf{E}_a(X_T)$ and $\mathbf{E}_g(g)$ respectively. These two latent factors are concatenated and passed through the generator $\mathbf{G}$ along with the desired head pose h to generate an output image $\hat{X}_T^{g,h}$ with gaze g and head pose h, i.e.,

$$\hat{X}_T^{g,h} = \mathbf{G}(\mathbf{E}_a(X_T) \oplus \mathbf{E}_g(g),\ h) \tag{4.1}$$

Likewise, we can also control the individual factor of gaze (or head pose) by providing desired gaze (or head pose) and pseudo head pose (or gaze) label obtained from $\mathcal{T}$ to generate gaze redirected image given as

$$\hat{X}_T^{g} = \mathbf{G}(\mathbf{E}_a(X_T) \oplus \mathbf{E}_g(g),\ \hat{h}_T) \tag{4.2}$$

and head redirected image given as

$$\hat{X}_T^{h} = \mathbf{G}(\mathbf{E}_a(X_T) \oplus \mathbf{E}_g(\hat{g}_T),\ h) \tag{4.3}$$

4.3.2 Learning Objectives

The overall objective of our method is to learn a common factorized latent space for both the source and target domains, allowing for easy control of individual latent factors to manipulate target images. To ensure this, we train our framework using multiple objective functions, each of which is explained in detail below.

Reconstruction Loss. We apply pixel-wise L1 reconstruction loss between the generated image $\hat{X}_i$ and input image X_i to ensure the auto-encoding behavior.

$$\mathcal{L}_{\mathcal{R}}(\hat{X}_i, X_i) = \frac{1}{|X_i|}||\hat{X}_i - X_i||_1$$

Thus, the total reconstruction loss is defined as

$$\mathcal{L}_{recon} = \sum_{i \in \{S,T\}} \mathcal{L}_{\mathcal{R}}(\hat{X}_i, X_i)$$

Perceptual Loss. To ensure that our generated images perceptually match the input images, we apply the perceptual loss [161] which is defined as a mean-square loss between the activations of a pre-trained neural network applied between the generated image $\hat{X}_i$ and input image X_i. For this, we use VGG-16 [77] network trained on ImageNet [162].

$$\mathcal{L}_{\mathcal{P}}(\hat{X}_i, X_i) = \sum_{l=1}^{4} \frac{1}{|\psi_l(X_i)|} \, ||\psi_l(\hat{X}_i) - \psi_l(X_i)||_2$$

where ψ denotes VGG-16 network and l is the index of activation layer of ψ. Therefore, our overall perceptual loss becomes

$$\mathcal{L}_{perc} = \sum_{i \in \{S,T\}} \mathcal{L}_{\mathcal{P}}(\hat{X}_i, X_i)$$

Consistency Loss. To ensure disentangled behavior between implicit and explicit factors, we apply a consistency loss between the generated image $\hat{X}_i$ and input image X_i. For this, we use a pre-trained task network $\mathcal{T}$ which predicts the

pseudo-labels (gaze and head pose) for an image. The consistency loss consists of two terms: (a) *label consistency loss* is applied between pseudo-labels for input and the generated images to preserve the gaze and head pose information, and (b) *redirection consistency loss* guarantees to preserve the pseudo-labels for redirected images. For latter (b), we generate gaze and head redirected images using Equation 4.2 and 4.3 respectively, by applying gaze and head pose labels from source domain. We enforce the gaze prediction consistency between $\hat{X}_T^g$ and X_S, and head pose prediction consistency between $\hat{X}_T^g$ and X_T, i.e., $\mathcal{T}^g(\hat{X}_T^g) = \mathcal{T}^g(X_S)$ and $\mathcal{T}^h(\hat{X}_T^g) = \mathcal{T}^h(X_T)$. A similar argument holds for the head redirected image $\hat{X}_T^h$, i.e., $\mathcal{T}^g(\hat{X}_T^h) = \mathcal{T}^g(X_T)$ and $\mathcal{T}^h(\hat{X}_T^h) = \mathcal{T}^h(X_S)$. Here, $\mathcal{T}^g$ and $\mathcal{T}^h$ represent the gaze and head pose predicting layers of $\mathcal{T}$. The overall gaze consistency loss will become

$$\mathcal{L}_{gc} = \underbrace{\mathcal{L}_a(\mathcal{T}^g(\hat{X}_S), \mathcal{T}^g(X_S)) + \mathcal{L}_a(\mathcal{T}^g(\hat{X}_T), \mathcal{T}^g(X_T))}_{\textit{label consistency loss}} + \underbrace{\mathcal{L}_a(\mathcal{T}^g(\hat{X}_T^g), \mathcal{T}^g(X_S)) + \mathcal{L}_a(\mathcal{T}^g(\hat{X}_T^h), \mathcal{T}^g(X_T))}_{\textit{redirection consistency loss}}$$

Similarly, we can compute the head pose consistency loss $\mathcal{L}_{hc}$ as follows:

$$\mathcal{L}_{hc} = \underbrace{\mathcal{L}_a(\mathcal{T}^h(\hat{X}_S), \mathcal{T}^h(X_S)) + \mathcal{L}_a(\mathcal{T}^h(\hat{X}_T), \mathcal{T}^h(X_T))}_{\textit{label consistency loss}} + \underbrace{\mathcal{L}_a(\mathcal{T}^h(\hat{X}_T^g), \mathcal{T}^h(X_T)) + \mathcal{L}_a(\mathcal{T}^h(\hat{X}_T^h), \mathcal{T}^h(X_S))}_{\textit{redirection consistency loss}}$$

Here, $\mathcal{L}_a$ is defined as:

$$\mathcal{L}_a(\hat{\boldsymbol{u}}, \boldsymbol{u}) = \arccos\left(\frac{\hat{\boldsymbol{u}} \cdot \boldsymbol{u}}{||\hat{\boldsymbol{u}}|| \cdot ||\boldsymbol{u}||}\right)$$

Hence, total consistency loss becomes

$$\mathcal{L}_{consistency} = \mathcal{L}_{gc} + \mathcal{L}_{hc}$$

GAN Loss. To enforce photo-realistic output from the generator **G**, we apply the standard GAN loss [144] to image discriminator $\mathbf{D}_i$.

$$\mathcal{L}_{GAN_D}(\mathbf{D}_i, X_i, \hat{X}_i) = \log \mathbf{D}_i(X_i) + \log(1 - \mathbf{D}_i(\hat{X}_i))$$

$$\mathcal{L}_{GAN_G}(\mathbf{D}_i, \hat{X}_i) = \log \mathbf{D}_i(\hat{X}_i)$$

The final GAN loss is defined as

$$\mathcal{L}_{disc} = \sum_{i \in \{S,T\}} \mathcal{L}_{GAN_D}(\mathbf{D}_i, X_i, \hat{X}_i)$$

$$\mathcal{L}_{gen} = \sum_{i \in \{S,T\}} \mathcal{L}_{GAN_G}(\mathbf{D}_i, \hat{X}_i)$$

Feature Domain Adversarial Loss. We employ a latent domain discriminator network $\mathbf{D}_F$ and train it using the following domain adversarial loss [163] to push the distribution of Z_T closer to Z_S.

$$\mathcal{L}_{feat}(\mathbf{D}_F, Z_T, Z_S) = \log \mathbf{D}_F(Z_S) + \log(1 - \mathbf{D}_F(Z_T))$$

Overall Loss. Altogether, our final loss function for training encoders and generator network is

$$\mathcal{L}_{overall} = \lambda_R \mathcal{L}_{recon} + \lambda_P \mathcal{L}_{perc} + \lambda_C \mathcal{L}_{consistency} + \lambda_G \mathcal{L}_{gen} + \lambda_F \mathcal{L}_{feat}$$

Here, $\lambda_R, \lambda_P, \lambda_C, \lambda_G$, and λ_F represent the weights applied to each loss function.

4.4 Experiments

4.4.1 Data Pre-processing

We follow the same data pre-processing pipeline as done in Park et al. [3]. The pipeline consists of a normalization technique [127] initially introduced by Sugano et al. [124]. It is followed by face detection [128] and facial landmarks detection [129] modules for which open-source implementations are publicly available. The Surrey Face Model [6] is used as a reference 3D face model. Further details can be found in Park et al. [3]. To summarize, we use the public code[1] provided by Park et al. [3] to produce image patches of size 256×64 containing both eyes. The inputs gaze g and head pose h are 2-D pitch and yaw angles.

4.4.2 Architecture Details

Our framework CUDA-GHR. We use DenseNet architecture [164] to implement image encoder $\mathbf{E}_a$. The DenseNet is formed with a growth rate of 32, 4 dense blocks (each with four composite layers), and a compression factor of 1. We use instance normalization [165] and leaky ReLU activation function ($\alpha = 0.01$) for all layers in the network. We remove dropout and 1×1 convolution layers. The dimension of latent factor z^a is set to be equal to 16. Thus, to project CNN features to the latent features, we use global-average pooling and pass through a

[1]https://github.com/swook/faze_preprocess

Layer name	Activation	Output shape
Fully connected	LeakyReLU ($\alpha = 0.01$)	2
Fully connected	LeakyReLU ($\alpha = 0.01$)	2
Fully connected	LeakyReLU ($\alpha = 0.01$)	2
Fully connected	None	8

Table 4.1: Architecture of gaze encoder $\mathbf{E}_g$

fully-connected layer to output 16-dimensional feature from $\mathbf{E}_a$. The gaze encoder $\mathbf{E}_g$ is a MLP-based block whose architecture is shown in Table 4.1. The dimension of z^g is set as 8.

For the generator network $\mathbf{G}$, we use HoloGAN [160] architecture shown in Table 4.2. The latent vector z for each AdaIN [166] input is processed by a 1-layer MLP, and the rotation layer is the same as the one used in the original paper [160]. The latent domain discriminator $\mathbf{D}_F$ consists of 4 MLP layers as shown in Table 4.3. It takes the input of dimension 24 and gives 1-dimensional output. Both image discriminators $\mathbf{D}_T$ and $\mathbf{D}_S$ are PatchGAN [167] based networks as used in Zheng et al. [4]. The architecture of the discriminator is described in Table 4.4.

The task network $\mathcal{T}$ is a ResNet-50 [168] model with batch normalization [169] replaced by instance normalization [165] layers. It takes an input of 256×64 and gives a 4-dimensional output describing pitch and yaw angles for gaze and head directions. It is initialized with ImageNet [170] pre-trained weights and is fine-tuned on the GazeCapture training subset for around 190K iterations. The

Layer name	Kernel	Activation	Normalization	Output shape
Learned Constant Input	-	-	-	512×4×2×8
Upsampling	-	-	-	512×8×4×16
Conv3d	3×3×3	LeakyReLU	AdaIN	256×8× 4×16
Upsampling	-	-	-	256×16×8×32
Conv3d	3×3×3	LeakyReLU	AdaIN	128×16×8×32
Volume Rotation	-	-	-	128×16×8×32
Conv3d	3×3×3	LeakyReLU	-	64×16×8×32
Conv3d	3×3×3	LeakyReLU	-	64×16×8×32
Reshape	-	-	-	$(64 \cdot 16)$×8×32
Conv2d	1×1	LeakyReLU	-	512×8×32
Conv2d	4×4	LeakyReLU	AdaIN	256×8×32
Upsampling	-	-	-	256×16×32
Conv2d	4×4	LeakyReLU	AdaIN	64×16×64
Upsampling	-	-	-	64×32×128
Conv2d	4×4	LeakyReLU	AdaIN	32×32×128
Upsampling	-	-	-	32×64×256
Conv2d	4×4	Tanh	-	3×64×256

Table 4.2: Architecture of the generator network **G**

GazeCapture validation subset is used to select the best-performing model. The initial learning rate is 0.0016, decayed by a factor of 0.8 after about 34K iterations. Adam [171] optimizer is used for optimization with a weight decay coefficient of 10^{-4}. Note that $\mathcal{T}$ remains fixed during the training of our whole pipeline. The architecture of $\mathcal{T}$ is summarized in Table 4.5.

Layer name	Activation	Output shape
Fully connected	LeakyReLU ($\alpha = 0.01$)	24
Fully connected	LeakyReLU ($\alpha = 0.01$)	24
Fully connected	LeakyReLU ($\alpha = 0.01$)	24
Fully connected	None	1

Table 4.3: Architecture of latent domain discriminator $\mathbf{D}_F$

Layer name	Kernel, Stride, Padding	Activation	Normalization	Output shape
Conv2d	4×4, 2, 1	LeakyReLU ($\alpha = 0.2$)	-	64× 32×128
Conv2d	4×4, 2, 1	LeakyReLU ($\alpha = 0.2$)	InstanceNorm	128×16×64
Conv2d	4×4, 2, 1	LeakyReLU ($\alpha = 0.2$)	InstanceNorm	256×8×32
Conv2d	4×4, 1, 1	LeakyReLU ($\alpha = 0.2$)	InstanceNorm	512×7×31
Conv2d	4×4, 1, 1	-	-	1×6×30

Table 4.4: Architecture of the image discriminator networks $\mathbf{D}_T$ and $\mathbf{D}_S$.

4.4.3 Training Details

We train our framework in two settings: *GazeCapture→MPIIGaze*, trained with GazeCapture [79] as source domain and MPIIGaze [75] as target domain, and *GazeCapture→Columbia* is trained with Columbia [86] as the target domain. For GazeCapture, we use the training subset from the data split as labeled source domain data. From MPIIGaze and Columbia, we respectively choose the first 11 and 50 subjects as unlabeled target domain data for training. We call them as **'Seen'** subjects as our network sees them during training while remaining users fall into **'Unseen'** category. We evaluate our method on three test subsets: 'Unseen', 'Seen' and 'All'. 'All' includes both 'Seen' and 'Unseen' participants data.

Module/Layer name	Output shape
ResNet-50 layers with MaxPool stride=1	2048×1×1
Fully connected	4

Table 4.5: Architecture of the task network $\mathcal{T}$

Hyper-parameters. We use a batch size of 10 for both *GazeCapture*→*MPIIGaze* and *GazeCapture*→*Columbia* and are trained for 200K and 81K iterations, respectively. All network modules are optimized through Adam [171] optimizer with a weight decay coefficient of 10^{-4}. The initial learning rate is set to 0.0005 which is decayed by a factor of 0.8 after approximately 34K iterations. For *GazeCapture*→*MPIIGaze*, we restart the learning rate scheduler after around 160K iterations for better convergence. The weights of the objective function are set as $\lambda_R = 200$, $\lambda_P = 10$, $\lambda_C = 10$, $\lambda_G = 5$ and $\lambda_F = 5$.

4.4.4 Evaluation Metrics

We evaluate our framework using three evaluation metrics as previously adopted by [4]: perceptual similarity, redirection errors, and disentanglement errors.

Learned Perceptual Image Patch Similarity (LPIPS) [172] is used to measure the pairwise image similarity by calculating the distance in AlexNet [173] feature space.

Redirection Errors are computed as angular errors between the estimated

direction obtained from our task network $\mathcal{T}$ and the desired direction. It measures the accomplishment of the explicit factors, i.e., gaze and head directions in the image output.

Disentanglement Error measures the disentanglement of explicit factors like gaze and head pose. We evaluate $g \rightarrow h$, the effect of change in gaze direction on the head pose, and vice versa ($h \rightarrow g$). To compute $g \rightarrow h$, we first calculate the joint probability distribution function of the gaze direction values from the source domain and sample random gaze labels. We apply this gaze direction to the input image while keeping the head pose unchanged and measure the angular error between head pose predictions from task network $\mathcal{T}$ of the redirected image and the original reconstructed image. Similarly, we compute $h \rightarrow g$ by sampling random head pose orientations from the source labeled data.

4.5 Results

We adopt FAZE [3] and ST-ED [4] as our baseline methods. Both FAZE and ST-ED are based on transforming encoder-decoder architecture [98, 99] and apply known differences in gaze and head rotations to the embedding space for translating the input image to a redirected output image. FAZE inputs an image containing both eyes, which is the same as our method; thus, it is necessary to compare. We

use original implementation[2] and trained models provided by the FAZE authors for comparison. We re-implement the ST-ED on images containing both eyes for a fair comparison with our method using the public code[3] available. We use the same hyperparameters as provided by the original implementation. For the accurate comparison, we replaced *tanh* non-linearity with an identity function and removed a constant factor of 0.5π in all the modules. FAZE learns to control only explicit factors (gaze and head pose orientations), while ST-ED controls implicit factors, too. Note that for the ST-ED baseline, we compare only by altering explicit factors. Furthermore, we also compare CUDA-GHR to baseline ST-ED+PS, which is trained with source data GazeCapture and using pseudo-labels for the target dataset (MPIIGaze or Columbia). The pseudo-labels are obtained in the same manner as CUDA-GHR.

Quantitative Evaluation. Table 4.6 summarizes the quantitative evaluation of both our experiments *GazeCapture→MPIIGaze* and *GazeCapture→Columbia*. The left half of Table 4.6 shows evaluation on MPIIGaze test subsets {'Seen', 'Unseen', 'All'}, and we observe that our method outperforms the baselines (even ST-ED+PS) on all the evaluation metrics for each test subset. We get lower LPIPS (even on 'Unseen' users), indicating the generation of better quality images while achieving the desired gaze and head directions attested by lower gaze and head

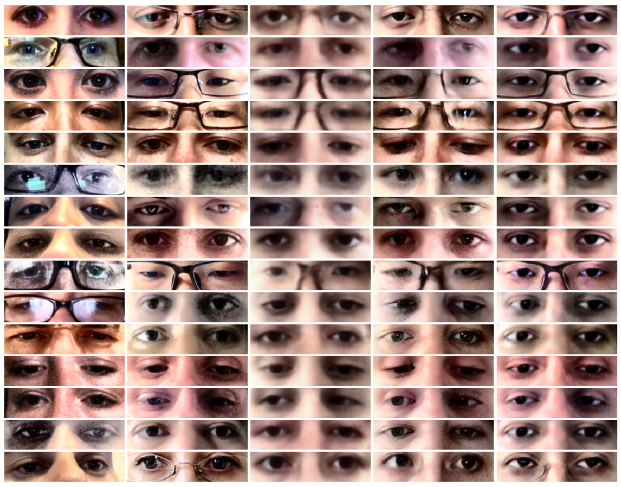
[2]https://github.com/NVlabs/few_shot_gaze
[3]https://github.com/zhengyuf/STED-gaze

		GazeCapture→MPIIGaze					*GazeCapture→Columbia*				
Test Set	Method	LPIPS ↓	Gaze Redir. ↓	Head Redir. ↓	$g \to h \downarrow$	$h \to g \downarrow$	LPIPS ↓	Gaze Redir. ↓	Head Redir. ↓	$g \to h \downarrow$	$h \to g \downarrow$
Unseen	FAZE	0.311	6.131	6.408	6.925	4.909	0.435	9.008	6.996	6.454	4.295
	ST-ED	0.274	2.355	1.605	1.349	2.455	0.265	2.283	1.651	1.364	2.190
	ST-ED+PS	0.266	2.864	1.576	1.472	2.346	0.266	2.117	1.437	**1.124**	2.356
	CUDA-GHR	**0.261**	**2.023**	**1.154**	**1.161**	**1.829**	**0.255**	**1.449**	**0.873**	1.209	**1.514**
Seen	FAZE	0.382	5.778	6.899	5.311	5.172	0.486	10.368	7.231	7.302	4.788
	ST-ED	0.315	2.405	1.669	1.209	2.341	0.319	2.484	1.616	1.343	2.456
	ST-ED+PS	0.288	2.269	1.888	1.179	2.229	0.299	2.071	1.536	1.088	2.330
	CUDA-GHR	**0.278**	**1.905**	**0.979**	**0.761**	**1.236**	**0.282**	**1.328**	**0.831**	**0.646**	**0.996**
All	FAZE	0.370	5.840	6.828	5.613	5.123	0.481	10.214	7.226	7.214	4.737
	ST-ED	0.307	2.392	1.660	1.232	2.359	0.314	2.473	1.618	1.350	2.435
	CUDA-GHR	**0.275**	**1.922**	**1.012**	**0.844**	**1.341**	**0.279**	**1.337**	**0.832**	**0.707**	**1.045**

Table 4.6: **Quantitative Evaluation.** Comparison of CUDA-GHR with the state-of-the-art methods [3, 4]. *GazeCapture→MPIIGaze* is evaluated on MPIIGaze subsets, and *GazeCapture→Columbia* is evaluated on Columbia subsets. All errors are in degrees (°) except LPIPS, and lower is better.

redirection errors. We also obtain better disentanglement errors, exhibiting that our method successfully controls each explicit factor individually. The improved performance on 'Unseen' users shows the superiority and generalizability of our method over baselines. We also notice improvements over the ST-ED+PS baseline, exhibiting that domain adaptation is essential to achieve better performance.

We show the evaluation of *GazeCapture→Columbia* experiment on the right half of Table 4.6. Note that due to the small size of the Columbia dataset, we initialize the model for this experiment with the previously trained weights on *GazeCapture→MPIIGaze* for better convergence. Recall that we do not use any labels from the target domain dataset in any experiment. As shown in Table 4.6,



(a) Gaze Redirected images

(b) Head Redirected images

Figure 4.3: Qualitative results for *GazeCapture→MPIIGaze* on the MPIIGaze dataset. 4.3a and 4.3b shows the gaze and head redirected images, respectively.

our method is consistently better than other baselines on all evaluation metrics, showing the generalizability of our framework on different domains and, thus, can be adapted to new datasets without the requirement of any labels.

(a) Gaze Redirected images

(b) Head Redirected images

Figure 4.4: Qualitative results for *GazeCapture→Columbia* on the Columbia dataset. 4.4a and 4.4b shows the gaze and head redirected images, respectively.

Qualitative Evaluation. We also report the qualitative comparison of generated images in Figure 4.3 and 4.4 using a model trained with *GazeCapture→MPIIGaze* and *GazeCapture→Columbia*, respectively. The results are shown respectively on

MPIIGaze and Columbia dataset images, which are the target domain datasets in these settings. As can be seen, our method produces better quality images while preserving the appearance information (e.g., skin color, eye shape) and faithfully manipulating the gaze and head pose directions when compared with FAZE [3] and ST-ED [4]. It is also worth noting that our method generates higher-quality images for people with glasses, e.g., row 3 in Figure 4.3a and row 2 in Figure 4.3b. These results are consistent with our findings in quantitative evaluation, thus showing that our method is more faithful in reproducing the desired gaze and head pose directions.

4.5.1 Ablation Study

We provide the following ablation study to understand the role of individual components of the objective function. In Table 4.7, we compare against the ablations of individual loss terms. The ablation on the perceptual loss is shown in the first row ($\lambda_P = 0$). The second row ($\lambda_C = 0$) represents when consistency loss is set to zero, while the third row ($\lambda_F = 0$) shows results when feature domain adversarial loss is not enforced during training. The fourth and fifth row shows an ablation on reconstruction ($\lambda_R = 0$) and GAN ($\lambda_G = 0$) loss, respectively. As can be seen, all of these loss terms are critical for the improvements in the performance. We see a substantial improvement by adding $\mathcal{L}_{consistency}$. The ablation study is

Ablation term	LPIPS ↓	Gaze Redir. ↓	Head Redir. ↓	$g \to h \downarrow$	$h \to g \downarrow$
$\lambda_P = 0$	0.307	6.450	0.922	0.655	1.334
$\lambda_C = 0$	0.326	15.183	3.412	**0.106**	11.616
$\lambda_F = 0$	0.281	4.791	**0.787**	0.636	**0.826**
$\lambda_R = 0$	0.304	4.958	0.911	0.463	0.876
$\lambda_G = 0$	0.309	11.130	0.942	0.355	0.868
Ours	**0.278**	**1.905**	0.979	0.761	1.236

Table 4.7: **Ablation Study:** An ablation study on different loss terms for *GazeCapture* → *MPIIGaze* on MPIIGaze 'Seen' subset. All errors are in degrees (°) except LPIPS, and lower is better.

performed for *GazeCapture* → *MPIIGaze* on the 'Seen' subset of MPIIGaze.

4.5.2 Controllability

Figure 4.5 shows the effectiveness of our method in controlling the gaze and head pose directions. We vary pitch and yaw angles from $-30°$ to $+30°$ for gaze and head redirections. We can see that our method faithfully renders the desired gaze direction (or head pose orientation) while retaining the head pose (or gaze direction), therefore exhibiting the efficacy of disentanglement. Furthermore, note that the range of yaw and pitch angles $[-30°, 30°]$ is the out-of-label distribution of the source dataset (GazeCapture), showing the extrapolation capability of CUDA-GHR in the generation process.

(a) Gaze redirected images with (pitch, yaw) $\in [-30°, 30°]$

(b) Head redirected images with (pitch, yaw) $\in [-30°, 30°]$

Figure 4.5: **Controllable Generation:** Illustration of controllable gaze and head redirection showing the effectiveness of disentanglement of various explicit factors.

4.5.3 Evaluation of Downstream Tasks

We also demonstrate the utility of generated images from our framework in improving the performance of the downstream gaze and head pose estimation task. For this, we conduct experiments for cross-subject estimation on both MPIIGaze and Columbia datasets. The primary objective of this experiment is to demonstrate that the generated "free" labeled data from our framework can effectively serve as a valuable resource for obtaining a well-performing pre-trained model, which can then be fine-tuned for the cross-subject estimation task. We compare it against three different initializations: random initialization, ImageNet initialization [170], and a pre-trained model obtained using ST-ED [4] generated images.

We generate around 10K samples per user from MPIIGaze dataset using *GazeCapture→MPIIGaze* trained generator and train a network similar to $\mathcal{T}$ as shown in Table 4.5. The initial learning rate is 0.0001 decayed by a factor of 0.5 after every 1500 iteration, and the pre-training is done for 10 epochs with a batch size of 64. Afterward, we fine-tune this network for 5 epochs with a batch size of 32 on the MPIIGaze dataset using leave-one-subject-out cross-validation for both gaze and head pose estimation, and we report the mean angular error. A similar method is followed for ST-ED generated images. We compare the errors obtained from four initialization methods: random, ImageNet, ST-ED, and CUDA-GHR. Analogously, we train gaze and head pose estimation models on generated images for Columbia data subjects (∼1.6K samples each) using *GazeCapture→Columbia* model and fine-tune the Columbia dataset using 4-fold cross-validation. The comparison of different initialization methods on two datasets is shown in Table 4.8.

It can be seen that the model trained with CUDA-GHR gives around 7% and 4% relative improvements over ST-ED initialization on Columbia and MPIIGaze, respectively, for the head pose estimation task. We also show results for the gaze estimation task in Table 4.8 giving a relative improvement of around 5.5% on the Columbia dataset while performing similarly to the ST-ED baseline on MPIIGaze. We hypothesize that this is because the gaze and head pose label distribution of GazeCapture is closer to MPIIGaze distribution than Columbia [29] and, thus,

Initialization Method	**Head Pose Estimation Errors↓**		**Gaze Estimation Errors↓**	
	Columbia	MPIIGaze	Columbia	MPIIGaze
Random	6.8 ± 1.2	6.7 ± 0.7	6.7 ± 0.7	6.7 ± 1.3
ImageNet	5.9 ± 1.3	5.7 ± 2.8	5.5 ± 0.1	5.7 ± 1.4
ST-ED	5.7 ± 1.1	5.1 ± 2.4	5.4 ± 0.4	**5.5 ± 1.3**
CUDA-GHR	**5.3 ± 1.1**	**4.9 ± 2.5**	**5.1 ± 0.4**	**5.5 ± 1.4**

Table 4.8: **Downstream Task Evaluation:** Comparison of mean angular errors (*mean* ± *std* in degrees) for various initialization methods on gaze and head pose estimation task. Lower is better.

performs closely for both ST-ED and CUDA-GHR. This indicates that domain adaptation is more advantageous for the Columbia dataset. Hence, it shows the effectiveness of our method over baselines when performing domain adaptation across datasets with significant distribution shifts.

4.6 Summary

This chapter presents an unsupervised domain adaptation framework trained using cross-domain datasets for gaze and head redirection tasks. The proposed method takes advantage of both the supervised source domain and the unsupervised target domain to learn the disentangled factors of variations. Experimental results demonstrate the effectiveness of our model in generating photo-realistic images in multiple domains while truly adapting the desired gaze direction and head pose

orientation. Because of removing the requirement of annotations in the target domain, the applicability of our work increases for new datasets where manual annotations are hard to collect. Our framework is relevant to various applications such as video conferencing, photo correction, and movie editing for redirecting gaze to establish eye contact with the viewer. It can also be extended to improve performances on the downstream task of gaze and head pose estimation.

Chapter 5

Self-supervised representation learning for gaze estimation

5.1 Introduction

In recent years, deep learning has shown promising results for gaze estimation [75, 78, 79]. This success is largely due to the availability of extensive annotated datasets. Consequently, to be effective, these datasets need to encompass a diverse array of gaze directions, appearances, and head poses, a process that is both labor-intensive and time-consuming. Moreover, obtaining accurate gaze annotations is a challenging task [174]. This difficulty adds to the challenge of creating large, representative datasets in the field. Consequently, methods that enable effective training with a limited number of gaze annotations are extremely valuable.

Self-supervised learning (SSL) has gained tremendous success over the past few years and emerged as a powerful tool for reducing over-reliance on human annotations [175, 176, 177]. Following a generally accepted paradigm, we consider

a pre-training stage that requires no labels, followed by a fine-tuning stage using a relatively small number of labeled samples. SSL is an effective approach for pre-training, where semantically meaningful representations are learned that can be seamlessly adapted during fine-tuning stage [178, 179, 180]. Specifically, a good pre-training would ensure that the embeddings for images associated with the same gaze direction are neighbors in the feature space, regardless of other non-relevant factors such as appearance. Arguably, this could accelerate the job of fine-tuning, possibly reducing the number of required labeled samples.

In this chapter, for SSL pre-training, we focus on *contrastive representation learning* (CRL), which aims to map "positive" pair samples to embeddings that are close to each other while mapping "negative" pairs apart from each other [181]. A popular approach is to generate pairs by applying two different transformations (or augmentations) to an input image forming a positive pair, and different images forming negative pairs. This method encourages invariance in representations w.r.t. similar types of transformations, where these transformations are assumed to model "nuisance" effects.

However, obtaining the necessary and sufficient set of positive and negative pairs remains a non-trivial and unanswered challenge for a given task. This chapter attempts to answer this question for gaze estimation. Recent CRL-based methods encourage the representations to be invariant to any image transformation, many

of which are not suitable for gaze estimation. For example, geometry-based image transformations (such as rotation) will change the gaze direction. In contrast, it is beneficial to have invariance to appearance, a person's identity, background, etc.

We propose *Gaze Contrastive Learning* (or *GazeCLR*) framework – a simple CRL-based unsupervised pre-training approach for gaze estimation, i.e., a pre-training method requiring no gaze label data. In detail, our approach relies on *invariance* to image transforms (e.g., color jitter) that do not alter gaze direction and *equivariance* to camera viewpoint, which requires additional information of multi-view geometry, i.e., images of the same person should be obtained at the same time by two or more cameras from different locations.

For learning *equivariance*, we leverage the fact that in *a common reference system*, two or more synchronous images of the same person from different camera viewpoints are associated with the same gaze direction. The knowledge of the relative pose of each camera to the *common reference system* provides the relation of gaze directions defined in the respective camera space. In other words, gaze direction has an equivariant relationship to camera viewpoints, as shown in next paragraph. We claim that the requirement of using multiple cameras may be less onerous than obtaining gaze annotations for each image.

Given a specific timestamp, let two samples from different camera viewpoints with gaze directions be g_{v_1} and g_{v_2} in their original respective camera reference

frame, then the relation between these two gaze directions through their relative camera pose (i.e., $R_{C_1}^{C_2}$), can be given as follows:

$$g_{v_2} = R_{C_1}^{C_2} g_{v_1}$$

$$g_{v_2} = R_S^{C_2} R_{C_1}^S g_{v_1}$$

$$(R_S^{C_2})^{-1} g_{v_2} = (R_S^{C_2})^{-1} R_S^{C_2} R_{C_1}^S g_{v_1}$$

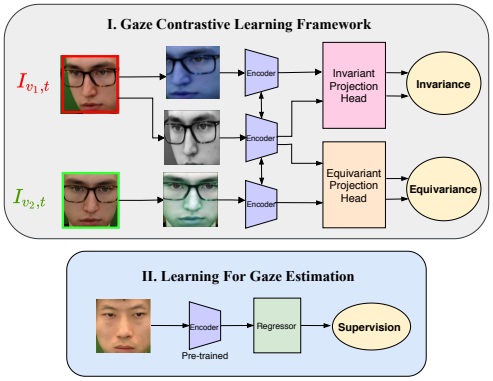
$$R_{C_2}^S g_{v_2} = R_{C_1}^S g_{v_1}$$

$$\bar{g}_{v_1} = \bar{g}_{v_2}$$

where $R_S^{C_i}$ is relative pose between camera view i and common reference frame S. Therefore, the equation $R_{C_2}^S g_{v_2} = R_{C_1}^S g_{v_1}$ shows the equivariance relationship between gaze directions in multi-view geometry, which is replicated for the corresponding embeddings, a key idea for *GazeCLR*.

We use an existing multi-view gaze dataset EVE [89], which provides video sequences captured from four calibrated and synchronized cameras and contains gaze annotations, which are obtained using a gaze tracking device [182]. We neglect labels during pre-training and use them only for fine-tuning and evaluation. Observe that the relative camera pose information available with the EVE dataset is used *only* during the pre-training stage. Figure 5.1 presents an overview of the proposed idea.

To evaluate the *GazeCLR*, we perform self-supervised pre-training using the EVE dataset and transfer the learned representations for the gaze estimation task in



(a) Two-stages of our framework

(b) Formation of positive pairs

Figure 5.1: **Overall idea.** (a) The proposed two-stage learning framework for gaze estimation. Stage I shows the Gaze Contrastive Learning (*GazeCLR*) framework trained using only unlabeled data and learns both *invariance* and *equivariance* properties. In Stage II, the pre-trained encoder is employed for gaze estimation tasks with small labeled data. (b) Two images (shown in **red** and **green**) captured at the same time with different camera views are used to create both invariant and equivariant positive pairs.

various evaluation settings. We demonstrate the effectiveness of representations by showing that the proposed method achieves superior performance on both within-dataset and cross-dataset (such as MPIIGaze [78] and Columbia [86]) evaluations by using only a small number of labeled samples for fine-tuning.

The major contributions of this chapter are summarized as follows:

1. We propose a simple contrastive learning method for gaze estimation that relies on the observation that gaze direction is invariant under selected appearance transformations and equivariant to any two camera viewpoints.

2. We also argue to learn equivariant representations by taking advantage of the multi-view data that can be seamlessly collected using multiple cameras.

3. Our empirical evaluations show that *GazeCLR* yields improvements for various settings of gaze estimation and is competitive with existing supervised [3] and unsupervised state-of-the-art gaze representation learning methods [7, 8].

5.2 Related Work

We refer the reader to Section 2.2.2.3 for an in-depth review of appearance-based gaze estimation, Section 2.3.3 for literature on representation learning for gaze estimation, and Section 2.3.5 which addresses person-specific gaze estimation using few labeled samples. In this section, we will provide a brief overview of related work on self-supervised learning, focusing on aspects that are relevant to the discussions in this chapter.

The goal of self-supervised representation learning is to learn good visual representations from a large collection of unlabeled images. Earlier works in SSL [183, 184, 185, 186] used pretext tasks to learn generalizable semantic representations. Some of the recent works [175, 176, 177, 187, 188, 189, 190] have shown great success on several vision tasks, e.g., image classification [178, 191], object detection [179], semantic segmentation [180], and pose estimation [192]. The work by Spurr et al. [193] extends SSL to hand pose estimation through geometric

equivariance representations. Tian et al. [194] propose to use more than two views to learn invariant representations through contrastive learning.

5.3 Proposed Method

5.3.1 Gaze Contrastive Learning (*GazeCLR*) Framework

GazeCLR is a framework to train an *encoder* that learns embeddings to induce the desired set of invariance and equivariance for the gaze estimation task. As stated earlier, the key intuition of *GazeCLR* is to enforce invariance using selected appearance transformations (e.g., color jitter) and equivariance using synchronous images of the same person captured from multiple camera viewpoints. Similar to previous SSL approaches [176, 193], we rely on the normalized temperature-scaled cross-entropy loss (NT-Xent)[176] to encourage invariance or equivariance by maximizing the agreement between positive pairs and disagreement between the negative pairs. In particular, we devise two variants of NT-Xent loss, namely, L^I for invariance and L^E for equivariance, discussed in further paragraphs.

The *GazeCLR* framework has three sub-modules: a CNN-based encoder and two projection heads based on MLP layers, as illustrated in Figure 6.2. The output of the encoder branches out into different projection head depending on the type of input positive pair. To abide by the invariance for gaze direction, we consider

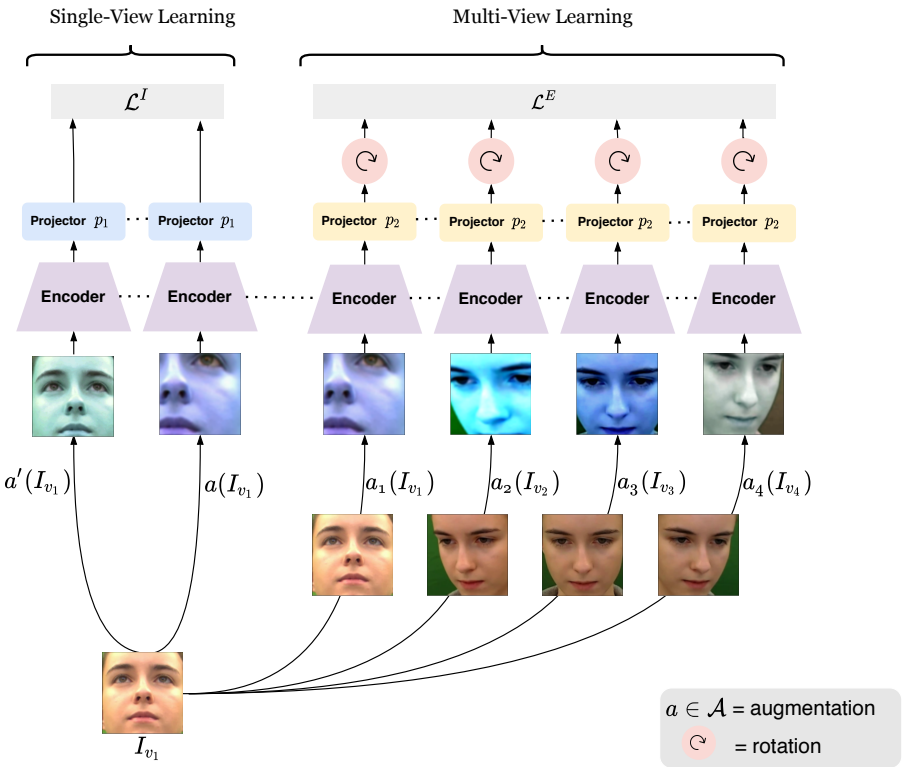


Figure 5.2: **Method schematic.** For synchronous view frames $\{I_{v_i}\}_{i=1}^4$, the above figure illustrates invariant and equivariant positive pairs anchored only for view v_1. The left branch shows *single-view* learning (L^I), and the right branch illustrates *multi-view* learning using four views (L^E). All images (after augmentation, $a \in \mathcal{A}$) are passed through a shared CNN encoder network, followed by MLP projectors (either p_1 or p_2) depending on the type of input positive pair. The embeddings for multi-view learning are further multiplied by an appropriate rotation matrix.

augmentations based on only appearance transformations denoted as $\mathcal{A}$.

Let $\{I_{v_i,t}\}_{i=1}^{|V|}$ be the synchronous frames for timestamp t coming from different camera views (i.e., $\{v_i\}_{i=1}^{|V|}$), then we create the following positive pairs:

1. *Single-view positive pairs:* We apply two randomly sampled augmentations

from $\mathcal{A}$ to create a single-view positive pair. Specifically, for any image $I_{v_i,t}$, at a given timestamp t and view v_i, we sample two augmentations a and a' from $\mathcal{A}$ and then $(a(I_{v_i,t}), a'(I_{v_i,t}))$ forms a positive pair to learn invariance. The left branch of Figure 6.2 shows one such positive pair for view v_1.

2. *Multi-view positive pairs:* We consider all unique pairs of camera viewpoints from the same timestamp t and apply random augmentations from $\mathcal{A}$, i.e., $\{(a_i(I_{v_i,t}), a_j(I_{v_j,t})) \mid i, j \in \{1, \ldots, |V|\} \mid i \neq j\}$. The corresponding outputs from the encoder are passed through projection head p_2 and multiplied by an appropriate rotation matrix to learn equivariance.

Next, to construct negative pairs, we do not sample them explicitly but use all other samples in the mini-batch as negative examples, similar to Chen et al. [176]. The exact formulation of both loss functions L^I and L^E is described below. For brevity, we omit t from $I_{v_i,t}$ and augmentation a in the following subsections.

5.3.1.1 Single-View Learning

The goal of single-view learning is to induce invariance amongst representations under various appearance transformations. Let $v_i \in V$ be any view and $b \in [1, \ldots, B]$ be the batch index. Given a batch size of B, we apply two augmentations to each sample in the batch, yielding $2B$ augmented images, and for each sample, we have one positive pair and $(2B - 1)$ negative pairs stemming from remaining

samples in the batch. Our encoder E extracts representations for all $2B$ augmented images, which are further mapped by projection head $p_1(\cdot)$ yielding embeddings $\{(z^b_{v_i}, z'^b_{v_i})\}_{b=1}^{B}$. With above notations, for any view v_i, the proposed invariance loss function L^I associated with a positive pair $(z^b_{v_i}, z'^b_{v_i})$ can be given as follows:

$$L^I(z^b_{v_i}, z'^b_{v_i}) = -\,\texttt{log}\,\frac{\texttt{sim}(z^b_{v_i}, z'^b_{v_i})}{\sum_{l=1}^{B} 1_{[l\neq b]}\,\texttt{sim}(z^b_{v_i}, z^l_{v_i}) + \sum_{l=1}^{B}\texttt{sim}(z^b_{v_i}, z'^l_{v_i})} \tag{5.1}$$

where, $z^b_{v_i} = p_1(E(I^b_{v_i}))$, $z'^b_{v_i} = p_1(E(I'^b_{v_i}))$, $\texttt{sim}(r,s) = \texttt{exp}\left(\frac{1}{\tau}\frac{r^T s}{||r||\cdot||s||}\right)$, $1_{[l\neq b]}$ is an indicator function and τ is the temperature coefficient parameter. It is worth noting that to minimize the loss in Eq. 5.1, it must hold that $z^b_{v_i}$ and $z'^b_{v_i}$ needs to be closer, which aligns with our goal of learning invariance to appearance transformations. One challenge, however, is the risk of collapse (e.g., the network could simply learn each person's identity). To avoid this, we create mini-batches such that all samples in a batch are taken from a single participant.

5.3.1.2 Multi-View Learning

We encourage equivariance in the gaze representations to different camera viewpoints through multi-view learning. To do so, we transform embeddings to a common reference system, chosen as the *screen reference system* used during the EVE data collection. Let $\{R^S_{C_{v_i}}\}$ be the rotation matrix relating the camera viewpoint v_i with the screen reference system.

For each sample $I^b_{v_i}$ in a batch of size B, the positive pair is given as $(I^b_{v_i}, I^b_{v_j})$

for two distinct camera viewpoints $(v_i, v_j)_{i \neq j}$. All images for viewpoints v_i and v_j are first augmented then passed through encoder E and the projector head $p_2(\cdot)$ which gives embeddings $\hat{z}^b_{v_i}, \hat{z}^b_{v_j} \in R^{3 \times d'}$. These embeddings are further multiplied by corresponding rotation matrices $R^S_{C_{v_i}}$ to project embeddings in the common (screen) reference system. We denote embeddings after rotation as $\{(\bar{z}^b_{v_i}, \bar{z}^b_{v_j})\}_{b=1}^B$ such that $\bar{z}^b_{v_i} = R^S_{C_{v_i}} \hat{z}^b_{v_i}$. Therefore, for a batch of size B, our equivariant loss L^E associated with the positive pair $(\bar{z}^b_{v_i}, \bar{z}^b_{v_j})$ is as follows:

$$L^E(\bar{z}^b_{v_i}, \bar{z}^b_{v_j}) = -\,\texttt{log}\,\frac{\texttt{sim}(\bar{z}^b_{v_i}, \bar{z}^b_{v_j})}{\sum_{l=1}^B 1_{[l \neq b]}\,\texttt{sim}(\bar{z}^b_{v_i}, \bar{z}^l_{v_i}) + \sum_{l=1}^B \texttt{sim}(\bar{z}^b_{v_i}, \bar{z}^l_{v_j})} \tag{5.2}$$

Overall loss function. Given $|V|$ camera viewpoints, we apply both L^I and L^E loss functions to each view. Thus, our overall objective function for a batch size of B becomes

$$L^O = \frac{1}{2B} \sum_{i=1}^{|V|} \sum_{b=1}^{B} \left(L^I(z^b_{v_i}, z'^b_{v_i}) + L^I(z'^b_{v_i}, z^b_{v_i}) + \sum_{j=1, j \neq i}^{|V|} L^E(\bar{z}^b_{v_i}, \bar{z}^b_{v_j}) \right) \tag{5.3}$$

5.3.2 Learning For Gaze Estimation

After pre-training, the encoder learned by the *GazeCLR* framework is used for the task of gaze estimation and fine-tuned on a small labeled dataset. To this end, we remove both projection heads p_1 and p_2, and replace them with MLP regressor layers to predict 3D gaze direction. For training MLP regressor, we use

the supervised loss function given as

$$L^{ang} = \frac{180}{\pi}\arccos\left(\frac{\boldsymbol{g}\cdot\hat{\boldsymbol{g}}}{||\boldsymbol{g}||\cdot||\hat{\boldsymbol{g}}||}\right) \tag{5.4}$$

where $\boldsymbol{g}$ and $\hat{\boldsymbol{g}}$ are the ground-truth and predicted gaze directions, respectively.

5.4 Experiments

5.4.1 Setup

We train our *GazeCLR* framework on the EVE [89] dataset, which has videos collected in a constrained indoor setting with four different synchronized and calibrated camera views. It has approximately 12 million frames collected from 54 participants with natural eye movements. Following the splits considered by Park et al. [89], there are 40 subjects in training and 6 subjects in the validation set. We discard the data of test subjects due to the non-availability of labels. We use training subjects for the pre-training stage, *without* using any gaze annotations. For the gaze estimation stage, we evaluate on the data of validation subjects to report the performance. We use all four camera views (i.e., $|V| = 4$) as well as the information about the relative pose between camera and screen (R_C^S) provided with the EVE dataset. Note that our framework can be extended to more number of camera views ($|V| > 4$) using ETH-XGaze [87] dataset.

Data pre-processing. We use face images available in the EVE dataset, obtained after applying a data-normalization procedure [124, 127]. The normalization pipeline transforms the gaze annotation to a normalized camera space through a rotation matrix M. Note that we post-multiply R_C^S with M^{-1} as R_C^S is defined w.r.t. the original camera reference frame, i.e., $\bar{z}_v = R_{C_v}^S (M)^{-1} \hat{z}_v$.

Training details. *GazeCLR* is trained using SGD optimizer with initial learning rate = 0.03, momentum = 0.9, and cosine annealing [195] for the learning rate decay. We use a single 1080 GeForce GTX GPU for training, with a batch size of 128, and train for 50K iterations. Our mini-batch is made up of samples from a single participant. The temperature coefficient τ is set to 0.1. For the augmentation transformations $\mathcal{A}$, we apply random spatial cropping and resizing, gaussian blur, color perturbation ($p = 0.8$) on brightness, contrast, saturation and hue, grayscale conversion ($p = 0.2$), and auto-contrast ($p = 0.5$).

All experiments use ResNet-18 [168] as the encoder network and take the output from the average pooling layer. The encoder is trained from scratch. Following Chen et al. [176], both projection heads $p_1(\cdot)$ and $p_2(\cdot)$ are two-layer MLP networks with ReLU non-linearity. The output dimensions for the first and second layers are 512 and 180, respectively. The input image size is 128×128.

We train the *GazeCLR* framework in two different settings: (i) *GazeCLR (Equiv)*: where we only consider equivariance through the loss function L^E and

(ii) *GazeCLR (Inv+Equiv)*: where we consider both invariance and equivariance with equal weights using the overall objective L^O. We present the performance of both training setups in all the considered experimental settings. Observe that, *GazeCLR (Inv)* trained with only L^I loss function is equivalent to SimCLR [176] baseline method.

5.4.2 Baselines

We compare our approach with six following baselines: (i) *w/o Pre-training*, i.e., an encoder is initialized using random weights, (ii) the vanilla *Autoencoder*, which has an encoder network that consists of the same encoder layers as *GazeCLR* and five DenseNet [164] deconvolution blocks as decoder, and is trained with L2 loss, (iii) *Novel View Synthesis* [192] framework is trained on our dataset using the same architecture as the auto-encoder, (iv) BYOL [177], (v) SimCLR [176] and (vi) *Fully-Supervised* is a ResNet-18 model trained on the whole EVE training data and represents possibly an upper bound for the performance of *GazeCLR*. For SimCLR and BYOL, we use the same augmentation set as in our proposed method. The following paragraphs discuss the implementation details of *Autoencoder* and *Novel View Synthesis* baseline approaches.

Autoencoder. We use the same encoder layers as the *GazeCLR* framework for a fair comparison. The decoder is implemented using DenseNet [164] architecture by

replacing convolutional layers with deconvolutional layers of stride 1. The average pooling layer of transition layers is replaced by 3×3 deconvolutions (with stride 2). The decoder consists of 5 dense blocks, where each block has 4 composite layers with a growth rate of 32. The compression factor is set to 1.0. All layers are implemented using instance normalization [165] and leaky ReLU activation functions (with $\alpha = 0.01$). We use SGD optimizer with momentum 0.9, weight decay 5×10^{-4}, and initial learning rate is 0.003 (which is decayed using cosine annealing scheduler [195]). The batch size is 24, and the model is trained for 200K iterations. For inference, we remove decoder layers and use the encoder only for gaze estimation tasks.

Novel View Synthesis [192]. This approach was originally proposed for a 3D human pose estimation task and aimed to learn novel view synthesis, where separate representations for the body's 3D geometry ($\mathbf{L}^{\text{3D}}$), appearance ($\mathbf{L}^{\text{app}}$), and background ($\mathbf{B}$) are trained. For a fair comparison, we train a novel view synthesis framework on our dataset using the same encoder architecture as in the *GazeCLR* framework. The decoder layers are the same as that of the autoencoder baseline. The dimension of appearance-based code ($\mathbf{L}^{\text{app}}$) is 32 and of 3D geometry code ($\mathbf{L}^{\text{3D}}$) is 480. We ignore the background factor ($\mathbf{B}$) in our implementation, as the EVE dataset has the same background across all images. The whole framework is trained using SGD optimizer with learning rate = 0.03, momentum = 0.9, weight

decay $= 5 \times 10^{-4}$, and cosine annealing for learning rate decay. The training is done for 200K iterations, with a batch size of 16. At each iteration, we randomly sample two views from the EVE dataset and generate one view image from other view images similar to Rhodin et al. [192]. The trained encoder is then adapted for the gaze estimation, similar to other baselines.

5.5 Results

5.5.1 Within-dataset Evaluation

For within-dataset evaluation, we perform pre-training on the training split of the EVE dataset without using labels. Then, we adapt the pre-trained encoder for gaze estimation on a small subset of labeled data. Precisely, we took five training subjects out of 40 (which form around 10% samples out of the whole EVE dataset) for the supervised gaze estimation stage and called it "MiniEVE". We validate on fixed subject data chosen from training subjects and report the final performance for validation subjects.

Table 5.1 shows the mean angular errors (in degrees) obtained for different pre-training baselines and the proposed *GazeCLR* method. To this end, we freeze the pre-trained encoder and simply train an MLP regressor using the "MiniEVE" dataset. Note that, for two baselines, Autoencoder and BYOL, we fine-tune the

Method	Pre-Train Data	Task Data	Frozen	MAE ↓ (degrees)
w/o Pre-training	EVE	MiniEVE	✗	8.47
Autoencoder	EVE	MiniEVE	✗	6.91
Novel View Synthesis [192]	EVE	MiniEVE	✓	6.79
BYOL [177]	EVE	MiniEVE	✗	8.35
SIMCLR [176]	EVE	MiniEVE	✓	6.57
GazeCLR (Equiv)	EVE	MiniEVE	✓	**4.83**
GazeCLR (Inv+Equiv)	EVE	MiniEVE	✓	<u>4.92</u>
Fully-Supervised	-	EVE	✗	4.15

Table 5.1: **Within-dataset Evaluation.** We report the mean angular errors (MAE) in degrees for within-dataset evaluation for gaze estimation. The "EVE" shows the whole EVE data while "MiniEVE" indicates a small subset of data. The Frozen column is ✓ if the pre-trained encoder is frozen, otherwise fine-tuned ✗. The best performing method is shown in **bold** and second best is <u>underlined</u>.

whole end-to-end framework along with the encoder as otherwise, they fail to converge when only their representations are used. We indicate this behavior in Table 5.1, using the *Frozen* column as ✓ if encoder is frozen otherwise as ✗.

We observe that our method *GazeCLR* outperforms other pre-training baseline methods by only training an MLP regressor on the small amount of labeled data ("MiniEVE" is $\sim 10\%$ of whole data). Specifically, it can be seen that the performance achieved from *GazeCLR* helps in closing the gap with the fully supervised baseline. Our method *GazeCLR (Inv+Equiv)* shows a relative improvement of

25.1% compared to the popular contrastive learning method SimCLR. Additionally, *GazeCLR (Equiv)* shows a boost of 26.4% relative improvement over the SimCLR approach, suggesting that equivariant representations are very effective for the gaze estimation task. We hypothesize that since we utilize similar augmentation strategies for creating both single-view and multi-view positive pairs, *GazeCLR (Equiv)* performs almost comparable to *GazeCLR (Inv+Equiv)*.

5.5.2 Transfer Learning/Cross-dataset Evaluation

We perform a cross-dataset evaluation using a few-shot personalized gaze estimation to further demonstrate the cross-data generalization capabilities of the learned representations. For this, we use *Linear Layer Training (LLT)* and *Finetuning (FT)* protocols, as discussed below. We evaluate *GazeCLR* representations on two domain datasets different from pre-training data: MPIIGaze [75] and Columbia [86]. MPIIGaze is a challenging dataset that has higher inter-subject variations. We use the standard evaluation subset MPIIFaceGaze [78], containing around 37667 images captured from 15 subjects. The Columbia dataset consists of 5880 images collected from 56 subjects and has large head pose variations.

Linear Layer Training (LLT). In the LLT protocol, we freeze the trained encoder and learn a linear regressor on the target dataset. For this experiment, we investigate under a few-shot setting where we sample a few calibration samples

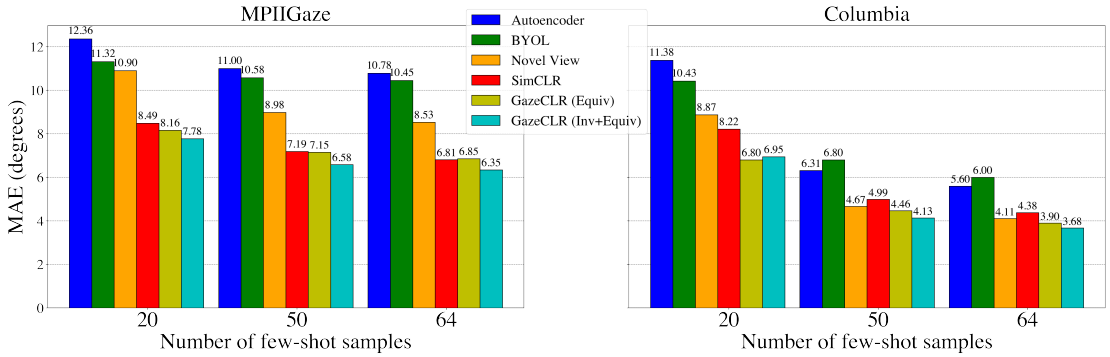


Figure 5.3: **Transfer Learning Evaluation (LLT).** Performance evaluation using *Linear Layer Training* protocol for both MPIIGaze and Columbia dataset under different few-shot settings. Each bar is computed by averaging over 10 runs.

from the test subject for adaptation and evaluate the remaining samples of the same test subject.

Figure 5.3 shows the mean angular errors for LLT protocol on 20-shot, 50-shot, and 64-shot gaze estimation. We first extract the gaze representations of a few calibration samples for each subject and learn a linear model on top of these representations. We evaluate the trained model on the remaining samples of the subject. We repeat above 10 times for each subject on both datasets and report mean angular errors for the same in Figure 5.3.

Observe that both proposed *GazeCLR* variants outperform all other baselines in all few-shot settings for both datasets. Moreover, *GazeCLR(Equiv)* gives a relative improvement of around 17.2% over SimCLR with only 20 calibration samples for Columbia. We hypothesize that this behavior is due to high head-pose variations

within Columbia, and it suggests that: a) learning equivariance over multi-views is beneficial for the GazeCLR framework in improving performance, and b) *GazeCLR* representations are relatively more generalizable for cross-domain datasets than other baselines.

Finetuning (FT). Next, we evaluate the transferable capability of learned representations obtained from *GazeCLR* framework using *Finetuning (FT)* protocol. Here, we fine-tune the entire network (including the encoder) in an end-to-end manner on the target dataset using a few calibration samples from the test subject and evaluate the remaining samples.

In Table 5.2, we present the results for FT on MPIIGaze and Columbia, where we fine-tune the whole end-to-end network. For this experiment, we adopt architecture from Chen and Shi [5], where a subject-dependent bias term is learned along with an end-to-end network. 4-fold and leave-one-out (15-fold) evaluation protocols are used for Columbia and MPIIGaze, respectively.

Unlike Chen and Shi [5], our input is a full-face image, and the backbone is a pre-trained encoder. We take a few calibration samples for each subject during inference and estimate the subject-dependent bias term. We evaluate performance on the remaining samples and repeat this calibration for 10 runs for each subject. Table 5.2 provides the mean and standard deviation of angular errors over 10 runs. We compare the performance of our method with other baselines for various

	MPIIGaze						
Method	1	3	5	9	15	50	64
w/o Pre-training [5]	$5.57^{\pm 1.60}$	$4.65^{\pm 0.71}$	$4.40^{\pm 0.40}$	$4.22^{\pm 0.27}$	$4.13^{\pm 0.17}$	$4.00^{\pm 0.04}$	$4.00^{\pm 0.04}$
Autoencoder	$5.65^{\pm 1.60}$	$4.69^{\pm 0.76}$	$4.42^{\pm 0.45}$	$4.16^{\pm 0.21}$	$4.10^{\pm 0.16}$	$3.97^{\pm 0.05}$	$3.96^{\pm 0.04}$
Novel View Synthesis [192]	$5.53^{\pm 1.32}$	$4.75^{\pm 0.63}$	$4.46^{\pm 0.40}$	$4.27^{\pm 0.25}$	$4.17^{\pm 0.15}$	$4.06^{\pm 0.04}$	$4.06^{\pm 0.04}$
BYOL [177]	$5.71^{\pm 1.63}$	$4.71^{\pm 0.66}$	$4.35^{\pm 0.31}$	$4.22^{\pm 0.21}$	$4.11^{\pm 0.15}$	$4.01^{\pm 0.05}$	$4.00^{\pm 0.04}$
SIMCLR [176]	$4.87^{\pm 1.51}$	$3.93^{\pm 0.54}$	$3.74^{\pm 0.35}$	$3.57^{\pm 0.24}$	$3.47^{\pm 0.12}$	$3.39^{\pm 0.04}$	$3.38^{\pm 0.03}$
GazeCLR (Equiv)	$\mathbf{4.70}^{\pm 1.49}$	$\mathbf{3.77}^{\pm 0.51}$	$\mathbf{3.51}^{\pm 0.32}$	$\mathbf{3.39}^{\pm 0.18}$	$\mathbf{3.33}^{\pm 0.11}$	$\mathbf{3.25}^{\pm 0.03}$	$\mathbf{3.24}^{\pm 0.02}$
GazeCLR (Inv+Equiv)	$4.72^{\pm 1.33}$	$3.93^{\pm 0.54}$	$3.68^{\pm 0.34}$	$3.54^{\pm 0.19}$	$3.44^{\pm 0.11}$	$3.37^{\pm 0.03}$	$3.35^{\pm 0.03}$
	Columbia						
w/o Pre-training [5]	$6.96^{\pm 0.55}$	$5.73^{\pm 0.20}$	$5.38^{\pm 0.14}$	$5.23^{\pm 0.09}$	$5.13^{\pm 0.05}$	$5.04^{\pm 0.08}$	$5.00^{\pm 0.09}$
Autoencoder	$7.00^{\pm 0.57}$	$5.79^{\pm 0.18}$	$5.49^{\pm 0.15}$	$5.24^{\pm 0.07}$	$5.15^{\pm 0.04}$	$5.03^{\pm 0.08}$	$5.03^{\pm 0.07}$
Novel View Synthesis [192]	$7.38^{\pm 0.60}$	$6.05^{\pm 0.22}$	$5.78^{\pm 0.14}$	$5.51^{\pm 0.05}$	$5.43^{\pm 0.06}$	$5.33^{\pm 0.06}$	$5.27^{\pm 0.08}$
BYOL [177]	$6.09^{\pm 0.41}$	$4.97^{\pm 0.22}$	$4.70^{\pm 0.13}$	$4.55^{\pm 0.09}$	$4.43^{\pm 0.04}$	$4.35^{\pm 0.05}$	$4.34^{\pm 0.06}$
SIMCLR [176]	$4.36^{\pm 0.20}$	$3.67^{\pm 0.13}$	$3.44^{\pm 0.07}$	$3.34^{\pm 0.05}$	$3.27^{\pm 0.04}$	$3.21^{\pm 0.04}$	$3.19^{\pm 0.05}$
GazeCLR (Equiv)	$\mathbf{4.34}^{\pm 0.25}$	$\mathbf{3.60}^{\pm 0.12}$	$\mathbf{3.42}^{\pm 0.09}$	$\mathbf{3.30}^{\pm 0.04}$	$\mathbf{3.26}^{\pm 0.02}$	$\mathbf{3.17}^{\pm 0.04}$	$\mathbf{3.17}^{\pm 0.02}$
GazeCLR (Inv+Equiv)	$4.54^{\pm 0.24}$	$3.75^{\pm 0.12}$	$3.59^{\pm 0.08}$	$3.45^{\pm 0.05}$	$3.39^{\pm 0.03}$	$3.31^{\pm 0.04}$	$3.31^{\pm 0.04}$

Table 5.2: **Transfer Learning Evaluation (Finetuning).** Comparison of various baselines for the *Finetuning* experimental protocol on multiple few-shot settings for both MPIIGaze and Columbia. We fine-tune the whole end-to-end network and utilize a few calibration samples during test time. The errors are computed from 10 runs and reported as ($mean^{\pm std}$).

few-shot settings. Results demonstrate that our method consistently outperforms all other pre-training baselines, including Chen and Shi [5] (w/o Pre-training) for all few-shot settings. This indicates the improved generalization capability of our learned representations, particularly on the MPIIGaze dataset. Also, our method is either superior or competitive with other baselines on the Columbia dataset.

5.5.3 Comparison with state-of-the-art methods

We further compare *GazeCLR* with existing state-of-the-art unsupervised [7, 8] and supervised [3] gaze representation learning methods. For a fair comparison, we adopt the same evaluation protocols as used by these baseline methods and compare the *GazeCLR* performance against their performance.

***GazeCLR* vs. Unsupervised Pre-training [7, 8].** We follow the same evaluation protocol as [7]. 5-fold and leave-one-out (15-fold) evaluations are used for the Columbia and MPIIGaze datasets, respectively. In each fold, we freeze the *GazeCLR* encoder and extract representations for randomly selected 50 samples with annotations and learn a simple MLP-based gaze estimator using these representations. We repeat the performance evaluation 10 times and report mean angular errors in Table 5.3. Note that previous methods [7, 8] exploit left and right eye patches to get SSL signal, whereas our approach relies on face patches obtained from multiple camera viewpoints.

In Table 5.3, we compare against the best-performing models of Yu and Odobez [7] and Sun et al. [8], for the 50-shot gaze estimation. Notice that our method outperforms baselines with absolute improvements of 2° and 0.9° on MPIIGaze and Columbia, respectively. It is worth emphasizing that our method is pre-trained on a different dataset than both evaluation datasets, unlike baseline approaches. Again, it illustrates the strength of our approach in learning semantically meaningful

Method	Pre-Train Data	MPIIGaze	Columbia
Yu and Odobez [7]†	Columbia	-	8.9
Sun et al. [8]	MPIIGaze	8.5	-
Sun et al. [8]	Columbia	-	7.0
***GazeCLR* (Equiv)**	EVE	7.0	**6.1**
***GazeCLR* (Inv+Equiv)**	EVE	**6.5**	6.6

Table 5.3: Comparison of *GazeCLR* with other unsupervised gaze representation learning methods [7, 8] for 50-shot gaze estimation. † denotes the method that uses additional head pose information. The metric reported is mean angular errors averaged over 10 runs (in degrees).

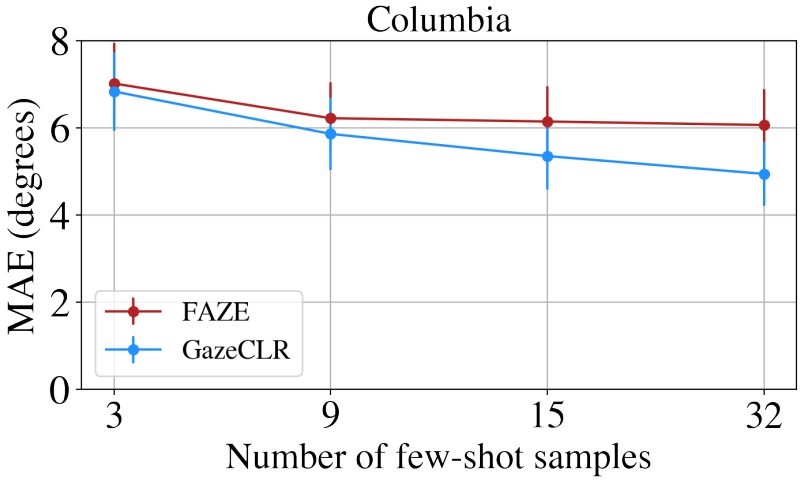


Figure 5.4: ***GazeCLR* vs FAZE [3].** Comparison of *GazeCLR* with supervised pre-training baseline (FAZE) for various few-shot settings on the Columbia dataset. The plot shows mean angular error (MAE, in degrees) and standard error bars *versus* number of few-shot samples, reported after 10 runs.

representations for generalizable to other domains. Moreover, note that Yu and Odobez [7] use additional head-pose information, unlike our method.

***GazeCLR* vs. Supervised Pre-training [3].** We evaluate the effectiveness of *GazeCLR* representations using the MAML framework [110], similar to FAZE [3]. For both *GazeCLR* and FAZE, we train a MAML-based gaze estimator on the representations for subjects from the GazeCapture [79] dataset. Then, we adapt the gaze estimator model to each test subject of Columbia with k calibration samples and test on the remaining samples. Figure 5.4 depicts the performance comparison of *GazeCLR* with FAZE [3] for four different values of k. It can be seen that our method consistently outperforms supervised pre-training baseline FAZE for all values of k. Notably, our framework uses *zero* labeled information to obtain gaze representations, unlike FAZE, which is pre-trained using $\sim 2M$ labeled samples from the GazeCapture dataset.

5.5.4 Ablation Studies

Increasing number of views improves pre-training. In Table 5.4, we demonstrate the effect of increasing the number of views used in the pre-training stage of *GazeCLR*. For this ablation study, we conducted an experiment for cross-dataset under LLT (similar to Figure 5.3) and within-dataset (similar to Table 5.1) settings, shown in Table 5.4(a) and Table 5.4(b) respectively. For 2 views, we considered the center and right cameras, and for 3 views, the left camera is included. For the LLT setting, the difference in *GazeCLR* performance for 2/3 views and

Dataset	# of views	$k = 20$	$k = 50$	$k = 64$
MPIIGaze	2	8.94	7.59	7.25
Columbia	2	7.63	4.58	4.02
MPIIGaze	3	8.38	7.09	6.78
Columbia	3	7.20	4.45	3.88
MPIIGaze	4	8.16	7.15	6.85
Columbia	4	6.80	4.46	3.90

(a) LLT Cross-dataset evaluation

# of views	MAE (degrees)
2	7.72
3	7.06
4	4.83

(b) Within-dataset evaluation

Table 5.4: **Ablation on the increasing number of views.** Within-dataset and cross-dataset (LLT) evaluation with the increasing number of views used for the pre-training stage of *GazeCLR* on both MPIIGaze and Columbia. The ablation study is performed for *GazeCLR(Equiv)* method, and the evaluation metric is a mean angular error (MAE) in degrees, averaged over 10 runs.

all 4 views is relatively higher, especially with a smaller number of shots. This shows that for smaller k, more views are helpful for *GazeCLR*. Similarly, for within-dataset, *GazeCLR* performance deteriorates with 2/3 views compared to 4.

More data, better pre-training. In Table 5.5(a), we study the impact of the amount of unlabeled data used for the pre-training stage of *GazeCLR* framework. We observe that the representations learned by *GazeCLR* benefit from more training data and help in generalizing across different domain datasets.

Larger batch size is useful. Next, we vary the batch size to analyze its effect on pre-training, for which results are shown in Table 5.5(b). We notice that the larger batch size considerably impacts the quality of representations and improves the

Pre-Train Data	MPIIGaze	Columbia
MiniEVE	11.25	9.63
EVE	**8.16**	**6.80**

(a) Varying amount of pre-training data

Batch size	MPIIGaze	Columbia
32	12.21	12.83
128	**8.16**	**6.80**

(b) Varying batch-size for pre-training

Table 5.5: **Ablation Study.** 20-shot *linear layer training* for the cross-data gaze estimation on MPIIGaze and Columbia for two different ablation settings. Ablations are performed for the *GazeCLR(Equiv)* method, and the evaluation metric is a mean angular error (MAE) in degrees.

performance significantly. This observation is consistent with previously observed findings in the self-supervised learning literature [175, 176].

Task Data	Batch Type	MAE (degrees)
MiniEVE	Single	**4.83**
MiniEVE	Multiple	23.58

Table 5.6: **Ablation Study for mini-batch containing single *vs.* multiple participants.** Within-dataset evaluation under two different types of batches created for the *GazeCLR(Equiv)* method and evaluation metric is mean angular error (MAE) in degrees.

Mini-batch of single *vs.* multiple participants. In Table 5.6, we experiment with creating batches from single and multiple subject samples and compare them under within-dataset evaluation. We observe that the performance on the gaze estimation task with multiple subject samples was close to the performance of

random weights. We hypothesize that this is because, in batches with different subjects, negative pairs are easy to classify, given the subject's identity. Therefore, the network has no incentive to focus on gaze information over subject identity.

Varying amount of data for fine-tuning. Here, we investigate how performance varies with respect to the amount of data available for finetuning. We evaluate for the within-dataset gaze estimation using LLT protocol, starting from 10% of EVE training dataset and gradually increasing to 100%. We compare *GazeCLR*(*Equiv*) and *GazeCLR*(*Inv*+*Equiv*) against "w/o Pre-training" baseline with random initialization, as shown in the Figure 5.5. *GazeCLR* outperforms the baseline in all training set sizes. It is worth noting that the *GazeCLR* approach only requires 20% of training data to match the performance of the "w/o Pre-training" baseline with 100%. Furthermore, notice that the gap between the performance of *GazeCLR* and baseline decreases as the training dataset size increases, showing that *GazeCLR* is effective for training with a few samples.

5.5.5 Visualization of Gaze Representations

To further investigate the quality of learned representations, we project the gaze representations into 2-dimensions using t-SNE [196] algorithm as shown in Figure 5.6. In Fig 5.6(a), we compute 2D visualization of equivariant representations obtained after applying rotation matrices, i.e., $\bar{z}$. Projections in Fig 5.6(a)

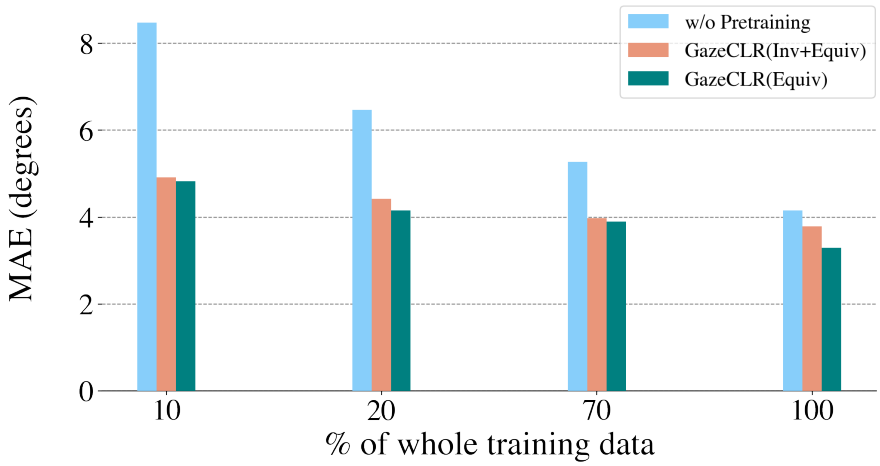
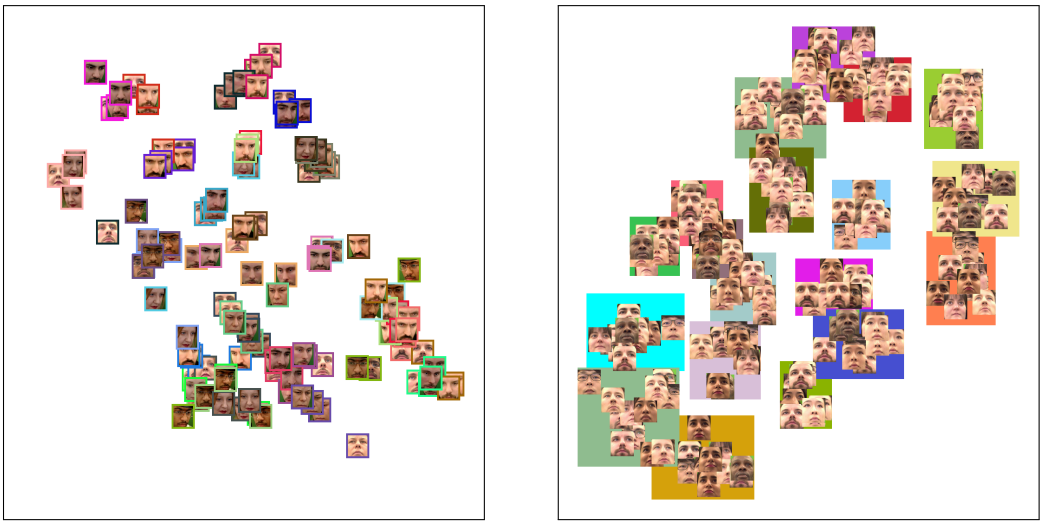


Figure 5.5: Comparison of the gaze estimation performance for within-dataset using LLT protocol, versus different % of the labeled training data.

clearly demonstrate that gaze direction is invariant to the viewpoint, as images at the same timestamp from different views are mapped closer (shown with the same color border). In Fig 5.6(b), we apply the t-SNE algorithm on gaze representations obtained at the output of the encoder network, i.e., $z = \mathrm{E}(\cdot)$, for images from single camera viewpoint. Projections corresponding to roughly similar gaze directions are naturally clustered and highlighted with different background colors. Also, we observe clear patterns in the learned feature space where images within close vicinity are invariant to the subject's identity, showing invariance towards appearances.

We further qualitatively analyze the relationship between learned gaze representations and the ground-truth 2D Point-of-Gaze (PoG). For this, we project gaze representations to 2-D space using t-SNE [196] algorithm and normalize them



(a) Representations after applying rotation matrices, $\bar{z}$

(b) Representation obtained from the output of encoder, $z = \mathrm{E}(\cdot)$

Figure 5.6: **t-SNE visualization.** Qualitative visualization of gaze representations in 2-dimensional space using the t-SNE algorithm. (a) shows the visualization of projection embeddings for multi-view images obtained after applying rotation matrices, i.e., $\bar{z}$. The images with the same timestamp for all four views are highlighted using the same border color. (b) depicts representations for the output of the encoder network, i.e., $z = \mathrm{E}(\cdot)$ obtained for images from a single camera viewpoint. Best viewed in color and after zooming.

between 0 and 1. Next, we plot Euclidean distance between 2D t-SNE projections and the normalized 2D PoG (dividing by the width and height of the screen), as shown in Figure 5.7. The black line is for the $y = x$ equation. Notice that data is scattered symmetrically around $y = x$, exhibiting a strong correlation (correlation coefficient = 0.623) between gaze representations and ground-truth PoG.

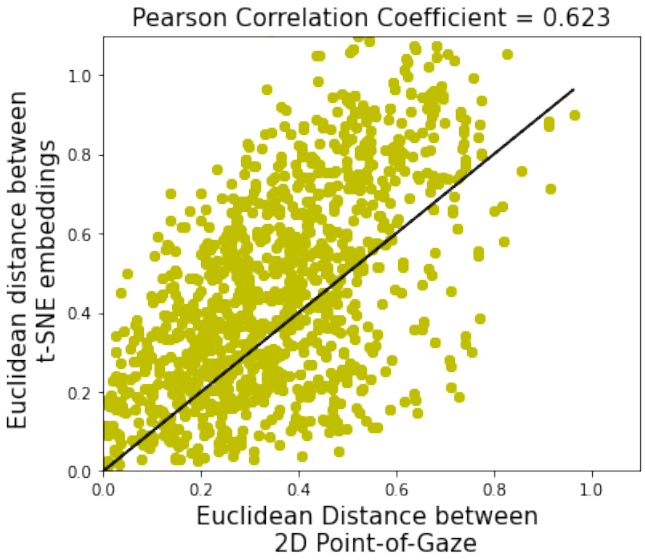


Figure 5.7: Scatter plot between Euclidean distance of normalized 2D PoG and 2D t-SNE projections of gaze representations. The black line is for $y = x$.

5.6 Summary

In this chapter, we presented *GazeCLR*, a contrastive learning framework for gaze representations using multi-view camera images. Our framework induces invariance and equivariance properties simultaneously in the learned representations and is effective for gaze estimation tasks in various settings. Furthermore, we showed that *GazeCLR* representations have the potential to be effective across different domain datasets using only a few calibration samples. *GazeCLR* is a general framework for equivariant representation learning and thus can be explored in the future for other geometry-based applications such as head pose estimation.

Chapter 6

Spatial-temporal attention and Gaussian processes personalization for video gaze estimation

6.1 Introduction

The human gaze is an essential cue for conveying people's intent, making it promising for real-world applications such as human-robot interaction [197, 198], AR/VR [199, 200], and saliency detection [201, 202]. Despite the primary research emphasis on gaze estimation from images, the potential benefits of understanding the temporal dynamics of eye movements for video gaze estimation have been relatively overlooked. Constructing an accurate video-based gaze estimation model requires addressing the unique challenges inherent to videos. These include the evolution of eye movements throughout the video, correlations between gaze directions in successive frames, the predominance of a static background in most pixels, and variations due to individual-specific traits [3, 109, 203]. This chapter

responds to these challenges by aiming to develop an accurate gaze estimation technique for videos using deep networks.

Realizing the potential of spatial and motion cues in videos, prior research has utilized residual frames and optical flows for several other vision tasks [204, 205, 206]. Specifically, these methods integrate RGB and residual frames as different input streams, requiring larger models with higher inference time and memory requirements [207, 208, 209]. Similarly, 3D convolutional neural networks (CNNs) can also capture spatiotemporal information from videos, but they require many model parameters [210, 211, 212, 213, 214, 215]. In addition, it is non-trivial to transfer knowledge from pre-trained 3D CNNs to new video tasks, as most pre-trained models rely on large 2D image datasets such as the ImageNet dataset [170]. Despite the critical role of detecting spatial and motion cues in videos, there is a strong need to design efficient attention-based approaches for video-related tasks, including video gaze estimation.

In this chapter, we draw inspiration from the *change captioning* task to develop an approach for video gaze estimation. The change captioning task requires describing the changes between a pair of before and after images, expressed through a natural language sentence [216, 217, 218]. Both change captioning and gaze estimation tasks require differentiating irrelevant distractors, such as background movement and facial expression changes, from the relevant ones. Specifically, change

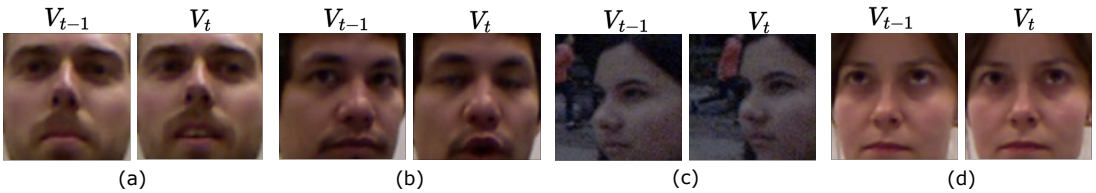


Figure 6.1: The figure illustrates a range of irrelevant factors for video gaze estimation, also referred to as distractors: (a) and (b) depict alterations in facial expression, (c) highlight background movement, and (d) represent a scenario without any distractors. These examples show the importance of accurately distinguishing between spatial changes due to eye movements and irrelevant distractors for the video gaze estimation task.

captioning focuses on recognizing object movements, whereas gaze estimation concentrates on detecting eye movements. Similar to prior works [216, 217], our approach utilizes a spatial attention mechanism to focus on gaze-relevant information while minimizing the impact of distractors. For example, Figure 6.1 illustrates various distractors that may obfuscate gaze information in videos.

We introduce ***S**patio-**T**emporal **A**ttention for **G**aze **E**stimation (STAGE)*, a deep learning model for video gaze estimation. STAGE utilizes spatial changes in consecutive frames to integrate motion cues via a Spatial Attention Module (SAM) and captures global dynamics with a Temporal Sequence Model (TSM). The SAM module focuses on gaze-relevant information by applying local spatial attention between consecutive frames and effectively suppresses irrelevant distractors. Meanwhile, the TSM considers global dynamic movements across the

temporal dimension, enabling enhanced prediction of gaze direction sequences. STAGE adeptly encodes motion information through the attention modules with fewer parameters than existing approaches like 3D CNNs [210] or two-branch networks [207], thus offering a more feasible solution for real-world applications.

To enhance the accuracy of gaze estimation models, previous studies have suggested personalization to address significant variability in individual-specific traits, such as eye geometry and appearance [3, 109, 219]. Concretely, this is done by training a person-agnostic gaze model on a large labeled dataset and then fine-tuning it for individual users with a small set of labeled data. Consistent with this approach, we integrate Gaussian processes (GPs) [220], known for their effectiveness in low-data scenarios, to personalize the STAGE for individual users.

We use GPs to learn an additive bias correction and personalize the gaze estimate of the general STAGE model with just a few labeled samples. GPs enable the estimation of personalized 3D gaze directions and provide uncertainty measurements in interval form. These intervals represent a range of possible gaze directions instead of a single vector, making our approach more suitable for practical applications, such as monitoring attention on screens [221, 222]. To evaluate the efficacy of the proposed STAGE model and personalization using GPs, we use three publicly available video gaze datasets: EYEDIAP [2], Gaze360 [88] and EVE [89].

To summarize, the key contributions of this chapter are outlined as follows:

- We introduce STAGE, a novel model for video gaze estimation. STAGE leverages an attention mechanism that is sensitive to spatial changes in sequential frames, effectively extracting gaze-relevant details from videos. This facilitates gaze prediction along the temporal axis for videos.
- We propose a sample-efficient approach to personalize the STAGE model, aiming to learn a bias correction model for gaze prediction using pre-trained Gaussian processes [223].
- Our approach either surpasses or matches to the state-of-the-art performance on three publicly available datasets for video gaze estimation. In particular, we obtain state-of-the-art results on the Gaze360 dataset in both cross-data and within-data experimental settings.

6.2 Proposed Method

The main goal of video gaze estimation is to learn a deep network f defined as $f : V \mapsto G$ that maps a sequence of video frames $V \in \mathbb{R}^{n \times h_0 \times w_0 \times 3}$ to a sequence of gaze directions $G \in \mathbb{R}^{n \times 2}$, where n is the number of frames and h_0 and w_0 are height and width of each frame, respectively. The output gaze sequence G possesses pitch and yaw angles, which correspond to each frame in V.

The proposed STAGE model employs three modules for setting up the deep network f. Firstly, a ResNet-based CNN model receives the input video and extracts feature maps for all the frames. Then, in the following module of the STAGE model, we process feature maps using a *Spatial Attention Module* (SAM) to focus on the spatial motion information between consecutive frames followed by a *Temporal Sequence Model* (TSM) to learn temporal dynamics using past frame embeddings. Next, the gaze prediction layer (GPL) maps the features from the output of the TSM block to a sequence of gaze directions defined in terms of yaw and pitch angles. Figure 6.2 shows the schematic of the STAGE and its modules.

6.2.1 Spatial Attention Module (SAM)

Recall that SAM is aimed to distinguish gaze-relevant motion by analyzing differences between consecutive frames, focusing on crucial cues like eye or head movements for gaze estimation while filtering out irrelevant distractions like facial expressions or background movements. It aims to prioritize relevant video changes, particularly eye movements, and disregard non-essential ones.

First, we convert each frame in the video sequence V to features $X = [X_1, X_2, \ldots, X_n] \in \mathbb{R}^{n \times h \times w \times k}$, using the ResNet-based CNN model, where w, h, and k are the width, height, and the number of channels of the feature maps extracted by ResNet. The next step is to pass each consecutive feature pair (X_{t-1}, X_t)

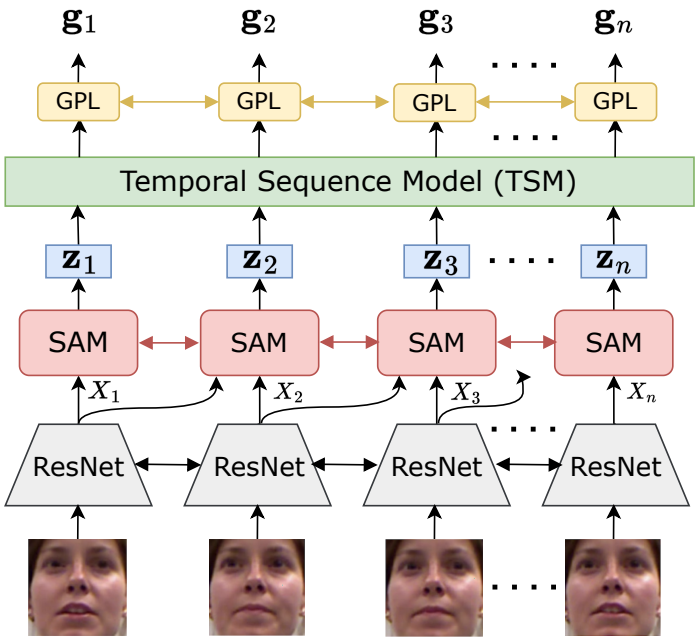
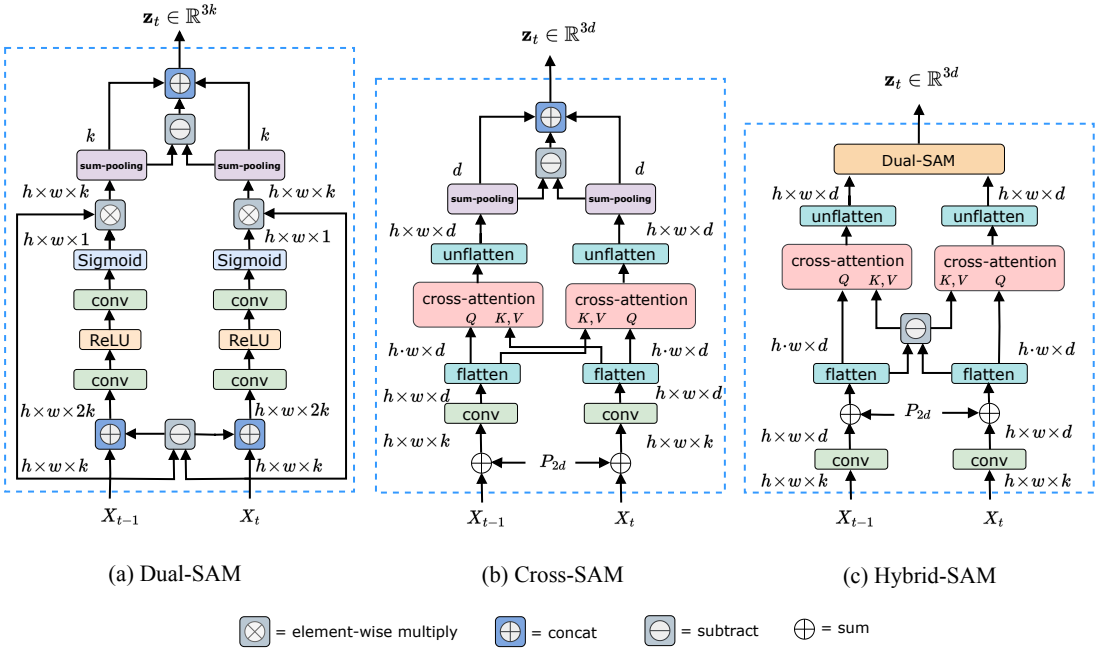


Figure 6.2: *A schematic overview of the proposed (person-agnostic) STAGE model.* The proposed model has three modules: spatial attention module (SAM), temporal sequence model (TSM), and gaze prediction layer (GPL). The SAM is designed to extract information relevant to the gaze by concentrating on the spatial differences between consecutive frames. In the figure, X_i represents features from ResNet, $\mathbf{z}_i$ denotes the motion-informed output of the SAM, and $\mathbf{g}_i$ corresponds to the predicted gaze direction.

through a shared SAM. Concretely, the SAM module aggregates information from RGB features of X_{t-1} and X_t, and the feature differences $(X_t - X_{t-1})$ through a fusion strategy. Figure 6.3 provides an overview of all three SAM variants considered in this work. All SAM variants are optimized during model training and outputs $\mathbf{z}_t$, a feature representation with spatial motion information for the t^{th} frame of the video.



(a) Dual-SAM (b) Cross-SAM (c) Hybrid-SAM

Figure 6.3: **Block diagram of SAM variants.** For each variant, the input is a pair of the consecutive frame features X_{t-1} and X_t, and the output is a 1D feature vector encoding both RGB and motion information. P_{2d} are 2D positional embeddings with height and width same as the input feature map. The *cross-attention* block in Cross-SAM and Hybrid-SAM is a standard transformer operation. The *sum-pooling* block applies feature pooling by summing them over height and width dimensions. In Hybrid-SAM, the keys and values for the cross-attention block are residual features, i.e., the difference in features at t and $t-1$.

Dual-SAM. Dual-SAM predicts separate spatial attention maps for both current X_t and past X_{t-1} frame. It compares the spatial attention maps of the current and past frames and identifies the region that is most relevant to the observed motion changes. If the spatial attention maps are very similar, SAM infers that there is no substantial change between consecutive frames and encodes these minimal

differences in the output vector $\mathbf{z}_t \in \mathbb{R}^{3k}$. Conversely, if there is a difference, SAM incorporates this change into the output vector $\mathbf{z}_t$. This SAM variant is inspired by Park et al. [216] in the change captioning task and is shown in Figure-6.3a. The Dual-SAM algorithm is provided in Algorithm 1. σ denotes the sigmoid function, and $\odot$ is an element-wise dot product.

Algorithm 1 Dual-Spatial Attention Module (Dual-SAM)

Input: X_{t-1}, X_t $\in \mathbb{R}^{h \times w \times k}$

Output: $\mathbf{z}_t$ $\in \mathbb{R}^{3 \cdot k}$

1: $X'_{t-1} = [X_{t-1}; X_t - X_{t-1}]$

$X'_t \quad = [X_t; X_t - X_{t-1}]$ $\in \mathbb{R}^{h \times w \times 2 \cdot k}$

2: $A_{t-1} = \sigma(\text{conv}(\text{ReLU}(\text{conv}(X'_{t-1}))))$

$A_t = \sigma(\text{conv}(\text{ReLU}(\text{conv}(X'_t))))$ $\in \mathbb{R}^{h \times w \times 1}$

3: $\mathbf{v}_{t-1} = \sum_{h,w} A_{t-1} \odot X_{t-1}$

$\mathbf{v}_t = \sum_{h,w} A_t \odot X_t$ $\in \mathbb{R}^{k}$

4: $\mathbf{z}_t = [\mathbf{v}_{t-1}; \mathbf{v}_t - \mathbf{v}_{t-1}; \mathbf{v}_t]$ $\in \mathbb{R}^{3 \cdot k}$

5: **return** $\mathbf{z}_t$

Cross-SAM. Unlike Dual-SAM, this variant utilizes cross-attention from transformer models [224] to encapsulate dense correlation between each pair of image patches in the past and current frames. This allows Cross-SAM to identify multiple changes between two frames, as opposed to Dual-SAM, which can only capture a single change. Practically, detecting multiple changes and subsequently filtering out irrelevant distractors is more useful for video gaze estimation tasks. Similar to the Dual-SAM, this variant utilizes both RGB and transformed motion signals at

the output. Qiu et al. [217] motivates the design of Cross-SAM and is shown in the Figure-6.3b. Algorithm 2 describes the Cross-SAM module.

Algorithm 2 Cross-Spatial Attention Module (Cross-SAM)

Input: X_{t-1}, X_t $\in \mathbb{R}^{h \times w \times k}$

Output: $\mathbf{z}_t$ $\in \mathbb{R}^{3 \cdot d}$

1: $X_{t-1} = \text{flat}(\text{conv}(X_{t-1}) + \mathbb{1}_{h,w} \odot P_{2d})$

$X_t = \text{flat}(\text{conv}(X_t) + \mathbb{1}_{h,w} \odot P_{2d})$ $\in \mathbb{R}^{h \cdot w \times d}$

2: $X_{t-1} = \text{crossatten}(X_{t-1}, X_t, X_t)$

$X_t = \text{crossatten}(X_t, X_{t-1}, X_{t-1})$ $\in \mathbb{R}^{h \cdot w \times d}$

3: $\mathbf{v}_{t-1} = \sum_{h,w} \text{unflat}(X_{t-1}, h \times w)$

$\mathbf{v}_t = \sum_{h,w} \text{unflat}(X_t, h \times w)$ $\in \mathbb{R}^{d}$

4: $\mathbf{z}_t = [\mathbf{v}_{t-1}; \mathbf{v}_t - \mathbf{v}_{t-1}; \mathbf{v}_t]$ $\in \mathbb{R}^{3 \cdot d}$

5: **return** $\mathbf{z}_t$

Hybrid-SAM. The Hybrid-SAM combines the strengths of both Dual-SAM and Cross-SAM variants. Dual-SAM focuses on one local change, while Cross-SAM focuses on global context and captures multiple changes. Similar to Cross-SAM, Hybrid-SAM encapsulates multiple changes by applying a cross-attention mechanism using global context through position embeddings. However, unlike the Cross-SAM variant, it uses the difference between current and past frames as a key and value, emphasizing regions with the most significant motion differences.

Algorithm 3 Hybrid-Spatial Attention Module (Hybrid-SAM)

Input: X_{t-1}, X_t $\in \mathbb{R}^{h \times w \times k}$

Output: $\mathbf{z}_t$ $\in \mathbb{R}^{3 \cdot d}$

1: $X_{t-1} = \text{flat}(\text{conv}(X_{t-1}) + \mathbf{1}_{h,w} \odot P_{2d})$

$X_t = \text{flat}(\text{conv}(X_t) + \mathbf{1}_{h,w} \odot P_{2d})$ $\in \mathbb{R}^{h \cdot w \times d}$

2: $X_{\text{diff}} = X_t - X_{t-1}$

3: $X_{t-1} = \text{crossatten}(X_{t-1}, X_{\text{diff}}, X_{\text{diff}})$

$X_t = \text{crossatten}(X_t, X_{\text{diff}}, X_{\text{diff}})$ $\in \mathbb{R}^{h \cdot w \times d}$

4: $X_{t-1} = \text{unflat}(X_{t-1}, h \times w)$

$X_t = \text{unflat}(X_t, h \times w)$ $\in \mathbb{R}^{h \times w \times d}$

5: $X'_{t-1} = [X_{t-1}; X_t - X_{t-1}]$

$X'_t = [X_t; X_t - X_{t-1}]$ $\in \mathbb{R}^{h \times w \times 2 \cdot d}$

6: $A_{t-1} = \sigma(\text{conv}(\text{ReLU}(\text{conv}(X'_{t-1}))))$

$A_t = \sigma(\text{conv}(\text{ReLU}(\text{conv}(X'_t))))$ $\in \mathbb{R}^{h \times w \times 1}$

7: $\mathbf{v}_{t-1} = \sum_{h,w} A_{t-1} \odot X_{t-1}$

$\mathbf{v}_t = \sum_{h,w} A_t \odot X_t$ $\in \mathbb{R}^{d}$

8: $\mathbf{z}_t = [\mathbf{v}_{t-1}; \mathbf{v}_t - \mathbf{v}_{t-1}; \mathbf{v}_t]$ $\in \mathbb{R}^{3 \cdot d}$

9: **return** $\mathbf{z}_t$

The Dual-SAM is utilized as a pooling operator to selectively focus on the most relevant changes, like eye or head movements, which are crucial for the task of gaze estimation. The Hybrid-SAM is given in the Algorithm 3. The input is features of the past frame X_{t-1} the and current frame X_t, respectively. Both input features are projected to the higher-dimensional feature maps using the convolution operation, and 2-D position embeddings $P_{2d} \in \mathbb{R}^{h \times w}$ are added (Line 1). Line 2 computes difference features X_{diff} for the video's t^{th} frame, and cross-attention is applied in

Line 3. Lines 5-8 correspond to the same operations as in Dual-SAM. $\mathbf{1}_{h,w}$ is a one-hot vector spanning the spatial dimensions.

6.2.2 Temporal Sequence Model (TSM)

The temporal sequence model subsumes spatially enhanced representations $\mathbf{z}_t$ produced by the SAM module and is intended to capture the temporal dynamics of the eye movements in the video. In particular, we consider two variants for TSM: recurrent neural networks (RNN) [225], and transformer network [224]. The RNN consists of unidirectional LSTM layers [226], and the transformer variant is a causal transformer decoder, which is prevalent in generative language modeling, such as the GPT-2 model [227].

We incorporate learned temporal position embeddings to enable the transformer model to discern temporal relationships within the input feature sequence. These embeddings are uniquely associated with each position, providing the model with explicit information about the relative ordering of elements within the sequence. The embedded features are then passed through multiple layers, each consisting of masked multi-head attention, LayerNorm (LN), and a Multi-Layer Perceptron (MLP) as shown in Figure 6.4. Masked multi-head attention allows the transformer model to attend to only past frame features. The output of the TSM is a feature sequence passed through an LN layer, similar to the GPT-2 model [227].

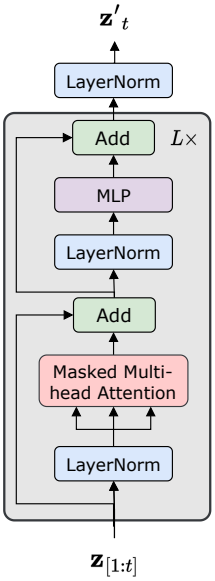


Figure 6.4: Block diagram of single transformer layer used in the temporal sequence model of the STAGE method. MLP is a Multi-Perceptron layer, and we use L blocks stacked together in the TSM.

6.2.3 Gaze Prediction Layer and Training Objective

The gaze prediction layer is shared across all timestamps and uses an MLP to predict the gaze direction from the frame embeddings generated by the TSM module. For i^{th} sample and t^{th} frame, let $\{\mathbf{g}_t^i\}$ and $\{\hat{\mathbf{g}}_t^i\}$ denote the sequences of true and predicted gaze directions, respectively. Similarly, $\{\mathbf{p}_t^i\}$ and $\{\hat{\mathbf{p}}_t^i\}$ represent the sequences of true and predicted 2D Point-of-Gaze (PoG). We use the following objective function for training STAGE parameters (similar to Park et al. [89]):

$$\mathcal{L}_{final} = \frac{1}{b \cdot n} \sum_{i=1}^{b} \sum_{t=0}^{n-1} \frac{180}{\pi} \arccos\left(\frac{{\mathbf{g}_t^i}^T \hat{\mathbf{g}}_t^i}{|\mathbf{g}_t^i| \cdot |\hat{\mathbf{g}}_t^i|}\right) + \lambda \cdot ||\mathbf{p}_t^i - \hat{\mathbf{p}}_t^i|| \tag{6.1}$$

Here, λ is the weight parameter that controls the trade-off between 3D gaze angular error and 2D PoG mean absolute error. The second term is applied exclusively to datasets that have available ground-truth PoG.

6.2.4 Personalizing STAGE using Gaussian Processes

As previously mentioned, we propose person-specific Gaussian processes for modeling bias correction terms for each user, which operates on top of the proposed (person-agnostic) STAGE model. Specifically, if $f : V \mapsto G$ is the STAGE model, then the final prediction for person p is $\hat{\mathbf{f}}_p(V) = \mathbf{f}(V) + \mathbf{r}_p(V)$, where $\mathbf{r}_p$ is GP-based bias correction model for the person p, i.e., it predicts the residual in addition to the model-agnostic prediction. $\mathbf{r}_p$ models the components of gaze direction (i.e., yaw and pitch) independently at the frame level, using two one-dimensional independent GPs. Concretely, $\mathbf{r}_p(V) = [(r_{p,\theta}(V_1), r_{p,\phi}(V_1)), (r_{p,\theta}(V_2), r_{p,\phi}(V_2)), \cdots, (r_{p,\theta}(V_n), r_{p,\phi}(V_n))]$, where $r_{p,\theta}$ and $r_{p,\phi}$ are the one-dimensional GP predictions for pitch and yaw components, respectively.

For GP hyper-parameter tuning and inference, we collect a set of training frames $\mathcal{D} = \{\mathbf{h}_i, y_i\}_{i=1}^{\ell}$ that are available for person p, where $\mathbf{h}_i \in \mathbb{R}^d$ are the flattened ResNet output features from the STAGE model, and y_i is either pitch or yaw of residual gaze angle, i.e., $\mathbf{g}_i - \hat{\mathbf{g}}_i$, where, $\mathbf{g}_i$ and $\hat{\mathbf{g}}_i$ are true gaze direction and STAGE's predicted direction, respectively. To represent the dataset $\mathcal{D}$ in matrix

format, we let $\mathbf{y} \in \mathbb{R}^{\ell}$ be the vector of residual angles, where the i^{th} entry equal to y_i, and $H \in \mathbb{R}^{\ell \times d}$ have its i^{th} row equal to the ResNet features $\mathbf{h}_i$. For brevity, we omit the person index p from henceforth discussion on GPs.

A Gaussian process associated with kernel (covariance) function $k(\mathbf{h}, \mathbf{h}') : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a distribution over functions that maps features to residual angles such that, for any $\mathbf{h}_1, \ldots, \mathbf{h}_\ell \in \mathbb{R}^d$:

$$\mathbf{r} = [r(\mathbf{h}_1), ..., r(\mathbf{h}_\ell)] \sim \mathcal{N}(\mu_0, K_H), \tag{6.2}$$

where $K_H = [k(\mathbf{h}_i, \mathbf{h}_j)]_{i,j=1}^{\ell} \in \mathbb{R}^{\ell \times \ell}$ is the kernel (covariance) matrix on the data points H, and r has a constant mean function with its value set to μ_0. The observed residual angle y_i is modeled as the i.i.d. Gaussian noise, i.e., $y_i \sim \mathcal{N}(r(\mathbf{h}_i), \sigma^2 I)$. In particular, we use the (squared-exponential) automatic-relevance-determination (ARD) kernel, given as $k(\mathbf{h}, \mathbf{h}') = \tau \cdot e^{-\sum_{s=1}^{d} \frac{\left(\mathbf{h}(s) - \mathbf{h}'(s)\right)^2}{\theta(s)^2}}$, where τ and $\theta \in \mathbb{R}^d$ are kernel hyper-parameters. The ARD kernel's per-dimension scaling, being more expressive than the RBF kernel's use of a single length-scale, often leads to superior practical performance [228]. Intuitively, this flexibility allows the model to adapt to varying feature relevance and noise levels, potentially leading to improved accuracy and generalization [229]. Upon conditioning the GP model on the collected training

dataset, the predictive posterior mean and covariance functions are as follows:

$$\textbf{mean: } \mu_{r|\mathcal{D}}(\mathbf{h}) = \mathbf{k}_{\mathbf{h}}^T (K_H + \sigma^2 I)^{-1} \mathbf{y}$$

$$\textbf{variance: } \sigma_{r|\mathcal{D}}(\mathbf{h}) = k(\mathbf{h}, \mathbf{h}) - \mathbf{k}_{\mathbf{h}}^T (K_H + \sigma^2 I)^{-1} \mathbf{k}_{\mathbf{h}}$$

where the vector $\mathbf{k}_{\mathbf{h}} \in \mathbb{R}^{\ell}$ has i^{th} entry $k(\mathbf{h}, \mathbf{h}_i)$, i.e., kernel value between any feature vector $\mathbf{h}$ and i^{th} data point. The posterior mean function predicts the residual gaze angles and is utilized for correction. The posterior covariance function determines the uncertainty in this prediction, as illustrated in Figure 6.8.

Optimizing GP hyper-parameters using very few samples. GPs are non-parametric models and thus do not require tuning many parameters [220]. However, they still necessitate optimizing hyperparameters, which in our case are μ_0, σ, τ, and θ, totaling $d + 3$ hyperparameters as $|\theta| = d$. The ARD kernel adds flexibility to the GP model but also increases the number of hyperparameters to be tuned. Specifically, since $d = 16384$ when using features from the ResNet model, directly tuning hyperparameters using the log-likelihood of data $\mathcal{D}$ is prone to overfitting, particularly when as few as three samples are present in $\mathcal{D}$. To overcome this challenge, we propose the application of pre-trained GPs, similar to the concurrent work by Wang et al. [223]. Pre-trained GPs entail the initial optimization of hyperparameters on data used for training the STAGE model, coupled with early stopping during optimization to maximize the log-likelihood of

dataset $\mathcal{D}$ for each individual. This approach provides GPs with flexibility through the use of an expressive ARD kernel and ensures a strong initial position, thanks to pre-training.

6.3 Experiments

6.3.1 Setup

Datasets. EVE [89] is a large-scale video-based gaze dataset comprising over 12 million frames collected from 54 participants in a controlled indoor setting with four synchronized and calibrated camera views. Following the splits used by Park et al. [89], there are 40 subjects in training and 6 subjects in the validation set. We discard the data from test subjects due to the unavailability of labels and evaluate our models on the validation set. Gaze360 [88] is a large-scale, physically unconstrained gaze dataset collected from 238 subjects with a wide range of head poses. It has 129K training images, 17K validation images, and 26K test images. We evaluate our models on all three subsets of the dataset: the full Gaze360 dataset, the front 180° subset, and the front 20° subset, as done by Kellnhofer et al. [88]. EyeDiap [2] consists of 94 videos totaling 237 minutes, collected from 16 subjects in a laboratory environment. The EyeDiap dataset includes videos for both screen and floating targets and we select VGA videos of screen targets.

Implementation Details. The input video sequence V consists of 30 frames containing a full-face image of 128×128 pixels. We use ResNet-18 [230] initialized with GazeCLR [231] weights shared between all timestamps to extract visual features from the image sequence. The third convolutional layer block of ResNet-18 outputs features with a dimension of $256 \times 8 \times 8$. We pass these features through the SAM module, followed by TSM and gaze prediction layers. We train STAGE end-to-end for 50K iterations using the SGD optimizer with an initial learning rate of 0.016 and momentum of 0.9. The learning rate is decayed using cosine annealing [195], and the batch size is set to 16.

The Dual-SAM consists of two convolutional layers with kernel size 1 and output feature maps of 64 and 1, respectively. The first convolutional layer has a group normalization layer [232] applied to the output features, followed by a dropout layer with $p = 0.5$. In Cross-SAM and Hybrid-SAM, we project the incoming features to higher channels through a convolution layer with $d = 512$ and a kernel size 1. After adding 2D positional embeddings to the projected feature maps, they go through the cross-attention encoder, which consists of four heads and two layers with an embedding size of 64.

The TSM model has two variants: an LSTM variant and a transformer variant. The LSTM variant consists of one unidirectional LSTM layer with a hidden dimension of 128. The transformer variant is based on GPT-2 [227] network with

6-heads and 6-layers, operating on a dimension of $d = 128$, and initialized randomly. The gaze prediction layer consists of two fully connected (FC) layers. The first FC layer has a SeLU activation function and a hidden dimension of the same size as the input dimension. The second FC layer outputs the 2D gaze direction angles, pitch, and yaw.

Our STAGE model is implemented in PyTorch [233]. We set $\lambda = 0.001$ for cross-data and $\lambda = 0$ for within-data evaluations. For GP hyper-parameter optimization, we use Adam optimizer with a learning rate of 0.001, implemented using GPytorch [234].

6.4 Results

In Section 6.4.1, we provide visual examples of attention maps superimposed on video frames, illustrating the qualitative impact of the SAM block in improving the overall performance of the STAGE. In addition to qualitative assessment, we provide quantitative evaluation of the SAM and TSM variants in two experimental settings: within-dataset (in Section 6.4.2) and cross-dataset (in Section 6.4.3). The primary objective of these experiments is to evaluate the effectiveness of incorporating a SAM block prior to the temporal sequence model in enhancing video gaze estimation accuracy. In Section 6.4.4, we also benchmark our proposed method against current leading methods in video gaze estimation for a within-dataset

setting. Both qualitative and quantitative evaluation of GP-based personalization on EyeDiap participants is provided in Section 6.4.5. We discuss an ablation study on the number of SAM blocks in Section 6.4.6.

Baselines. We benchmarked our framework against EyeNet [89], which consists of ResNet-18 and RNN layers and uses both eye image patches as input. We adopted EyeNet to our setting and trained it on full-face images using $\mathcal{L}_{final}$ with $\lambda = 0.001$. We also train another variant of EyeNet by replacing the RNN module with a TSM similar to that used in our framework. For a fair comparison, we also implement EyeNet with our version of ResNet-18 initialized with GazeCLR [231] weights and call it *EyeNet (GazeCLR)*. Further, we adapt the work of Chang et al. [235] for gaze estimation, which introduces motion-aware-unit (MAU) for the video-prediction task. We also compare with a simple baseline by removing the SAM modules and concatenating X_t and $X_{\text{diff}} = (X_t - X_{t-1})$ before passing through TSM, termed *Concat-Residual*. Finally, we compare the three variants of SAM modules combined with two variants of TSM for cross-dataset and within-dataset experiments. For the sake of completion, we also evaluate the Hybrid-SAM method without the Dual-SAM module at the output, named as Hybrid-SAM†.

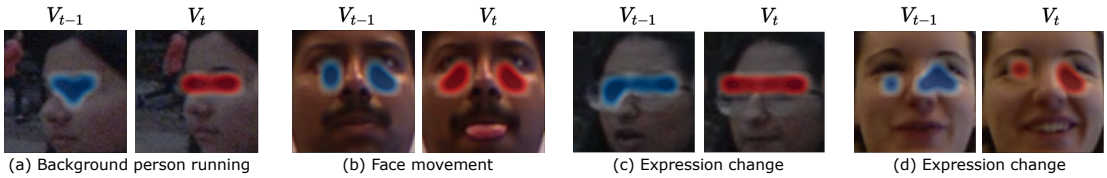


Figure 6.5: Illustration of attention maps A_{t-1} and A_t, generated by the Hybrid-SAM, superimposed on sequential video frames V_{t-1} and V_t. The SAM module proficiently highlights the ocular area, key for analyzing eye movements, while simultaneously diminishing irrelevant distractions such as background motion (a), tongue movement (b), and changes in emotional expressions (c and d).

6.4.1 Qualitative Evaluation

We conducted a qualitative analysis primarily centered on assessing the Hybrid-SAM ability to distinguish between gaze-irrelevant distractors and gaze-relevant eye movements, which is crucial for video gaze estimation, as stated earlier. Specifically, we examined attention maps A_{t-1} and A_t, strategically overlaid on sequential video frames V_{t-1} and V_t, as depicted in Figure 6.5. We analyzed several frames showcasing scenarios from background activities to facial movements, all concurrent with dominant eye movements.

In Figure 6.5(a), the network adeptly focuses on eye movements in frame V_t (red pixels) and prior frame changes (blue pixels) despite significant background pixel shifts from a walking person. This underscores the effectiveness of spatial attention in filtering out irrelevant distractors to accurately identify subtle eye movements and gaze direction. As a result, it eases the process of temporal modeling in video

gaze estimation. Additionally, as illustrated in Figure 6.5(b), although tongue movement presents a potential distraction, it is efficiently disregarded. Moreover, changes in facial expressions, depicted in Figure 6.5(c, d), are effectively overlooked by the Hybrid-SAM. These qualitative findings affirm that the spatio-temporal attention strategy adeptly minimizes significant distractions, particularly in the eye region, which is essential for accurately tracking gaze and eye movements in video gaze estimation tasks.

6.4.2 Within-dataset Evaluation

In the within-dataset experiments, we train and evaluate our model on the same domain dataset. Table 6.1 shows results for the within-dataset evaluation. We train our framework on the training subset of Gaze360 with $\lambda = 0$ and evaluate it over three test subsets as done in Kellnhofer et al. [88]. Our model demonstrates superior performance compared to the baseline models, including 'Concat-Residual', across all three subsets. Specifically, it achieves absolute improvements of 2.5°, 2.2° and 2.5° on full Gaze360, front 180° and front 20° subsets, respectively. Furthermore, it is noteworthy that Hybrid-SAM performs better in comparison to Hybrid-SAM†, illustrating the advantage of incorporating Dual-SAM as the pooling operator.

Method	**Full**	**180°**	**Front 20°**
EyeNet [89](GazeCLR)	12.53	12.08	9.45
EyeNet + Tx	13.00	12.55	9.73
Concat-Residual + LSTM	10.35	10.16	7.45
Concat-Residual + Tx	12.22	11.78	9.09
Dual-SAM + LSTM	10.12	9.92	<u>7.08</u>
Dual-SAM + Tx	10.13	9.93	7.23
Cross-SAM + LSTM	12.00	11.59	9.51
Cross-SAM + Tx	10.12	9.91	7.34
Hybrid-SAM† + LSTM	12.69	12.26	9.66
Hybrid-SAM† + Tx	12.33	11.90	9.53
Hybrid-SAM + LSTM	**10.05**	**9.84**	**6.92**
Hybrid-SAM + Tx	<u>10.10</u>	<u>9.90</u>	7.33

Table 6.1: **Within-dataset Evaluation.** Comparison of mean angular errors (in degrees) between the proposed STAGE model, SAM and TSM variants, and other baseline approaches. Full, 180° and 20° are subsets of the Gaze360 dataset. Tx is the transformer TSM model. The **first** and <u>second</u> best results are bold-ed and underlined, respectively.

6.4.3 Cross-dataset Evaluation

We performed a cross-dataset evaluation, where the model was trained on the EVE dataset and evaluated on two different domain datasets, EyeDiap and Gaze360. Table 6.2 shows the comparison of mean angular errors (MAE) for the baselines and our proposed method. We observed a significant improvement in both

Method	**EyeDiap**	**Full**	**180°**
MAU	21.30	34.18	33.57
EyeNet [89]	16.07	31.37	30.77
EyeNet (GazeCLR)	7.74	26.57	25.95
EyeNet + Tx	8.40	26.25	25.64
Concat-Residual+ LSTM	7.12	24.12	23.52
Concat-Residual+ Tx	7.27	24.26	23.64
Dual-SAM + LSTM	7.04	24.18	23.58
Dual-SAM + Tx	6.77	23.99	23.38
Cross-SAM + LSTM	8.42	23.19	22.61
Cross-SAM + Tx	8.75	**22.57**	**22.01**
Hybrid-SAM† + LSTM	8.48	23.31	22.72
Hybrid-SAM† + Tx	7.79	<u>22.66</u>	<u>22.09</u>
Hybrid-SAM + LSTM	<u>6.70</u>	23.73	23.13
Hybrid-SAM + Tx	**6.54**	23.77	23.17

Table 6.2: **Cross-dataset Evaluation.** Comparison of mean angular error (in degrees) between the proposed STAGE model, SAM and TSM variants, and other baseline approaches. Full and 180° are subsets of the Gaze360 dataset. Tx is the transformer TSM model. For each column, the **first** best result is bold-ed, and <u>second</u> best result is underlined.

datasets even with a simple concatenation of X_t and X_{diff}, i.e., Concat-Residual approach outperforms EyeNet variants and MAU approach, which demonstrates that residual frames are an effective cue for video-gaze estimation.

The Dual-SAM and Cross-SAM variants show improvements over the Concat-Residual approach, indicating that the adapted methods are more accurate than naively using residual frames. Notably, the Hybrid-SAM approach improves over baselines by 1.2° in absolute and 14.28% in relative, on the EyeDiap dataset. It also outperformed the other Dual-SAM and Cross-SAM variants on all three evaluation sets. The last two columns of Table 6.2 show results on the full and front 180° Gaze360 subsets, which are similar to the subsets used in Kellnhofer et al. [88]. The Hybrid-SAM approach improved up to 3.6° on both subsets, further emphasizing the effectiveness of the SAM module. It is also worth noting that the performance improvements for the SAM variants hold for both LSTM and transformer-based TSM in both Tables 6.1 and 6.2. This shows that the SAM is helpful irrespective of the choice of the TSM model.

6.4.4 Comparison with State-of-the-art Methods

Table 6.3 compares the proposed STAGE method with state-of-the-art approaches for a within-dataset setting. Video-based gaze estimation methods such as the original work of Gaze360 [88] and MSA+Seq [236] employ the LSTM model and learn through the Pinball loss function. We also compare our proposed gaze estimation approach with image-based methods such as L2CS-Net [238], both variants of GazeTR [239], and self-supervised learning-based method SwAT [237].

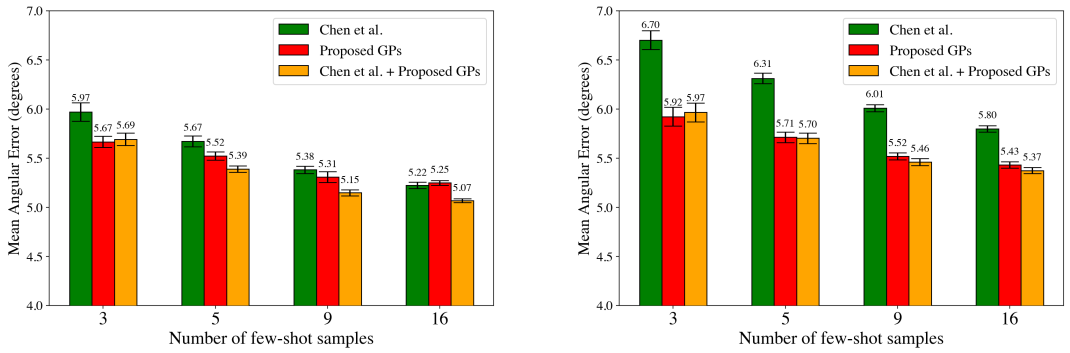
Method	Full	180°	Front 20°
Gaze360 [88]	13.50	11.40	11.10
MSA+Seq [236]	12.50	10.70	-
SwAT [237]	11.60	-	-
L2CS-Net [238]	-	10.41	9.02
GazeTR-Pure [239]	-	13.58	-
GazeTR-Hybrid [239]	-	10.62	-
Hybrid-SAM + LSTM	**10.05**	**9.84**	**6.92**
Hybrid-SAM + Tx	<u>10.10</u>	<u>9.90</u>	<u>7.33</u>

Table 6.3: **STAGE *vs.* State-of-the-art.** Comparison with state-of-the-art methods on Gaze360 data subsets under the within-dataset setting (Tx = transformer-based TSM). The metric is the mean angular error (in degrees). The **first** and <u>second</u> best results are bold-ed and underlined, respectively.

We report the performance of these methods from the original work and show a comparison with our method. Our best results outperform these methods by 1.5°, 0.5° and 2.1° on full Gaze360, front 180° and front 20°, respectively. The superior performance of our method demonstrates the effectiveness of SAM and our choice for other components of the overall STAGE model.

6.4.5 Evaluating STAGE with GP Personalization

As stated earlier, we first optimize the hyper-parameters of the GP model $\mathbf{r}_p$ for residual gaze direction prediction using the training subset of the EVE dataset.



(a) Comparison of Dual-SAM + Tx (b) Comparison of Hybrid-SAM + Tx

Figure 6.6: The figure shows the comparison of ℓ-shot GP personalization on the STAGE model with Chen and Shi [5] for the EyeDiap dataset. The bars indicate the mean angular error (in degrees) and standard error over 10 iterations. The *Proposed GPs* consistently outperform the baseline for both SAM variants and achieve the best results when used in conjunction with Chen and Shi [5].

Then, we adapt $\mathbf{r}_p$ for personalization on the EyeDiap participants. We randomly sample ℓ video frames for each participant 10 times and report the performance in Figure 6.6. We perform GP personalization on two SAM variants: Dual-SAM and Hybrid-SAM, using a transformer TSM model. The baseline method proposed by Chen and Shi [5], involves learning a single person-specific bias during training and utilizing a few labeled samples to predict bias during inference.

We obtain an absolute improvement of around 0.8° with the Hybrid-SAM over the baseline with as few as 3 samples. Applying GPs with the baseline objective, i.e., "Chen *et al.* + GPs", we see consistent improvements over both GPs and the method proposed by Chen and Shi [5]. These results demonstrate that GPs' are a

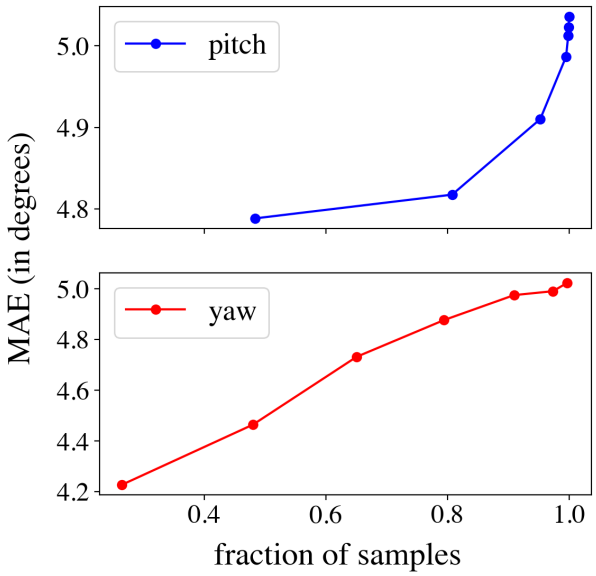


Figure 6.7: Comparison of Mean Angular Error (in degrees) of gaze components (yaw or pitch) with increasing fraction of test samples sorted with respect to the uncertainty of GP predictions. Plots exhibit that GPs are more accurate when the prediction is relatively more confident (with less variance).

valuable tool and provide complementary strengths to Chen and Shi [5]. Unlike Chen and Shi [5], GPs do not require altering the objective for training the deep network. They can be utilized for adaptation with any pre-trained existing model, such as STAGE.

For assessing the effectiveness of the GP model's uncertainty, we provide both qualitative and quantitative analysis of gaze predictions, as illustrated in Figures 6.7 and 6.8. Our evaluation begins with an analysis of the GP's posterior variance diagonal. We arrange this in ascending order and then apply different uncertainty thresholds to it. For each selected threshold, we compute the MAE on test samples

that exhibit uncertainty levels below the threshold. This procedure is repeated across a range of different thresholds to evaluate performance. Figure 6.7 presents a comparison of the MAE for yaw and pitch against increasing fractions of test data samples. These samples are sorted according to the uncertainty in the GP prediction. This analysis demonstrates that GPs tend to deliver more accurate results when their variance is lower, signifying greater confidence in the predictions. Therefore, the uncertainty measure in the GP model can act as an effective indicator to avoid making inaccurate predictions.

We then examine the qualitative results depicted in Figure 6.8, which showcase the differences between confident and uncertain gaze predictions after personalization using the EyeDiap dataset. Notably, the uncertainty region typically includes the ground truth, as illustrated by the pink arrows within the green area. It is crucial to note that gaze predictions with higher uncertainty often align with situations that are challenging for human interpretation like extreme head poses or closed eyes.

6.4.6 Ablation Study

In the ablation study, we study the impact of adding multiple SAM blocks in the STAGE model, where the output of one SAM goes as input to the next. The ablation study on the number of Dual- and Hybrid-SAM blocks (four blocks vs.

(a) Examples of certain predictions

(b) Examples of uncertain predictions

Figure 6.8: The figure depicts a few (a) certain and (b) uncertain predictions for gaze directions after GP's personalization on the EyeDiap dataset. The blue and pink arrows show ground truth and predicted gaze directions, respectively. The green-colored region offers uncertainty of the predictions in the pink arrows.

one block) for within-data and cross-data settings are shown in Tables 6.4(a) and (b), respectively. We observe no significant improvements over a single block of SAM, indicating that one SAM block is enough to provide spatial motion cues between consecutive frame features and improve performance.

Method	Full	180°	20°
Dual-SAM(1-block)+Tx	10.13	9.93	7.23
Hybrid-SAM(1-block)+Tx	10.10	9.90	7.33
Dual-SAM(4-blocks)+Tx	12.13	11.68	9.33
Hybrid-SAM(4-blocks)+Tx	10.25	10.08	7.27

(a) Within-dataset evaluation

Method	EyeDiap	Full	180°
Dual-SAM(1-blocks)+Tx	6.77	23.99	23.38
Hybrid-SAM(1-blocks)+Tx	6.54	23.77	23.17
Dual-SAM(4-blocks)+Tx	7.27	23.34	22.74
Hybrid-SAM(4-blocks)+Tx	7.55	23.52	22.91

(b) Cross-dataset evaluation

Table 6.4: **Ablation Study:** Comparison of different numbers of SAM blocks employed in our STAGE method. Tx is transformer-based TSM, and training is performed for within-data and cross-data settings in (a) and (b), respectively. The metric reported is mean angular errors (in degrees).

6.5 Summary

This chapter presents a novel video gaze estimation method called STAGE that uses a deep learning network to encode spatial motion and temporal dynamics. The method uses a spatial attention module to implicitly focus on the frame difference between consecutive frames and highlight relevant changes. We show that the performance of the proposed STAGE model can be further improved using a few labeled samples with Gaussian processes. In the future, extending the receptive field of attention modules and fusing the long-term spatial and temporal dynamics would be interesting.

Chapter 7

Conclusion and future directions

Gaze has intriguing properties and plays a key role in revealing user's intentions and areas of interest, thus enabling the advancement of intelligent interactive systems. Traditional model-based and feature-based approaches [37] are capable of delivering high precision; however, they require specialized hardware, such as high-resolution cameras and external infrared light sources, to detect eye features. In contrast, appearance-based gaze estimation methods learn to map directly from raw pixel images to gaze direction, enabling them to function with low-resolution images taken by webcams [30, 240]. Numerous learning-based approaches employ deep learning techniques and, therefore, require a large amount of labeled training data to achieve good performance in gaze direction estimation. For these reasons, significant progress has been made in cross-domain and few-shot gaze estimation.

In this thesis, we focused on bridging the gap between the need for labeled data and the performance of gaze estimation across both image and video inputs captured via webcams. Our initial contribution involves introducing methods for efficiently acquiring annotated gaze datasets for appearance-based approaches.

We proposed utilizing commercial trackers and calibrating them with webcams to gather annotations in the camera's reference frame, which is required to train neural networks. Furthermore, we developed a generative network to create an augmented gaze-labeled dataset by manipulating the gaze direction in the generated images. This augmented dataset was then employed to boost gaze estimation performance. In our second contribution, we focused on improving representation learning to enhance gaze estimation performance. We introduced a self-supervised framework, referred to as *GazeCLR*, designed to learn gaze representations from unlabeled multi-view images. These representations are subsequently utilized to enhance the accuracy of cross-domain, few-shot, personalized gaze estimation. Furthermore, we developed a spatio-temporal model, named STAGE, designed to learn representations for video inputs. This model employs attention modules to detect local spatial variations and understand global temporal dynamics. We personalize this model by learning an additive bias model through Gaussian processes.

In this chapter, we further discuss the contributions of this thesis (Section 7.1) and provide a brief overview of the promising future research directions for gaze estimation related to work presented in the thesis (Section 7.2).

7.1 Thesis Contributions

The contributions of this thesis are outlined on a chapter-by-chapter basis in the subsequent paragraphs.

In *Chapter 3*, we introduced a simpler and faster approach to calibrating commercial infrared-based gaze trackers with cameras embedded in laptops. While commercial trackers supply the necessary and precise annotations, such as gaze origin crucial for gaze estimation tasks, these annotations are typically defined in the tracker's reference frame. Consequently, previous data collection methods [89] overlook these labels due to the absence of tracker-camera calibration and instead depend on 3D face models to determine the gaze origin's location. Our proposed calibration algorithm facilitates the expression of all quantities in the camera's reference frame, accelerating the collection of gaze-labeled data to enhance the performance of appearance-based gaze estimation methods. In our approach, we instruct users to look directly at the camera across various head poses and distances, ensuring they remain within the tracker's permissible range. We then apply a pupil detection algorithm to the images, creating pairs of 3D gaze origin (in the tracker frame) and 2D pupil center (in the image plane). These paired data points are processed through a Perspective-n-Point (PnP) algorithm to determine the rotation and translation between the camera and the tracker. Our empirical analysis shows that our method yields more accurate ground-truth gaze directions

compared to those derived from 3D face models.

In *Chapter 4*, we introduced an unsupervised domain adaptation framework, named CUDA-GHR, designed for controlling gaze and head pose in generated images without altering the person's appearance. This framework effectively disentangles factors like gaze, appearance, and head pose, learning to individually manipulate these elements in the output image. The framework is trained with supervision from the source domain and then adapted to an unlabeled target domain. Our experimental findings indicate that our framework produces more photo-realistic images while accurately redirecting gaze and head pose. We also demonstrate how the dataset generated by this framework contributes to improving the accuracy of gaze and head pose estimation performance. However, like any method that enables photo-realistic image generation, there is a possibility of malicious use, including the creation of deepfakes. It is important to note that our method has limitations when it comes to extreme gaze and head pose directions. These limitations are primarily due to the constraints posed by the label distributions in publicly available gaze datasets, and addressing this issue remains an open challenge.

In *Chapter 5*, we explore self-supervised learning to achieve improved representations for gaze estimation tasks. By using contrastive learning, we form positive and negative pairs from unlabeled multi-view camera images. We then leverage

the geometric consistency across these camera views to instill equivariance and invariance within the learned gaze representations. Before being put into the CNN encoder, the images underwent various augmentations. Subsequently, two-branch MLP networks were employed: one to induce invariance and the other to promote equivariance. These representations were then empirically evaluated across different domain datasets for the task of person-specific few-shot gaze estimation. Our method demonstrated enhanced performance in nearly all the proposed settings across various datasets and achieved better results compared to the state-of-the-art methods.

In *Chapter 6*, we introduced STAGE, a model designed to learn spatio-temporal attention for the task of video gaze estimation. STAGE incorporates two distinct attention modules: spatial and temporal attention. The spatial attention module, inspired by the change captioning task, is designed to detect local pixel changes between consecutive frames. Meanwhile, the temporal attention module aims to grasp the global dynamics of eye movements by analyzing past frames to predict the gaze direction of the current frame. We enhanced STAGE by personalizing it with Gaussian processes, learning an additive model to predict biases for the yaw and pitch angles of each frame. Additionally, it provides an uncertainty estimate for each frame's prediction, enabling the development of error-aware applications and the avoidance of frames with erroneous predictions.

7.2 Future Directions

In this section, we outline potential future directions to expand the work discussed in this thesis.

Truly self-supervised gaze estimation. Chapter 5 introduced a contrastive learning strategy for acquiring gaze representations via self-supervision. This approach is predicated on the assumption that the multi-view geometry is predetermined and the camera poses for all cameras are readily accessible. Consequently, while this method delivers satisfactory performance in both within-data and cross-data scenarios, the gaze representations it learns could be sensitive to variations in camera poses. An interesting future direction is to explore the special orthogonal group of dimension 3, $SO(3)$, to learn the rotation matrices between cameras. In a static camera setup, the relative rotations between cameras remain constant, and therefore, these rotations could be learned as part of the contrastive learning process such that $R \in SO(3)$. The properties of rotation matrices, such as $RR^T = I$ and $det(R) = 1$, could be incorporated into the overall training objective. Incorporating these constraints within the self-supervised learning model could significantly broaden the scope and yield a more robust gaze representation that is not affected by a particular multi-view camera setup.

Infusing self-supervision in videos. Self-supervised learning has shown remarkable performance using large-scale image datasets. The representations learned through this approach have surpassed those obtained via supervised learning in downstream tasks like gaze estimation [237]. Our exploration of contrastive learning for developing image-based gaze representations using multi-view image data has yielded promising results. Consequently, this self-supervised approach emerges as a viable solution for video datasets, effectively capturing eye movement dynamics without relying heavily on annotated gaze data. The field of video representation learning, through self-supervision, has been extensively explored across various video computer vision tasks [241]. Investigating self-supervision in conjunction with minimal supervision from video labeled gaze data can significantly improve the incorporation of eye movement knowledge into the learned representations.

Sparse video gaze estimation. Chapter 6 introduced a video gaze estimation technique that utilizes spatial attention to detect pixel changes between the current and past frames. These variations are quantified as 'change features,' indicating the level of change between two frames. These features act as a basis for identifying frames without significant changes from their preceding ones, thereby enabling the replication of the previous frame's gaze prediction for the current frame, rather than computing a new prediction. This approach can significantly boost the speed and reduce the computational load for real-time gaze estimation tasks.

Bibliography

[1] Elias Daniel Guestrin and Moshe Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133, 2006.

[2] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the symposium on eye tracking research and applications*, pages 255–258, 2014.

[3] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9368–9377, 2019.

[4] Yufeng Zheng, Seonwook Park, Xucong Zhang, Shalini De Mello, and Otmar Hilliges. Self-learning transformations for improving gaze and head redirec-

tion. *Advances in Neural Information Processing Systems*, 33:13127–13138, 2020.

[5] Zhaokang Chen and Bertram Shi. Offset calibration for appearance-based gaze estimation via gaze decomposition. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 270–279, 2020.

[6] Patrik Huber, Guosheng Hu, Rafael Tena, Pouria Mortazavian, P Koppen, William J Christmas, Matthias Ratsch, and Josef Kittler. A multiresolution 3d morphable face model and fitting framework. In *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016.

[7] Yu Yu and Jean-Marc Odobez. Unsupervised representation learning for gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7314–7324, 2020.

[8] Yunjia Sun, Jiabei Zeng, Shiguang Shan, and Xilin Chen. Cross-encoder for unsupervised gaze representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3702–3711, 2021.

[9] Dawei Yang, Xinlei Li, Xiaotian Dai, Rui Zhang, Lizhe Qi, Wenqiang Zhang, and Zhe Jiang. All in one network for driver attention monitoring. In

ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2258–2262. IEEE, 2020.

[10] Sumit Jha and Carlos Busso. Estimation of driver's gaze region from head position and orientation using probabilistic confidence regions. *IEEE Transactions on Intelligent Vehicles*, 8(1):59–72, 2022.

[11] Sayyed Mudassar Shah, Zhaoyun Sun, Khalid Zaman, Altaf Hussain, Muhammad Shoaib, and Lili Pei. A driver gaze estimation method based on deep learning. *Sensors*, 22(10):3959, 2022.

[12] Senuri De Silva, Sanuwani Dayarathna, Gangani Ariyarathne, Dulani Meedeniya, Sampath Jayarathna, Anne MP Michalek, and Gavindya Jayawardena. A rule-based system for adhd identification using eye movement data. In *2019 Moratuwa Engineering Research Conference (MERCon)*, pages 538–543. IEEE, 2019.

[13] Senuri De Silva, Sanuwani Dayarathna, Gangani Ariyarathne, Dulani Meedeniya, Sampath Jayarathna, and Anne MP Michalek. Computational decision support system for adhd identification. *International Journal of Automation and Computing*, 18(2):233–255, 2021.

[14] Mohamad A Eid, Nikolas Giakoumidis, and Abdulmotaleb El Saddik. A novel

eye-gaze-controlled wheelchair system for navigating unknown environments: case study with a person with als. *IEEE Access*, 4:558–573, 2016.

[15] Md Samiul Haque Sunny, Md Ishrak Islam Zarif, Ivan Rulik, Javier Sanjuan, Mohammad Habibur Rahman, Sheikh Iqbal Ahamed, Inga Wang, Katie Schultz, and Brahim Brahmi. Eye-gaze control of a wheelchair mounted 6dof assistive robot for activities of daily living. *Journal of NeuroEngineering and Rehabilitation*, 18(1):1–12, 2021.

[16] Zichen Kong, Shuying Rao, Hui Yang, Wenli Lan, Yue Leng, and Sheng Ge. Eye-tracking-based robotic arm control system. In *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, pages 663–667. IEEE, 2022.

[17] Craig A Chin, Armando Barreto, J Gualberto Cremades, and Malek Adjouadi. Integrated electromyogram and eye-gaze tracking cursor control system for computer users with motor disabilities. 2008.

[18] Geoffrey Underwood. *Cognitive processes in eye guidance*. Oxford University Press, 2005.

[19] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5396–5406, 2020.

[20] RG Vishnu Menon, Valdimar Sigurdsson, Nils Magne Larsen, Asle Fagerstrøm, and Gordon R Foxall. Consumer attention to price in social commerce: Eye tracking patterns in retail clothing. *Journal of Business Research*, 69 (11):5008–5013, 2016.

[21] Bridget K Behe, Patricia T Huddleston, Kevin L Childs, Jiaoping Chen, and Iago S Muraro. Seeing through the forest: The gaze path to purchase. *Plos one*, 15(10):e0240179, 2020.

[22] Carlos Bermejo, Dimitris Chatzopoulos, and Pan Hui. Eyeshopper: Estimating shoppers' gaze using cctv cameras. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2765–2774, 2020.

[23] Laurence R Young and David Sheena. Survey of eye movement recording methods. *Behavior research methods & instrumentation*, 7(5):397–429, 1975.

[24] Eduard Schott. Uber die registrierung des nystagmus und anderer augenbewegungen verm itteles des saitengalvanometers. *Deut Arch fur klin Med*, 140:79–90, 1922.

[25] OH Mowrer, Theodore C Ruch, and NE Miller. The corneo-retinal potential difference as the basis of the galvanometric method of recording eye movements. *American Journal of Physiology-Legacy Content*, 114(2):423–428, 1935.

[26] Craig Hennessey, Borna Noureddin, and Peter Lawrence. A single camera eye-gaze tracking system with free head motion. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 87–94, 2006.

[27] Stefania Cristina and Kenneth P Camilleri. Model-based head pose-free gaze estimation for assistive communication. *Computer Vision and Image Understanding*, 149:157–170, 2016.

[28] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 Chi conference on human factors in computing systems*, pages 1118–1130, 2017.

[29] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *arXiv preprint arXiv:2104.12668*, 2021.

[30] Kar-Han Tan, David J Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pages 191–195. IEEE, 2002.

[31] Maciej Nowakowski, Matthew Sheehan, Daniel Neal, and Alexander V Goncharov. Investigation of the isoplanatic patch and wavefront aberration along

the pupillary axis compared to the line of sight in the eye. *Biomedical optics express*, 3(2):240–258, 2012.

[32] David A Atchison, George Smith, and George Smith. *Optics of the human eye*, volume 2. Butterworth-Heinemann Oxford, 2000.

[33] Samuel Arba Mosquera, Shwetabh Verma, and Colm McAlinden. Centration axis in refractive surgery. *Eye and Vision*, 2(1):1–16, 2015.

[34] Roger HS Carpenter. *Movements of the Eyes, 2nd Rev.* Pion Limited, 1988.

[35] Laura Chamberlain. Eye tracking methodology; theory and practice. *Qualitative Market Research: An International Journal*, 2007.

[36] Dan Witzner Hansen and Riad I Hammoud. An improved likelihood model for eye tracking. *Computer Vision and Image Understanding*, 106(2-3): 220–230, 2007.

[37] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2009.

[38] Zhiwei Zhu and Qiang Ji. Eye and gaze tracking for interactive graphic display. *Machine Vision and Applications*, 15(3):139–148, 2004.

[39] Zhiwei Zhu and Qiang Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1):124–154, 2005.

[40] Pieter Blignaut. Mapping the pupil-glint vector to gaze coordinates in a simple video-based eye tracker. *Journal of Eye Movement Research*, 7(1), 2014.

[41] Zhiwei Zhu and Qiang Ji. Novel eye gaze tracking techniques under natural head movement. *IEEE TRANSACTIONS on biomedical engineering*, 54(12): 2246–2260, 2007.

[42] Arantxa Villanueva, Rafael Cabeza, and Sonia Porta. Eye tracking: Pupil orientation geometrical modeling. *Image and Vision Computing*, 24(7): 663–679, 2006.

[43] Arantxa Villanueva, Rafael Cabeza, and Sonia Porta. Gaze tracking system model based on physical parameters. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(05):855–877, 2007.

[44] Jian-Gang Wang, Eric Sung, and Ronda Venkateswarlu. Estimating the eye gaze from one eye. *Computer Vision and Image Understanding*, 98(1): 83–103, 2005.

[45] K Preston White, Thomas E Hutchinson, and Janine M Carley. Spatially dynamic calibration of an eye-tracking system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):1162–1168, 1993.

[46] Borna Noureddin, Peter D Lawrence, and CF Man. A non-contact device for tracking gaze in a human computer interface. *Computer Vision and Image Understanding*, 98(1):52–82, 2005.

[47] Marcio RM Mimica and Carlos Hitoshi Morimoto. A computer vision framework for eye gaze tracking. In *16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)*, pages 406–412. IEEE, 2003.

[48] Z.R. Cherif, A. Nait-Ali, J.F. Motsch, and M.O. Krebs. An adaptive calibration of an infrared light device used for gaze tracking. In *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276)*, volume 2, pages 1029–1033 vol.2, 2002. doi: 10.1109/IMTC.2002.1007096.

[49] Juan J Cerrolaza, Arantxa Villanueva, and Rafael Cabeza. Taxonomic study of polynomial regressions applied to the calibration of video-oculographic systems. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 259–266, 2008.

[50] Erna Demjén, V Abosi, and Zoltán Tomori. Eye tracking using artificial neural networks for human computer interaction. *Physiological research*, 60 (5):841, 2011.

[51] Massimo Gneo, Maurizio Schmid, Silvia Conforto, and Tommaso D'Alessio. A free geometry model-independent neural eye-gaze tracking system. *Journal of neuroengineering and rehabilitation*, 9(1):1–15, 2012.

[52] Yi-Leh Wu, Chun-Tsai Yeh, Wei-Chih Hung, and Cheng-Yuan Tang. Gaze direction estimation using support vector machine with active appearance model. *Multimedia tools and applications*, 70:2037–2062, 2014.

[53] Jianzhong Wang, Guangyue Zhang, and Jiadong Shi. 2d gaze estimation based on pupil-glint vector using an artificial neural network. *Applied Sciences*, 6(6):174, 2016.

[54] Laura Sesma, Arantxa Villanueva, and Rafael Cabeza. Evaluation of pupil center-eye corner vector for gaze estimation using a web cam. In *Proceedings of the symposium on eye tracking research and applications*, pages 217–220, 2012.

[55] Yanxia Zhang, Andreas Bulling, and Hans Gellersen. Sideways: A gaze interface for spontaneous interaction with situated displays. In *Proceedings*

of the SIGCHI Conference on Human Factors in Computing Systems, pages 851–860, 2013.

[56] Michael Xuelin Huang, Tiffany CK Kwok, Grace Ngai, Hong Va Leong, and Stephen CF Chan. Building a self-learning eye gaze model from user interaction data. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1017–1020, 2014.

[57] Yanxia Zhang, Jörg Müller, Ming Ki Chong, Andreas Bulling, and Hans Gellersen. Gazehorizon: Enabling passers-by to interact with public displays by gaze. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 559–563, 2014.

[58] Michael Xuelin Huang, Tiffany CK Kwok, Grace Ngai, Stephen CF Chan, and Hong Va Leong. Building a personalized, auto-calibrating eye tracker from user interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5169–5179, 2016.

[59] Kenneth A Funes-Mora and Jean-Marc Odobez. Gaze estimation in the 3d space using rgb-d sensors. *International Journal of Computer Vision*, 118 (2):194–216, 2016.

[60] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze:

dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28(5):445–461, 2017.

[61] Inessa Bekerman, Paul Gottlieb, and Michael Vaiman. Variations in eyeball diameters of the healthy adults. *Journal of ophthalmology*, 2014, 2014.

[62] John V Forrester, Andrew D Dick, Paul G McMenamin, Fiona Roberts, and Eric Pearlman. *The eye e-book: basic sciences in practice*. Elsevier Health Sciences, 2020.

[63] André Meyer, Martin Böhme, Thomas Martinetz, and Erhardt Barth. A single-camera remote eye tracker. In *Perception and Interactive Technologies: International Tutorial and Research Workshop, PIT 2006 Kloster Irsee, Germany, June 19-21, 2006. Proceedings*, pages 208–211. Springer, 2006.

[64] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[65] Jian-Gang Wang and Eric Sung. Gaze determination via images of irises. *Image and Vision Computing*, 19(12):891–911, 2001.

[66] Kosuke Takahashi, Shohei Nobuhara, and Takashi Matsuyama. A new mirror-based extrinsic camera calibration using an orthogonality constraint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1051–1058. IEEE, 2012.

[67] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 207–210, 2014.

[68] Li Jianfeng and Li Shigang. Eye-model-based gaze estimation by rgb-d camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 592–596, 2014.

[69] Xuehan Xiong, Zicheng Liu, Qin Cai, and Zhengyou Zhang. Eye gaze tracking using an rgbd camera: a comparison with a rgb solution. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1113–1121, 2014.

[70] Kang Wang and Qiang Ji. Hybrid model and appearance based eye tracking with kinect. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 331–332, 2016.

[71] Hirotake Yamazoe, Akira Utsumi, Tomoko Yonezawa, and Shinji Abe. Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 245–250, 2008.

[72] Kang Wang and Qiang Ji. Real time eye gaze tracking with 3d deformable

eye-face model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1003–1011, 2017.

[73] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. A 3d morphable eye region model for gaze estimation. In *European Conference on Computer Vision*, pages 297–313. Springer, 2016.

[74] Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 1–10, 2018.

[75] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015.

[76] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.

[77] K Simonyan and A Zisserman. Very deep convolutional networks for large-

scale image recognition. *3rd International Conference on Learning Representations*, pages 1–14, 2015.

[78] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–60, 2017.

[79] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[80] Yihua Cheng, Xucong Zhang, Feng Lu, and Yoichi Sato. Gaze estimation by exploring two-eye asymmetry. *IEEE Transactions on Image Processing*, 29: 5259–5272, 2020.

[81] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In *European Conference on Computer Vision*, pages 339–357, September 2018.

[82] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018.

[83] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. In *AAAI*, pages 10623–10630, 2020.

[84] Wangjiang Zhu and Haoping Deng. Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3143–3152, 2017.

[85] LRD Murthy and Pradipta Biswas. Appearance-based gaze estimation using attention and difference mechanism. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3137–3146. IEEE, 2021.

[86] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 271–280, 2013.

[87] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 365–381. Springer, 2020.

[88] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6912–6921, 2019.

[89] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. Towards end-to-end video-based eye-tracking. In *European Conference on Computer Vision*, pages 747–763. Springer, 2020.

[90] Daniil Kononenko and Victor Lempitsky. Learning to look up: Realtime monocular gaze correction using machine learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4667–4675, 2015.

[91] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European conference on computer vision*, pages 311–326. Springer, 2016.

[92] Yu Yu, Gang Liu, and Jean-Marc 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*, pages 11937–11946, 2019.

[93] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson,

and Andreas Bulling. Gazedirector: Fully articulated eye gaze redirection in video. In *Computer Graphics Forum*, volume 37, pages 217–225. Wiley Online Library, 2018.

[94] Jingjing Chen, Jichao Zhang, Enver Sangineto, Tao Chen, Jiayuan Fan, and Nicu Sebe. Coarse-to-fine gaze redirection with numerical and pictorial guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3665–3674, 2021.

[95] Zhe He, Adrian Spurr, Xucong Zhang, and Otmar Hilliges. Photo-realistic monocular gaze redirection using generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6932–6941, 2019.

[96] Harsimran Kaur and Roberto Manduchi. Subject guided eye image synthesis with application to gaze redirection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 11–20, 2021.

[97] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[98] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-

encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[99] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Interpretable transformations with encoder-decoder networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5726–5735, 2017.

[100] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Wensen Feng. Controllable continuous gaze redirection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1782–1790, 2020.

[101] Shiwei Jin, Zhen Wang, Lei Wang, Ning Bi, and Truong Nguyen. Redirtrans: Latent-to-latent translation for gaze and head redirection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5547–5556, June 2023.

[102] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106, 2021.

[103] Alessandro Ruzzi, Xiangwei Shi, Xi Wang, Gengyan Li, Shalini De Mello, Hyung Jin Chang, Xucong Zhang, and Otmar Hilliges. Gazenerf: 3d-aware

gaze redirection with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9676–9685, June 2023.

[104] John Gideon, Shan Su, and Simon Stent. Unsupervised multi-view gaze representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5009, 2022.

[105] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[106] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent cnn for 3d gaze estimation using appearance and shape cues. *29th British Machine Vision Conference, BMVC*, 2018.

[107] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer, 2005.

[108] Kang Wang, Hui Su, and Qiang Ji. Neuro-inspired eye tracking with eye movement dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9831–9840, 2019.

[109] Gang Liu, Yu Yu, Kenneth Alberto Funes Mora, and Jean-Marc Odobez. A differential approach for gaze estimation with calibration. In *BMVC*, volume 2, page 6, 2018.

[110] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[111] Dietmar Ott, Scott H Seidman, and R John Leigh. The stability of human eye orientation during visual fixation. *Neuroscience letters*, 142(2):183–186, 1992.

[112] Murat Aytekin, Jonathan D Victor, and Michele Rucci. The visual input to the retina during natural head-free fixation. *Journal of Neuroscience*, 34 (38):12701–12715, 2014.

[113] Siegfried Wahl, Denitsa Dragneva, and Katharina Rifai. The limits of fixation—keeping the ametropic eye on target. *Journal of vision*, 19(13):8–8, 2019.

[114] RJ Leigh, SE Thurston, RL Tomsak, GE Grossman, and DJ Lanska. Effect of monocular visual loss upon stability of gaze. *Investigative ophthalmology & visual science*, 30(2):288–292, 1989.

[115] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138, 2016.

[116] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.

[117] A. Kar and P. Corcoran. A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access*, 5:16495–16519, 2017.

[118] Veronica Sundstedt. Gazing at games: An introduction to eye tracking control. *Synthesis Lectures on Computer Graphics and Animation*, 5(1): 1–113, 2012.

[119] Thammathip Piumsomboon, Gun Lee, Robert W Lindeman, and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 36–39. IEEE, 2017.

[120] Mohamed Khamis, Axel Hoesl, Alexander Klimczak, Martin Reiss, Florian

Alt, and Andreas Bulling. Eyescout: Active eye tracking for position and movement independent gaze interaction with large public displays. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 155–166, 2017.

[121] Yixuan Li, Pingmei Xu, Dmitry Lagun, and Vidhya Navalpakkam. Towards measuring and inferring user interest from gaze. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 525–533, 2017.

[122] David Thomson. Eye tracking and its clinical application in optometry. *Optician*, 2017(6):6045–1, 2017.

[123] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Evaluation of appearance-based methods and implications for gaze-based applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[124] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, 2014.

[125] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eye-

diap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*. ACM, March 2014. doi: 10.1145/2578153.2578190.

[126] 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*, 41(1):162–175, 2019.

[127] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 1–9, 2018.

[128] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 951–959, 2017.

[129] Jiankang Deng, Yuxiang Zhou, Shiyang Cheng, and Stefanos Zaferiou. Cascade multi-view hourglass model for robust 3d face alignment. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 399–403. IEEE, 2018.

[130] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and vision computing*, 28(5):807–813, 2010.

[131] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[132] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[133] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[134] Rui Rodrigues, Joao P Barreto, and Urbano Nunes. Camera pose estimation using images of planar mirror reflections. In *European Conference on Computer Vision*, pages 382–395. Springer, 2010.

[135] Anjul Patney, Joohwan Kim, Marco Salvi, Anton Kaplanyan, Chris Wyman, Nir Benty, Aaron Lefohn, and David Luebke. Perceptually-based foveated virtual reality. In *ACM SIGGRAPH 2016 Emerging Technologies*, pages 1–2. 2016.

[136] Thies Pfeiffer. Towards gaze interaction in immersive virtual reality: Evaluation of a monocular eye tracking set-up. In *Virtuelle und Erweiterte Realität-Fünfter Workshop der GI-Fachgruppe VR/AR*, 2008.

[137] Päivi Majaranta and Andreas Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, pages 39–65. Springer, 2014.

[138] Robert JK Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The mind's eye*, pages 573–605. Elsevier, 2003.

[139] Guy Thomas Buswell. How people look at pictures: a study of the psychology and perception in art. 1935.

[140] Constantin A Rothkopf, Dana H Ballard, and Mary M Hayhoe. Task and context determine where you look. *Journal of vision*, 7(14):16–16, 2007.

[141] Ryo Ishii, Kazuhiro Otsuka, Shiro Kumano, and Junji Yamato. Prediction of who will be the next speaker and when using gaze behavior in multiparty meetings. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 6 (1):1–31, 2016.

[142] Catharine Oertel, Kenneth A Funes Mora, Joakim Gustafson, and Jean-Marc Odobez. Deciphering the silent participant: On the use of audio-visual cues

for the classification of listener categories in group discussions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 107–114, 2015.

[143] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[144] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[145] Ben Usman, Nick Dufour, Kate Saenko, and Chris Bregler. Puppetgan: Cross-domain image manipulation by demonstration. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9449–9457, 2019. doi: 10.1109/ICCV.2019.00954.

[146] Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. In *European Conference on Computer Vision (ECCV)*, 2020.

[147] Yaoming Wang, Yangzhou Jiang, Jin Li, Bingbing Ni, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Teng Li. Contrastive regression for domain adaptation on gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19376–19385, 2022.

[148] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[149] Marco Toldo, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. Unsupervised domain adaptation in semantic segmentation: a review. *Technologies*, 8(2):35, 2020.

[150] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. *Advances in neural information processing systems*, 29, 2016.

[151] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a face: Towards arbitrary high fidelity face manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10042, 2019.

[152] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9459–9468, 2019.

[153] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

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

[155] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[156] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[157] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018.

[158] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[159] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8867–8876, 2018.

[160] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019.

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

[162] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classifica-

tion with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[163] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[164] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[165] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6924–6932, 2017.

[166] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[167] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.

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

[169] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

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

[171] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[172] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[173] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[174] Shreya Ghosh, Abhinav Dhall, Munawar Hayat, Jarrod Knibbe, and Qiang Ji. Automatic gaze analysis: A survey of deep learning based approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1): 61–84, 2023.

[175] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[176] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[177] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[178] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[179] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3412–3420, 2019.

[180] Takayasu Moriya, Holger R Roth, Shota Nakamura, Hirohisa Oda, Kai Nagara, Masahiro Oda, and Kensaku Mori. Unsupervised segmentation of 3d medical images based on clustering and deep representation learning. In *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 10578, page 1057820. International Society for Optics and Photonics, 2018.

[181] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[182] Tobii Pro AB. Tobii pro lab. Computer software, 2014. URL http://www.tobiipro.com/.

[183] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[184] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representa-

tions by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[185] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

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

[187] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[188] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[189] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[190] Xinlei Chen and Kaiming He. Exploring simple siamese representation

learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[191] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022.

[192] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–767, 2018.

[193] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11230–11239, 2021.

[194] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[195] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with

warm restarts. In *International Conference on Learning Representations*, 2017.

[196] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[197] AJung Moon, Daniel M Troniak, Brian Gleeson, Matthew KXJ Pan, Minhua Zheng, Benjamin A Blumer, Karon MacLean, and Elizabeth A Croft. Meet me where i'm gazing: how shared attention gaze affects human-robot handover timing. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 334–341, 2014.

[198] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5048–5054. IEEE, 2016.

[199] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35 (6):1–12, 2016.

[200] Nitish Padmanaban, Robert Konrad, Tal Stramer, Emily A Cooper, and Gordon Wetzstein. Optimizing virtual reality for all users through gaze-

contingent and adaptive focus displays. *Proceedings of the National Academy of Sciences*, 114(9):2183–2188, 2017.

[201] Dmitry Rudoy, Dan B Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1147–1154, 2013.

[202] Daniel Parks, Ali Borji, and Laurent Itti. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision research*, 116:113–126, 2015.

[203] Erik Lindén, Jonas Sjostrand, and Alexandre Proutiere. Learning to personalize in appearance-based gaze tracking. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[204] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[205] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[206] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41 (11):2740–2755, 2018.

[207] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[208] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

[209] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 971–980, 2017.

[210] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[211] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar

Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[212] Xuanhan Wang, Lianli Gao, Peng Wang, Xiaoshuai Sun, and Xianglong Liu. Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length. *IEEE Transactions on Multimedia*, 20(3):634–644, 2017.

[213] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[214] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[215] Jun Li, Xianglong Liu, Mingyuan Zhang, and Deqing Wang. Spatio-temporal deformable 3d convnets with attention for action recognition. *Pattern Recognition*, 98:107037, 2020.

[216] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4624–4633, 2019.

[217] Yue Qiu, Shintaro Yamamoto, Kodai Nakashima, Ryota Suzuki, Kenji Iwata, Hirokatsu Kataoka, and Yutaka Satoh. Describing and localizing multiple changes with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1971–1980, 2021.

[218] Yunbin Tu, Tingting Yao, Liang Li, Jiedong Lou, Shengxiang Gao, Zhengtao Yu, and Chenggang Yan. Semantic relation-aware difference representation learning for change captioning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 63–73, 2021.

[219] Zhaokang Chen and Bertram E. Shi. Offset calibration for appearance-based gaze estimation via gaze decomposition. *2020 IEEE Winter Conference on Applications of Computer Vision*, pages 259–268, 2019.

[220] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[221] Yanxia Zhang, Ken Pfeuffer, Ming Ki Chong, Jason Alexander, Andreas Bulling, and Hans Gellersen. Look together: using gaze for assisting co-located collaborative search. *Personal and Ubiquitous Computing*, 21:173–186, 2017.

[222] Julius Albiz, Olga Viberg, and Andrii Matviienko. Guiding visual attention

on 2d screens: Effects of gaze cues from avatars and humans. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*, pages 1–9, 2023.

[223] Zi Wang, George E Dahl, Kevin Swersky, Chansoo Lee, Zelda Mariet, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained gaussian processes for bayesian optimization. *arXiv preprint arXiv:2109.08215*, 2021.

[224] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[225] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[226] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[227] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[228] Radford M. Neal. Assessing relevance determination methods using delve. 1998. URL https://api.semanticscholar.org/CorpusID:59749732.

[229] Ian A. Delbridge, David S. Bindel, and Andrew Gordon Wilson. Randomly projected additive gaussian processes for regression. In *International Conference on Machine Learning*, 2019.

[230] Muhammad Shafiq and Zhaoquan Gu. Deep residual learning for image recognition: a survey. *Applied Sciences*, 12(18):8972, 2022.

[231] Swati Jindal and Roberto Manduchi. Contrastive representation learning for gaze estimation. In *Annual Conference on Neural Information Processing Systems*, pages 37–49. PMLR, 2023.

[232] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision*, pages 3–19, 2018.

[233] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[234] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[235] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and

beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021.

[236] Ashesh Mishra and Hsuan-Tien Lin. 360-degree gaze estimation in the wild using multiple zoom scales. In *British Machine Vision Conference*, 2020.

[237] Arya Farkhondeh, Cristina Palmero, Simone Scardapane, and Sergio Escalera. Towards self-supervised gaze estimation. *arXiv preprint arXiv:2203.10974*, 2022.

[238] Ahmed A. Abdelrahman, Thorsten Hempel, Aly Khalifa, Ayoub Al-Hamadi, and Laslo Dinges. L2cs-net : Fine-grained gaze estimation in unconstrained environments. In *2023 8th International Conference on Frontiers of Signal Processing (ICFSP)*, pages 98–102, 2023.

[239] Yihua Cheng and Feng Lu. Gaze estimation using transformer. In *2022 26th International Conference on Pattern Recognition*, pages 3341–3347. IEEE, 2022.

[240] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. Adaptive linear regression for appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):2033–2046, 2014.

[241] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s):1–37, 2023.