Title

Learning-Based Facial Attribute Estimation and Manipulation

Author

Jin, Shiwei

Publication Date

2024

UNIVERSITY OF CALIFORNIA SAN DIEGO

Learning-Based Facial Attribute Estimation and Manipulation

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

in

Electrical Engineering (Signal and Image Processing)

by

Shiwei Jin

Committee in charge:

Professor Truong Nguyen, Chair
Professor Cheolhong An
Professor Pamela Cosman
Professor Ravi Ramamoorthi

2024

The dissertation of Shiwei Jin is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

To my parents, X.W., and Lychee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

My time at UCSD has been an unforgettable journey that I will always cherish. I am deeply grateful to everyone who has contributed to making my experience here so meaningful and memorable.

First and foremost, I extend my deepest appreciation to Professor Truong Q. Nguyen, my Ph.D. advisor, for his continuous support and mentorship throughout my journey at UCSD. It all began back in August 2019 during the Summer Research Internship Program when he generously offered me the opportunity to pursue a Ph.D., a privilege I deeply treasure. Professor Nguyen's mentorship has been invaluable, guiding me in shaping thesis topics, improving technical writing, refining research presentations, and fostering critical thinking skills. Under his guidance, I have experienced significant professional growth. I am also immensely grateful to Professor Cheolhong An, Professor Pamela Cosman, and Professor Ravi Ramamoorthi for serving as my committee members. Their insights and suggestions were crucial in completing my Ph.D. journey.

Furthermore, I want to express my gratitude to the Qualcomm multimedia team for their support and guidance. Zhen Wang, Peng Liu, Lei Wang, and Ning Bi provided invaluable insights and ideas that significantly contributed to both my Ph.D. projects and summer internship. I am especially grateful to Zhen and Lei, who taught me to adopt an application-oriented engineer's perspective in research. Their generous sharing of scientific experiences and consistent feedback at every step of the way were significant in my growth as a researcher.

I also want to express my deep appreciation for another internship experience at Sony, which was incredibly valuable and memorable due to the collaborative projects with Jong Hwa Lee, Matthew Wnuk, Allison Alvarado, Ed Winter, and Gary Lyons. Their support and the freedom they gave me allowed me to delve into real-world problem-solving. I am particularly grateful to Jong for his insightful feedback and innovative ideas, which significantly expanded the scope of my research during the internship.

I am deeply grateful for the support and collaboration of my fellow lab mates at the

Video Processing Lab, whose seamless coordination and insightful discussions have played a crucial role in my research growth. Specifically, I would like to extend my heartfelt thanks to Ji Dai, Chen Du, and Le Dinh An for their exceptional coordination and consistent support throughout our collaborative projects. Moreover, I extend my appreciation to Abdullah Albattal, Junkang Zhang, Yiqian Wang, Bang Du, Kunyao Chen, Runfa Li, and Haochen Zhang for their active participation and insightful discussions. Their diverse perspectives and contributions have enriched our research endeavors and led to fruitful outcomes.

Finally, my deepest gratitude goes to my family. My parents' unwavering encouragement has been a cornerstone of strength throughout my journey. Their unconditional support, both financially and emotionally, has provided me with the foundation to successfully complete my Ph.D. Their belief in me has been a constant source of motivation and determination. I also want to extend a special thank you to my girlfriend, X.W., who has always been my first audience along the journey. Her unwavering companionship has been instrumental in helping me navigate and overcome various difficulties and challenges I have encountered. Her support has been a source of inspiration and resilience, and I am truly grateful for her presence in my life.

To all other professors, colleagues, friends, and family members not mentioned here, I extend my heartfelt thanks. This chapter of my life will forever hold a special place in my heart.

Chapter 2, in full, is a reprint of the material as it appears in the publication of "Kappa Angle Regression with Ocular Counter-Rolling Awareness for Gaze Estimation", Shiwei Jin, Ji Dai, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in the publication of "ReDirTrans: Latent-to-Latent Translation for Gaze and Head Redirection", Shiwei Jin, Zhen Wang, Lei Wang, Ning Bi, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in the publication of “AUEditNet: Dual-Branch Facial Action Unit Intensity Manipulation with Implicit Disentanglement,”, Shiwei Jin, Zhen Wang, Lei Wang, Peng Liu, Ning Bi, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2024. The dissertation author was the primary investigator and author of this paper.

VITA

2018 B. S. in Opto-Electronics Information Science and Engineering, Zhejiang University, Hangzhou, China

2020 M. S. in Electrical Engineering (Signal and Image Processing), University of California San Diego, CA

2024 Ph. D. in Electrical Engineering (Signal and Image Processing), University of California San Diego, CA

PUBLICATIONS

S. Jin, Z. Wang, L. Wang, P. Liu, N. Bi and T. Nguyen, “AUEditNet: Dual-Branch Facial Action Unit Intensity Manipulation with Implicit Disentanglement,” 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2024.

S. Jin, Z. Wang, L. Wang, N. Bi and T. Nguyen, “ReDirTrans: Latent-to-Latent Translation for Gaze and Head Redirection,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 5547-5556.

S. Jin, J. Dai and T. Nguyen, “Kappa Angle Regression with Ocular Counter-Rolling Awareness for Gaze Estimation,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 2023, pp. 2659-2668.

A. D. Le, **S. Jin**, Y. S. Bae and T. Nguyen, “A Novel Learnable Orthogonal Wavelet Unit Neural Network with Perfection Reconstruction Constraint Relaxation for Image Classification,” 2023 IEEE International Conference on Visual Communications and Image Processing (VCIP), Jeju, Korea, Republic of, 2023, pp. 1-5.

S. Jin, M. Vo, C. Du, H. Garudadri, A. Py, D. J. Moore, K. M. Erlandson, R. C. Moore, T. Nguyen, “Unsupervised Sequence Alignment between Video and Human Center of Pressure,” 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico, 2021, pp. 1980-1984.

S. Jin, J. Dai and T. Nguyen, “Differential Gaze Estimation with Ocular Counter-Rolling Compensation,” 2021 18th International SoC Design Conference (ISOCC), Jeju Island, Korea, Republic of, 2021, pp. 23-24.

J. Dai, **S. Jin**, J. Zhang and T. Q. Nguyen, “Boosting Feature Matching Accuracy With Pairwise Affine Estimation,” in IEEE Transactions on Image Processing, vol. 29, pp. 8278-8291, 2020.

C. Du, S. Graham, **S. Jin**, C. Depp and T. Nguyen, “Multi-Task Center-Of-Pressure Metrics Estimation from Skeleton Using Graph Convolutional Network,” 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 2313-2317.

ABSTRACT OF THE DISSERTATION

Learning-Based Facial Attribute Estimation and Manipulation

by

Shiwei Jin

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2024

Professor Truong Nguyen, Chair

Facial attribute analysis plays a crucial role in various fields such as surveillance, entertainment, healthcare, and human-computer interaction. The advert of deep neural networks has sparked a growing interest in learning-based facial attribute analysis. This dissertation focuses on learning-based facial attribute analysis, encompassing facial attribute estimation and manipulation tasks. We focused on solving challenges including addressing data scarcity in supervised facial attribute estimation, handling facial attribute manipulation in high-resolution images, efficiently disentangling targeted attributes from others, and training a facial attribute manipulator with datasets from a small number of subjects.

The dissertation is structured around three key facial attribute categories: head orientations, gaze directions, and facial action units. The first part delves into improving appearance-based gaze estimation by considering person-dependent anatomical variations and accounting for ocular countering-rolling (OCR) responses, resulting in a more efficient and accurate method. The second part introduces ReDirTrans, a portable network designed for gaze redirection in high-resolution face images. By focusing on latent-to-latent translation, ReDirTrans enables precise gaze and head pose redirection while preserving other attributes, expanding its applicability beyond limited ranges of faces. The final part presents AUEditNet, a model for manipulating facial action unit intensities. This addresses challenges posed by data scarcity by effectively disentangling attributes and identity within a limited subject pool. AUEditNet demonstrates superior accuracy in editing AU intensities across 12 AUs, showcasing its potential for fine-grained facial attribute manipulation.

Overall, this dissertation contributes novel methodologies in learning-based facial attribute analysis, paving the way for enhanced performance and versatility across various real-world applications.

Chapter 1

Introduction

Facial attributes refer to the various features and characteristics of a person's face, which are essential indicators applied across diverse fields like surveillance, entertainment, healthcare, automobile, and human–computer interaction. The advancement of deep neural networks has fueled growing interest in learning-based facial attribute analysis due to their superior estimation capabilities.

Learning-based facial attribute analysis involves two main tasks [1]: facial attribute estimation and facial attribute manipulation. Facial attribute estimation predicts specific facial attributes from whole or partial face images. Facial attribute manipulation edits aimed facial attributes while preserving others. These two tasks complement each other. The manipulation effectively disentangles the aimed attributes from others, working as features distillation for estimation. While a well-trained estimator enhances manipulation performance by serving as a pretrained loss to evaluate edited facial attributes in synthesized face images.

Supervised facial attribute estimation typically demands extensive labeled data for the targeted attributes, which can be time-consuming and sometimes require expertise for accurate annotation. Conditional facial attribute manipulation provides an alternative solution to address annotated data scarcity. The edited images resulting from conditional manipulation, paired with corresponding conditions, augment the raw dataset and potentially improve the performance of the facial attribute estimation model. Moreover, as generators produce increasingly realistic images, it opens up new possibilities for leveraging facial attribute manipulation in creative endeavors, enhancing storytelling, character development in various media forms.

In this dissertation, we focus on three categories of the facial attributes: head orientations, gaze directions, and facial action units.

1.1 Background

1.1.1 Human Gaze

Definition. The human gaze indicates the direction a person is looking in. The optical axis of the eyeball is an theoretical line that passes through the the eyeball center and pupil center, which is associated with the observed iris [2]. The gaze direction and the optical axis are misaligned and the angle between them is termed as the Kappa Angle ($\boldsymbol{\kappa}$). This angle is a person-dependent anatomical variable that is not observable from external appearance. The details are illustrated in Fig. 2.1.

Gaze Estimation. There are three categories of gaze estimation methods: 2D gaze estimation, gaze following and 3D gaze estimation. 2D gaze estimation attempts to estimate the 2D points of the intersection between the gaze directions and the calibrated objects, such as glasses and screens of portable devices. But, the prerequisite for devices restricts the generalized ability of 2D gaze estimation model among different devices. Gaze following aims at predicting the gazing objects by the estimated saliency maps in an image or videos. But gaze following methods tend to predict the gazing target given the head orientations instead of the gaze directions, which yields inaccurate performance of cases with large or extreme gaze directions. 3D gaze estimation focuses on retrieving the 3D sight direction of eyes, which can be further classified into two main categories: geometric-based methods and appearance-based methods. Geometric-based methods predict gaze directions based on the geometric eye model whose parameters are estimated given the extracted eyes' features such as eye corners, corneal reflection and iris centers [3, 4, 5, 6, 7]. These methods usually require high resolution eye images, time-consuming personal calibration and near frontal head orientations, which can achieve more precise estimation at the cost of application ranges. Appearance-based methods [8, 9, 10, 11, 12] directly learn a mapping function from eye appearance images to the gaze directions without requirements of camera calibration and the geometric model. With the development of the deep neural network and techniques, appearance-based methods enjoyed significant improvements (more accurate and more efficient) for in-the-wild settings, which further speeds up gaze-related downstream applications.

Gaze Redirection. Methods for redirecting gaze directions can be classified into two main categories: warping-based methods and generator-based methods. Ganin *et al.* [13] proposed Deepwarp which learned warping fields to rearrange the pixels' locations for gaze redirection given the input images. Yu *et al.* [14] introduced a cycle pipeline with semantic segmentation consistency to supervise warping-field-based gaze redirection and they [15] further extended the warping-field-based methods with an unsupervised learning strategy by representation learning. These warping-based methods can maintain the original content very well since the pixel values in the redirected images are interpolated given the original ones. However, such methods cannot synthesize the change of lighting conditions or extreme gaze directions and head orientations [16, 17]. Besides warping-field-based methods, He *et al.* [18] first applied the Generative Adversarial Network to the gaze redirection task, generating photo-realistic eye images with desired gaze directions. Park *et al.* [16] proposed FAZE: an encoder-decoder structure to change gaze directions and head orientations in feature space with desired labels. Zheng *et al.* [17] presented ST-ED, which first advanced the encoder-decoder method from eye images to face images. However, ST-ED focused on generating 128×128 images with the restricted face range (no hair area), which limited the potential applications of gaze and head redirection.

1.1.2 Facial Action Units

Definition. A facial action unit (AU) is a specific movement or combination of movements of the facial muscles that produce a distinct facial expression. These AUs are objectively quantified on a six-level ordinal scale as defined by the Facial Action Coding System (FACS) [19]. A relatively small number of AUs (30) have ability to generate over $7,000$ observed facial expressions through various combinations.

AU Intensity Manipulation. Due to the scarcity of public AU-annotated datasets and the limited number of subjects included, AU intensity manipulation methods often rely on the pretrained AU intensity estimator [20] to increase the available training data. The estimator is used to estimate AU intensities on public face datasets with much larger subject pools [21, 22, 23], and the estimated AU intensities are then utilized as the ground truth. This process ensures the disentanglement of AU-related features from others based on a large amount of data, which is a common solution used in state-of-the-art AU intensity manipulation techniques to address the challenge of data scarcity in this field. Among these AU intensity manipulation methods, GANimation [24] was an early attempt to use AU intensities as conditions for facial expression manipulation. However, it faced attention mechanism issues, leading to overlap artifacts in areas of facial deformation [25]. Ling *et al.* [26] proposed a method using relative AU intensities between source and target images as conditions, avoiding the direct addition of new attributes onto existing expressions [27]. Alternatively, ICface [27] introduced a two-stage editing process. The first stage transforms the input image into a neutral state with all AU intensities set to zero. The second stage maps this neutral state to the final output, incorporating the desired driving attributes with two independent generators.

1.1.3 Face Image Editing

GAN Inversion. GAN inversion is the process of estimating latent codes that enable the faithful reconstruction of the given image via a pretrained generator. There are two main types of GAN inversion methods. The first type optimizes latent codes directly until they can reconstruct the image, which produces high-quality results but can be time-consuming because it requires optimization for each image. The second type trains an encoder to work with the pretrained generator. The encoder converts the image into latent codes, which the generator then uses to reconstruct the image. This encoder-based type balances processing time and reconstruction quality effectively. Richardson *et al.* introduced the multi-level structure encoder pSp [28], which corresponds to the multi-level structure of StyleGAN [29] to match coarse to fine features in style. Tov *et al.* proposed e4e [30] for both high-quality image reconstruction and efficient editing of latent codes, using a residual addition process. Alaluf *et al.* introduced ReStyle [31] with a self-correcting method to estimate residuals in inverted latent codes, enhancing inversion quality through iterative updates. Additionally, Alaluf *et al.* presented HyperStyle [32] to modulate StyleGAN's weights for better reconstruction quality.

Latent Space Manipulation. Editing facial attributes directly at the pixel level is highly challenging and time-exhausted due to the entangled nature of multiple facial attributes in the image space. Instead, many researchers place their focus to working in the latent space of generative models, particularly leveraging the well-disentangled latent space of StyleGAN [29] for image editing. Supervised methods like InterFaceGAN [33] determine hyperplanes for facial attribute editing based on provided labels. StyleFlow [34] maps a sample from a prior distribution to a latent distribution conditioned on target attributes estimated by pretrained attribute classifiers. Unsupervised methods such as GANSpace [35], SeFa [36], and TensorGAN [37] use principal components analysis, eigenvector decomposition, and higher-order singular value decomposition to discover semantic directions in the latent space, respectively. Other self-supervised methods involve mixing latent codes from other samples for local editing [38, 39] or incorporate language models like CLIP [40] for text-driven editing [41].

1.2 Contribution

Our objective in this dissertation has been carried out into three parts: 1) Appearance-based gaze estimation with considering the unobserved personal variations; 2) Gaze and head redirection in high-resolution face images; 3) Facial action unit intensity manipulation with limited subjects' pool.

1.2.1 Gaze Estimation

Conventional appearance-based 3D gaze estimation methods generally use the roll of the head pose to represent the eyeball's roll status by default. To reduce degrees of freedom of head poses, a normalization step was proposed to apply global transformations to images to make heads upright and eyelids horizontal. However, due to the ocular countering-rolling (OCR) response, the eyeball will rotate in the opposite direction when the head tilts to the side. After normalization, the eyeball will have an extra roll compared to the roll status of the eyeball when the head is not tilted. This roll from the OCR response causes a changed orientation of the eyeball in normalized eye images, which represents the roll status of the anatomical structure inside the eyeball and consequently affects gaze directions. Thus in this work, we propose a pipeline to regress the person-dependent anatomical variation as a calibration process with considering the OCR response, which can work with our proposed eye-image-based person-independent gaze estimator trained with real and synthetic eye images. The proposed method firstly brings the OCR response into the gaze estimation task, achieving better performances on the two benchmark datasets with fewer parameters under the real-time scenarios. With a replacement of a deeper network, compared to state-of-the-art methods, the proposed method is more efficient, achieving a). better average estimate (3.9% and 2.5% improvement), b). much better standard deviation (lower by 59.0% and 44.2%) and c). a much lower number of parameters (reduced by 88.0%).

1.2.2 Gaze Redirection

Learning-based gaze estimation methods require large amounts of training data with accurate gaze annotations. Facing such demanding requirements of gaze data collection and annotation, several image synthesis methods were proposed, which successfully redirected gaze directions precisely given the assigned conditions. However, these methods focused on changing gaze directions of the images that only include eyes or restricted ranges of faces with low resolution (less than 128 × 128) to largely reduce interference from other attributes such as hairs, which limits application scenarios. To cope with this limitation, we proposed a portable network, called ReDirTrans, achieving latent-to-latent translation for redirecting gaze directions and head orientations in an interpretable manner. ReDirTrans projects input latent vectors into aimed-attribute embeddings only and redirects these embeddings with assigned pitch and yaw values. Then both the initial and edited embeddings are projected back (deprojected) to the initial latent space as residuals to modify the input latent vectors by subtraction and addition, representing old status removal and new status addition. The projection of aimed attributes only and subtraction-addition operations for status replacement essentially mitigate impacts on other attributes and the distribution of latent vectors. Thus, by combining ReDirTrans with a pretrained fixed e4e-StyleGAN pair, we created ReDirTrans-GAN, which enables accurately redirecting gaze in full-face images with 1024 × 1024 resolution while preserving other attributes such as identity, expression, and hairstyle. Furthermore, we presented improvements for the downstream learning-based gaze estimation task, using redirected samples as dataset augmentation.

1.2.3 Action Unit Intensity Manipulation

Facial action unit (AU) intensity plays a pivotal role in quantifying fine-grained expression behaviors, which is an effective condition for facial expression manipulation. However, publicly available datasets containing intensity annotations for multiple AUs remain severely limited, often featuring a restricted number of subjects. This limitation places challenges to the AU intensity manipulation in images due to disentanglement issues, leading researchers to resort to other large datasets with pretrained AU intensity estimators for pseudo labels. In addressing this constraint and fully leveraging manual annotations of AU intensities for precise manipulation, we introduce AUEditNet. Our proposed model achieves impressive intensity manipulation across 12 AUs, trained effectively with only 18 subjects. Utilizing a dual-branch architecture, our approach achieves comprehensive disentanglement of facial attributes and identity without necessitating additional loss functions or implementing with large batch sizes. This approach offers a potential solution to achieve desired facial attribute editing despite the dataset's limited subject count. Our experiments demonstrate AUEditNet's superior accuracy in editing AU intensities, affirming its capability in disentangling facial attributes and identity within a limited subject pool. AUEditNet allows conditioning by either intensity values or target images, eliminating the need for constructing AU combinations for specific facial expression synthesis. Moreover, AU intensity estimation, as a downstream task, validates the consistency between real and edited images, confirming the effectiveness of our proposed AU intensity manipulation method.

1.3 Organization

The rest of the thesis is organized as follows. Chapter 2 discusses the eye-image-based gaze estimation. We proposed a gaze estimation pipeline with the OCR-aware Kappa Angle regression as a personal calibration process. Chapter 3 discusses the gaze and head redirection task. We presented an interpretable redirection network, which can work with trainable or fixed encoder-generator pairs to achieve gaze and head redirection accordingly given the provided pitch and yaw values of new gaze directions and head orientations. Chapter 4 discusses the action unit intensity manipulation task. We presented a promising solution for editing facial attributes despite the dataset's limited subject count, which successfully achieved action unit intensity manipulation based on intensity values or target images without retraining the network or requiring extra estimators. Chapter 5 presents conclusion and discussion on future work.

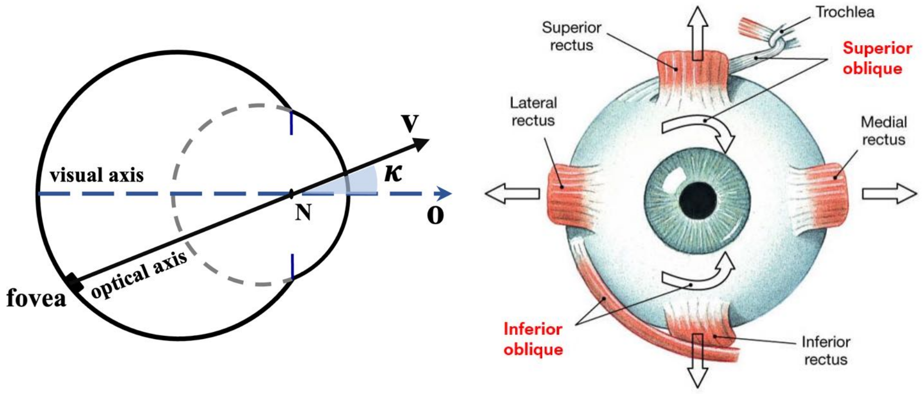
Chapter 2

Kappa Angle Regression With Ocular Counter-Rolling Awareness for Gaze Estimation

2.1 Introduction

Human gaze is an essential indicator for many applications such as human-computer interaction [42, 43], health assessment [44], automotive assistance [45, 46] and virtual reality [47, 48]. Non-invasive appearance-based gaze estimation methods enjoyed significant improvements [49, 16, 14] for in-the-wild settings due to the development of the Convolutional Neural Network (CNN). However, they still struggle with achieving high accuracy due to the challenges caused by variations of head poses [12, 50], noisy and limited annotations [16], eye shapes and anatomical variations of different subjects [2, 51], etc.

Several techniques, ranging from normalization for data pre-processing [11, 50] to individual-specific calibration after training [49], were proposed to reduce the variations stated above. Image normalization's fundamental idea is reducing the degrees of freedom of the object



(a) Eyeball Structure (b) Eyeball Muscles and Motions

Figure 2.1: Eyeball structure and muscles. (a) Kappa Angle $\boldsymbol{\kappa}$ is defined as the angle between visual axis $\boldsymbol{V}$ (the line connecting the fovea and nodal point N, which defines gaze and is unobserved) and optic axis $\boldsymbol{O}$ (the line connecting the eyeball center and pupil center, which is related to the observed iris [2]). (b) The arrows show the eyeball motions controlled by the corresponding muscles. The oblique muscles are used for the eyeball roll motion [52].

pose from six (head poses: pitch, yaw, roll and position: x, y, z) to two (head poses: pitch, yaw) by perspective image warping. This normalization step facilitates mapping from images to gaze directions across different samples or even datasets [53]. Another source of variation causing limited accuracy with a person-independent gaze estimator emerges from the anatomical structures of the eyes. As shown in Fig. 2.1 (a), the visual axis is not aligned with the optic axis (related to the observed iris) [2], and such alignment differences, called 'Kappa Angle', are subject-specific. Given this unobserved anatomical variation across different subjects, person-dependent calibration methods such as gaze differences estimation [51], models calibration with meta-learning [16], and personalized parameters regression [49] were proposed, which further improved gaze estimation performance with a few calibration samples.

However, normalization focuses more on global transformations of images according to head poses and ignores independent eyeball motions. It obeys the human eyeball movement response called ocular counter-rolling (OCR). When the head tilts to the side, the OCR response consists of a torsional conjugate eye movement opposite the static head roll direction around the

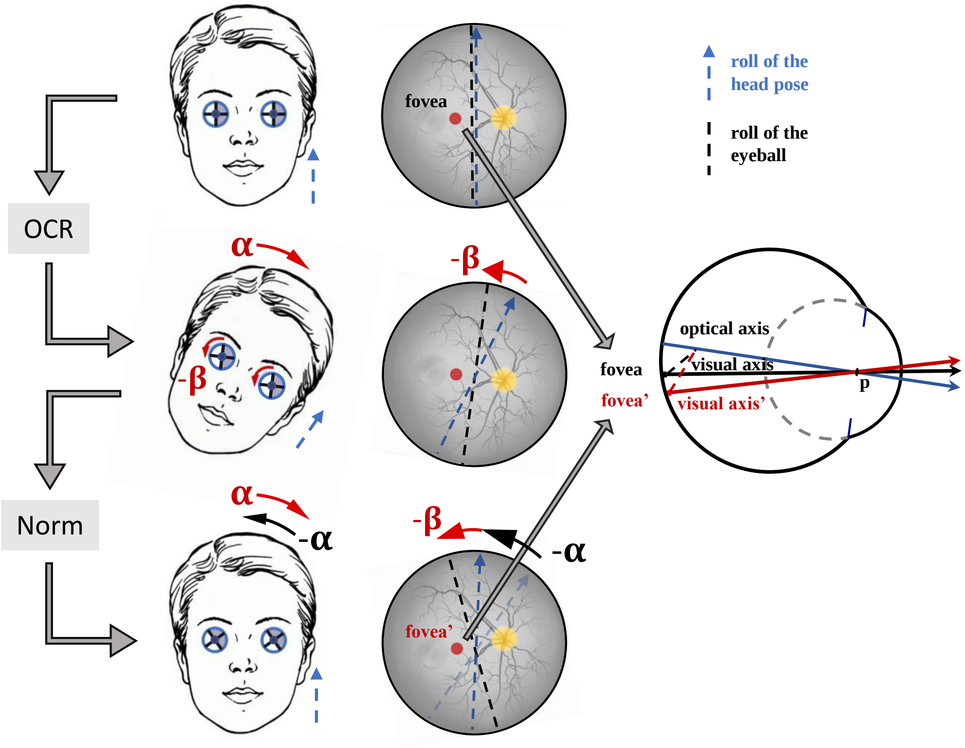


Figure 2.2: Changed gaze directions caused by the processes of OCR and normalization. When a subject rotates his head with a roll of α, the eyeball will rotate in the opposite direction with a roll of −β. After normalization, the roll of the head pose is normalized to zero by an rotation of −α. But the current eyeball still has a roll with −β, which causes different fovea locations and consequently changes visual axis directions.

optic axis [54]. As presented in Fig. 2.2, when the head has a roll motion, the eyeball will have an opposite roll motion to maintain the initial horizontal status instead of rotating together with the head given the indications from iris patterns. After normalization is applied to images, the head and eyelids are transformed to the upright status, but the eyeball's orientation in normalized images still has an extra roll caused by OCR. This extra roll is difficult to acquire from low-resolution eye images and its highly-related variable, the roll of the head pose, is abandoned after normalization. Failing to account for the eyeball's counter-rolling movement is undesired because this movement causes different roll status of the eyeball, which implies different fovea locations and consequently changes gaze directions, shown in Fig. 2.2. Inspired by this observation, we propose a new framework for gaze estimation, which considers the OCR response during the regression of the person-dependent variable: the Kappa Angle.

Our contributions are:

1) Propose a new pipeline to regress the Kappa Angle with considering the factor of OCR.

2) Integrate the Kappa Angle regression part with a unified eye-image-based gaze estimator to achieve person-dependent calibration during both training and evaluation.

3) Present a comparable estimation accuracy and much lower standard deviation with fewer network parameters on benchmark datasets, which indicates the effectiveness of our proposed gaze estimation pipeline with OCR-aware Kappa Angle compensation.

2.2 Related Work

2.2.1 Appearance-Based Gaze Estimation

Appearance-based gaze estimation methods aim at mapping eye-containing images the gaze directions (2D screen locations or 3D gaze direction vectors), which achieved significant improvements [55, 12] compared with geometric approaches [56, 57, 58] given supports from several large-scale datasets [59, 60, 53, 61] and constantly evolving deep learning techniques. GazeNet [55] was the first learning-based 3D gaze estimation method that took one eye image as the input. Except for the single input, multi-branch's inputs included both eyes inputs [62, 51, 63]; full-face inputs [64, 65]; and multi-model inputs [60, 66, 62] were proposed and achieved improvements given extra informative data, which were at the cost of calculation complexity and memory requirements. More recently, person-dependent calibration [51, 49, 16, 15, 67] (or domain adaptation [68, 69, 70]) approaches were proposed, which attempted to remove personal variations (or domain gaps) with a few annotated (or unannotated) samples. Strobl *et al.* [71] utilized the features from a person-independent model over the test subject's data to further train a person-specific Support Vector Regression for personalized gaze estimation. Liu *et al.* [51] proposed learning the gaze difference between two images of the same eye to remove the unobserved person-dependent variables. Chen *et al.* [67] decomposed gaze into a person-independent component estimated from images and a person-dependent bias regressed as network parameters. Liu *et al.* [68] used an ensemble of networks for collaborative learning, guided by outliers. Bao *et al.* [69] introduced the constraint of rotation consistency for unsupervised domain adaptation. Our method follows this gaze decomposition idea. A unified gaze estimator was utilized for estimating the person-independent component of gaze and the person-dependent part was regressed by including OCR.

2.2.2 Gaze Redirection

Given the need for large amounts of labeled data for training a robust gaze estimator, several conditional image synthesis methods were proposed to generate images with desired gaze directions. Ganin *et al.* [13] proposed learning warping fields to rearrange the pixels' locations for gaze redirection given the input images. Yu *et al.* [14] introduced a cycle pipeline with semantic segmentation consistency to supervise warping-field-based gaze redirection and they [15] further extended the warping-field-based methods with an unsupervised learning strategy by representation learning. Besides warping-field-based methods, He *et al.* [18] first applied the Generative Adversarial Network to the gaze redirection task, generating photo-realistic eye images with desired gaze directions. Park *et al.* [16] proposed FAZE: an encoder-decoder structure to change gaze directions and head orientations in feature space with desired labels. Zheng *et al.* [17] presented ST-ED, which first advanced the encoder-decoder method from eye images to full-face images. However, these gaze redirection methods did not consider modeling person-independent components of gaze. Much more accurate gaze estimation results trained and tested only with synthetic data also proved it. Thus in our work, we utilized ST-ED to generate synthetic face images with desired gaze directions for learning the person-independent component of the gaze.

2.2.3 Ocular Counter-Rolling

Ocular Counter-Rolling (OCR) is a partially compensatory torsional eye movement only when the head is tilted toward the shoulder [54]. In particular, the OCR response of human eyeballs is controlled separately by the surrounding muscles called superior and inferior obliques [52], shown in Fig. 2.1 (b). When the head is tilted with α toward the shoulder in a natural pose, these muscles make the eyeball rotate in the opposite direction with β, as shown in Fig. 2.2. After normalization, eye images look horizontally orientated, but the eyeball and the fovea location are tilted with an extra roll (β) owing to the OCR response. This extra roll (β) caused by OCR ***doesn't change*** the absolute value of the Kappa Angle, which is always invariant for the same subject. It only ***redistributes*** the pitch and yaw components of the Kappa Angle. Thus we can compensate this redistribution (counteract OCR) on pitch and yaw components of the Kappa Angle by applying a rotation matrix built by β. Given this, we proposed a Kappa Angle compensation method with OCR awareness, elaborated in Section 2.3.

2.3 Method

In this section, we will firstly discuss the cases without or with considering ocular counter-rolling (OCR). Secondly, we will show the difference between real and synthetic data based on some simulation results. Thirdly, we will introduce the training and evaluation pipeline with considering OCR and the person-dependent part of gaze. Lastly, we will introduce loss functions for supervising the whole process.

2.3.1 Processes W/O or W/ OCR

Fig. 2.1 (a) illustrates that the Kappa Angle ($\boldsymbol{\kappa}$) represents the angle between the optic axis ($\boldsymbol{O}$) and the visual axis ($\boldsymbol{V}$), and is dependent on the individual.

$$\boldsymbol{O} + \boldsymbol{\kappa} = \boldsymbol{V}, \tag{2.1}$$

where $\boldsymbol{O}$, $\boldsymbol{V}$ and $\boldsymbol{\kappa} \in \mathbb{R}^{2\times 1}$ (2D vectors representing **pitch and yaw**), and hence we can use the addition to depict the 3D relationship of these variables. According to Atchison's study [72], the absolute angle value (norm of pitch and yaw) of the Kappa Angle remains constant for the same subject. However, if the eyeball's roll status changes with respect to the head coordinate system, the pitch and yaw of the Kappa Angle will adjust accordingly, as depicted in Fig. 2.2.

W/O OCR: Diff-NN [51] is a typical method without considering OCR in the gaze estimation task. While the optical axis $\boldsymbol{O}$ can be estimated from images using a unified model, there is no ground truth available. On the other hand, the gaze direction $\boldsymbol{V}$ does have ground truth. However, because the Kappa Angle is person-specific and not directly observable from images, it is not possible to estimate $\boldsymbol{V}$ using images alone. To address this, Diff-NN estimates the difference in gaze by subtracting the unobservable Kappa Angle and leveraging the available ground truth. Given two images $(\boldsymbol{I}_1, \boldsymbol{I}_2)$ from the same eye, the gaze difference is

$$\boldsymbol{V}_1 - \boldsymbol{V}_2 = (\boldsymbol{O}_1 + \boldsymbol{\kappa}_1) - (\boldsymbol{O}_2 + \boldsymbol{\kappa}_2), \tag{2.2}$$

where the subscripts denote variables related to the respective images. If we don't consider OCR during gaze estimation, the pitch and yaw of the Kappa Angle maintain constant regardless of different head poses between images. In this case, Eq. 2.2 simplifies to

$$\boldsymbol{V}_1 - \boldsymbol{V}_2 = \boldsymbol{O}_1 - \boldsymbol{O}_2 \text{ if } \boldsymbol{\kappa}_1 = \boldsymbol{\kappa}_2, \tag{2.3}$$

indicating the scenario without considering OCR.

W/ OCR: Due to the presence of OCR response, the eyeball undergoes an additional roll ($-\beta$) that counteracts the roll motion of the head (α), as illustrated in Fig. 2.2. Even after normalization where the head roll is removed, the extra roll ($-\beta$) of the eyeball persists in the normalized eye images. As a result, the pitch and yaw of the Kappa Angle vary for the same subject's data and consequently Eq. 2.3 is no longer valid. We can update Eq. 2.1 to

$$\boldsymbol{O} + \mathcal{T}^{-1}\left[\boldsymbol{R}_{OCR} \cdot \mathcal{T}(\underline{\kappa})\right] = \boldsymbol{V}, \tag{2.4}$$

where $\boldsymbol{R}_{OCR} \in \mathbb{R}^{3\times 3}$ is a roll rotation matrix built given the OCR response; $\underline{\kappa}$ represents the Kappa Angle with invariant pitch and yaw components; $\mathcal{T}$ is a function to transform pitch and yaw to a 3D directional unit vector and $\mathcal{T}^{-1}$ represents the inverse process. As reported in the statistics [73, 74], the roll (β) from OCR is around $1/7.5$ of the roll motion (α) of the head.

2.3.2 Real and Synthetic Data

The optic axis is the line connecting the nodal point with the pupil center, which is related to the observed iris [2] and can be estimated from images [67]. However, the optic axis is not provided in gaze datasets. The visual axis defines gaze, which is what we want to estimate. However, due to the subject-dependent unobservable deviations between visual and optic axes [72, 2], a unified gaze estimator does not work well on new subject data. Considering this, we proposed to utilize synthetic redirected eye images with manually set gaze directions to learn the approximation of the optic axis given the simulation results. We quantitatively evaluated the subject-dependent variations across different subjects' eye images between real and synthetic data. Synthetic data was generated based on real data with assigned conditions by using ST-ED [17]. We trained a three-layer CNN with either real or synthetic eye images and tested its performance with the 'leave-one-subject-out' protocol. The mean angular errors are 6.89° and

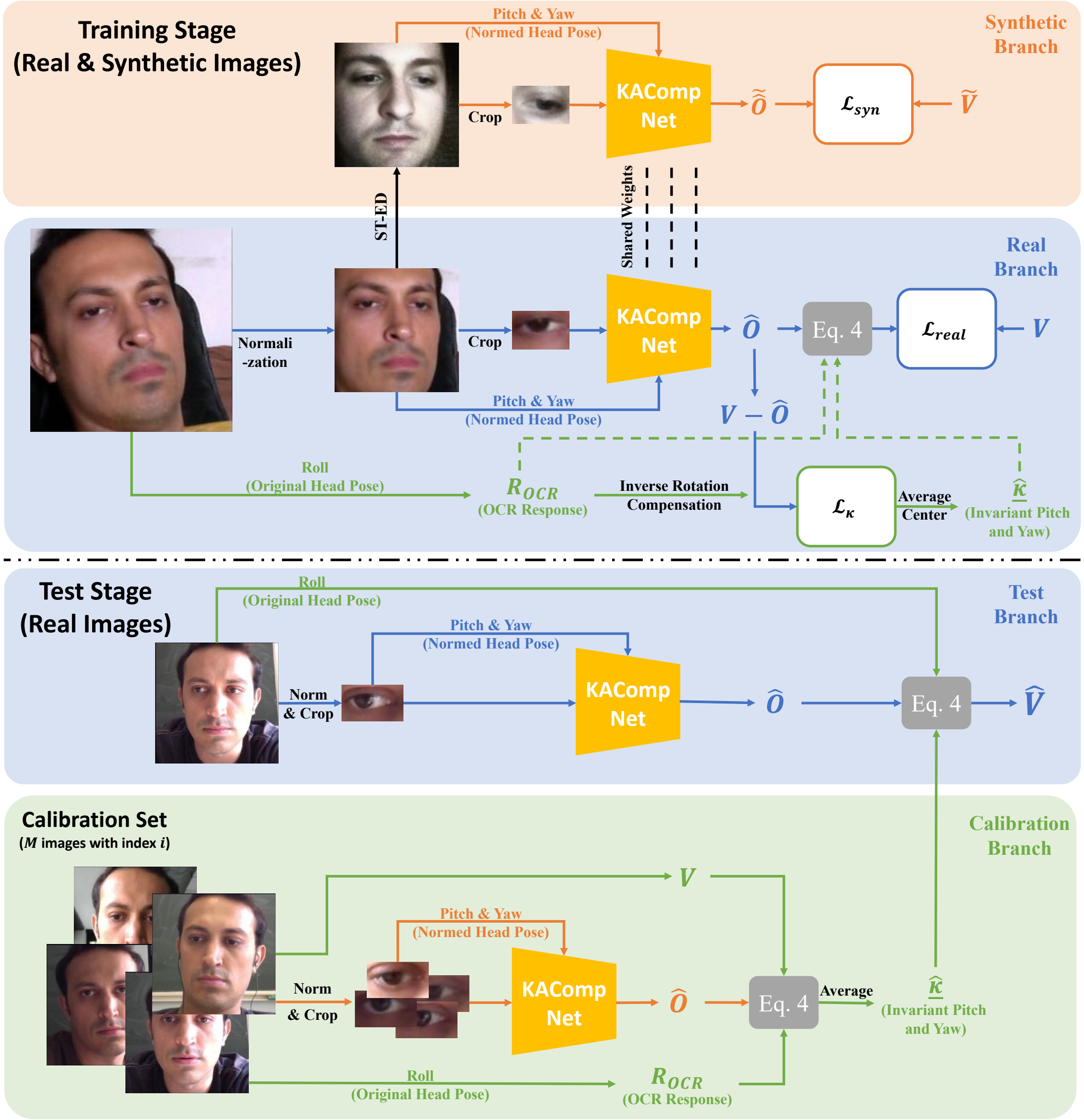


Figure 2.3: Training (top) and test (bottom) pipelines of KAComp-Net. Training stage has two branches: synthetic branch aids the CNN in learning the optic axis directions by generated data from ST-ED with manually assigned labels, while real branch focuses on estimating the OCR-compensated Kappa Angle $\underline{\widehat{\kappa}}$ with invariant pitch and yaw components. Test stage consists of calibration and test branches. Calibration branch estimates the Kappa Angle of the test subject using M labeled data. Then, test branch estimates the final gaze directions using the output from CNN and the estimated Kappa Angle. $\boldsymbol{V}$ ($\boldsymbol{O}$) denotes the visual (optic) axis directions. $\widehat{(\cdot)}$ denotes the estimated variables and $\widetilde{(\cdot)}$ denotes the synthetic variables used by ST-ED. $\mathcal{L}$ denotes the loss functions which are elaborated in Section 2.3.4.

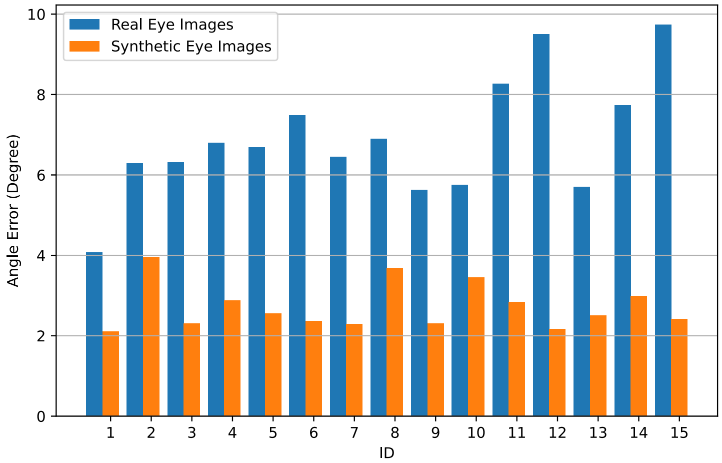


Figure 2.4: Comparison of angular errors given real and synthetic data from MPIIFaceGaze with the leave-one-subject-out protocol. The network has a single branch with three convolutional layers.

2.72° on the real and synthetic data from MPIIFaceGaze [64], shown in Fig. 2.4. The mean angular errors of EYEDIAP [59] are 7.31° (real data) and 2.30° (synthetic data). We noted that there existed a **large gap** in angle errors between real and synthetic data. In other words, the subject-dependent unobservable deviations from the Kappa Angle across different subjects were largely removed after gaze redirection. Based on this, we utilized synthetic eye images to learn the person-independent part of gaze, viewed as the approximation of the optic axis.

2.3.3 Pipeline

Training Stage. To begin with, given a real face image, we apply the preprocessing step utilized in ST-ED to obtain a normalized face image with the corresponding normalized ground truth gaze $\boldsymbol{V}$, the normalized head pose $\boldsymbol{H}$ and the roll motion $\boldsymbol{H}^{roll}$ of the head before normalization. Next, we input the normalized face image into ST-ED to generate redirected face images using the provided condition as pseudo labels: gaze $\widetilde{\boldsymbol{V}}$ and head pose $\widetilde{\boldsymbol{H}}$. We then crop the same-side eye images from the normalized real and synthetic face images and feed them into a single-stream convolutional neural network (CNN) that takes one eye image at a time as input. The normalized head pose is attached to the intermediate features of the eye images, similar to GazeNet [53]. The output of the CNN is the estimated direction (pitch and yaw) of the subject-independent component of the gaze, which approximates the optic axis.

We use the roll motion $\boldsymbol{H}^{roll}$ to calculate the roll status of the eyeball and build the rotation matrix $\boldsymbol{R}_{OCR}$, as described in Section 2.3.1. We can then estimate the Kappa Angle for each instance using Eq. 2.4, based on the estimated optic axis, the ground-truth gaze and the roll status of the eyeball. To ensure that the pitch and yaw of the estimated Kappa Angle are identical for the same subject, we employ the Kappa Angle loss, which is built on the Center Loss [75]. This loss aims at reducing standard deviations of the estimated instance-wise Kappa Angle within each subject's data and iteratively updates the average center as the subject-wise Kappa Angle. Finally, the estimated subject-wise Kappa Angle is used to update the unobservable subject-dependent part and combined with the estimated optic axis for final gaze estimation.

Evaluation Stage. During evaluation, we randomly pick a certain number (M) of calibrated samples with ground-truth labels from the same test subject. We input these samples into the trained CNN to estimate their optic axis directions. Using the provided gazes and the calculated OCR response, we derive several instance-wise Kappa Angles $\widehat{\underline{\kappa}}_i$ from calibrated samples based on Eq. 2.4. We then calculate their average as the estimated subject-wise Kappa Angle $\widehat{\underline{\kappa}}$. Finally, we use the estimated optic axis directions of the target samples and $\widehat{\underline{\kappa}}$ to determine the visual axis direction as the final gaze. Since we apply the Kappa Angle to the compensation of the subjects' variation, we call it '*Kappa Angle Compensation Neural Network*' and *KAComp-Net* in short.

2.3.4 Loss Functions

We trained our proposed KAComp-Net using multi-objective loss functions defined as

$$\mathcal{L} = \lambda_{syn} \cdot \mathcal{L}_{syn} + \lambda_{real} \mathcal{L}_{real} + \lambda_{\kappa} \cdot \mathcal{L}_{\kappa}, \tag{2.5}$$

where we empirically set $\lambda_{syn}, \lambda_{real} = 1.0$ and $\lambda_{\kappa} = 0.5$. In order to balance real and synthetic eye image groups, we use the same number of real and synthetic eye images during the training. The details of each loss component are elaborated on in the following paragraphs.

Kappa Angle loss. There is no ground truth to supervise the learning of the Kappa Angle and the only clue to restrict it is that the pitch and yaw of $\underline{\kappa}$ keep identical across samples within the same subject's data. Thus we propose the Kappa Angle loss, which aims at making the standard deviation (SD) of the calculated Kappa Angle with considering OCR within every subject's data as small as possible. Inspired by the Center Loss [75] designed for classification, which narrowed the intra-class distances from data points to the class center, we applied it to the Kappa Angle loss.

There are K subjects' data included in the training set. Each subject has N_k real eye images $\boldsymbol{I}^e$ and N_k synthetic eye images $\widetilde{\boldsymbol{I}}^e$. The Kappa Angle loss is defined as

$$\mathcal{L}_{\kappa} = \frac{1}{2N_k} \sum_{i=1}^{N_k} \left(||\widehat{\underline{\kappa}}_i - \boldsymbol{c}_k||_2^2 + ||\widetilde{\widehat{\underline{\kappa}}}_i - \boldsymbol{c}_{K+1}||_2^2 \right), \tag{2.6}$$

where $k = 1, \cdots, K$. The former part, $||\widehat{\underline{\kappa}}_i - \boldsymbol{c}_k||_2^2$, in Eq. (2.6) is designed for **real** eye images, where $\boldsymbol{c}_k$ represents the center point (mean values) of the calculated $\widehat{\underline{\kappa}}_i$ over all samples from the subject with identity number k and $\widehat{\underline{\kappa}}_i$ is calculated given Eq. 2.4 as

$$\widehat{\underline{\kappa}}_i = \mathcal{T}^{-1} \left\{ \boldsymbol{R}_{OCR,i}^{-1} \cdot \mathcal{T} \left[\boldsymbol{g}^{gt}(\boldsymbol{I}_i^e) - \psi(\boldsymbol{I}_i^e) \right] \right\}, \tag{2.7}$$

where $\boldsymbol{R}_{OCR,i}^{-1}$ is the inverse rotation matrix given the OCR response with regard to the i-th real eye image; $\psi(\cdot)$ denotes the output from *KAComp-Net*, which is the estimated direction of the optic axis; and $\boldsymbol{g}^{gt}(\cdot)$ denotes the ground truth gaze direction given the image. The latter part, $||\widetilde{\widehat{\underline{\kappa}}}_i - \boldsymbol{c}_{K+1}||_2^2$, in Eq. (2.6) is designed for **synthetic** eye images. Since the Kappa Angles are no longer varied across different subjects' synthetic data, we assign only one center point $\boldsymbol{c}_{K+1}$ to all synthetic data. The subscript $K+1$ means a new center point different from the previous K

center points of real data. $\widetilde{\widehat{\underline{\kappa}}}_i$ is defined as

$$\widetilde{\widehat{\underline{\kappa}}}_i = \boldsymbol{g}^{gt}(\widetilde{\boldsymbol{I}}_i^e) - \psi(\widetilde{\boldsymbol{I}}_i^e), \tag{2.8}$$

where we don't consider OCR in synthetic data.

Gaze loss for synthetic images. This loss aims at supervising the network learning the manually designed gaze from synthetic eye images, which have smaller and less varied Kappa Angles across different subjects' data. In other words, this loss guides the network to learn synthetic cases with nearly overlapped optic axis and visual axis.

$$\mathcal{L}_{syn} = \frac{1}{N_k}\sum_{i=1}^{N_k}\left|\left|\boldsymbol{g}^{gt}(\widetilde{\boldsymbol{I}}_i^e) - \psi(\widetilde{\boldsymbol{I}}_i^e)\right|\right|_1. \tag{2.9}$$

Gaze loss for real images. The aim of importing this gaze loss is to balance real and synthetic data influences. Since we have center points for every subject, which represent the estimated Kappa Angle, we can remove this unobservable subject-dependent part from ground truth gaze to acquire the optic axis directions for real eye images as the ground truth. To be specific,

$$\begin{aligned} \mathcal{L}_{real} &= \frac{1}{N_k}\sum_{i=1}^{N_k}\left|\left|\psi(\boldsymbol{I}_i^e) - \hat{\boldsymbol{O}}(\boldsymbol{I}_i^e)\right|\right|_1, \\ \hat{\boldsymbol{O}}(\boldsymbol{I}_i^e) &= \boldsymbol{g}^{gt}(\boldsymbol{I}_i^e) - \mathcal{T}^{-1}\left[\boldsymbol{R}_{OCR,i}\cdot\mathcal{T}(\boldsymbol{c}_k)\right]. \end{aligned} \tag{2.10}$$

2.4 Experiments

In this section, we thoroughly evaluated the performance of the proposed algorithm with other state-of-the-art methods on published datasets. We also elaborated several impacts on the proposed algorithm, such as the numbers of references in calibration, the proportion of synthetic images and the estimated Kappa Angles distribution.

Table 2.1: Quantitative comparison (MPIIFaceGaze and EYEDIAP) with state-of-the-art eye-image-based gaze estimation methods, which are classified given the need for the calibration (**CAL**) step. The network size can be described by the backbone category of a single channel and the number of channels. The types of input image(s) are: one single (**Single-**) eye, both-side (**Both-**) eyes from one face image and same-side (**Same-**) eyes from different face images of one subject. The estimated gaze type '**Eye**' represents that the origin of gaze directions is the center of the eye. The estimated gaze type '**Face**' represents that the origin of gaze directions is the center between two eyes' centers.

CAL	Methods	Backbone	Num of Channels	Input Image(s)	Estimated Gaze Type	Mean Angle Error (Degree)	
						MPIIFaceGaze	EYEDIAP
No	GazeNet [53]	VGG-16	1	Single-Eye	Eye	5.83	6.83
	ARE-Net [62]	6 CL	4	Both-Eyes	Face	5.02	6.08
	CA(Eye)-Net [63]	10 CL	2	Both-Eyes	Face	5.01	5.30
	AGE-Net[76]	9 CL + 4 Dilated CL	2	Both-Eyes	Face	4.64	-
Yes	Diff-NN [51]	3 CL	2	Same-Eyes	Eye	4.72±0.40	4.51±0.52
	KAComp-Net	3 CL	1	Single-Eye	Eye	**4.21±0.28**	**3.89±0.25**
	RedFTAdap [15]	VGG-16	1	Single-Eye	Eye	4.01	-
	Faze [16]	DenseNet	1	Both-Eyes	Face	3.90	-
	DAGEN[1] [77]	ResNet-18	1	Both-Eyes	Face	3.74	4.30
	Diff-NN-VGG [51]	VGG-16	2	Same-Eyes	Eye	3.8±0.61	3.53±0.52
	KAComp-Net-VGG	VGG-16	1	Single-Eye	Eye	**3.65±0.25**	**3.44±0.29**

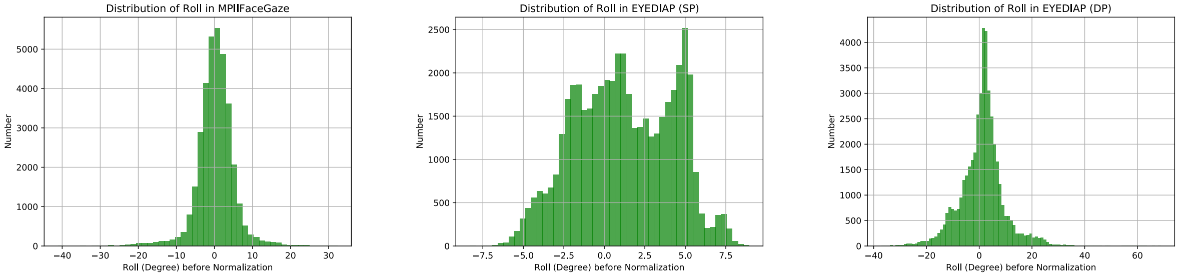


Figure 2.5: Distribution maps of the roll of the head pose before normalization in two benchmark datasets. These distribution maps revealed the presence of the OCR response when the roll of the head is not in a zero position. Since we utilized 'leave-one-subject-out' protocol, each sample in MPIIFaceGaze and EYEDIAP was included in the test subset.

2.4.1 Datasets

Real Eye Image Datasets. MPIIGaze [53] is a widely used benchmark dataset for the appearance-based in-the-wild gaze estimation task. In our experiments, due to the need to generate synthetic face images, we utilized its subset MPIIFaceGaze [64], which contains 37667 full-face images captured from 15 participants' images (nine males and six females). EYEDIAP [59] contains 94 full-face videos from 16 subjects with labeled outliers (blinking or distraction) of each frame. We utilized the data from discrete and continuous screen targets with both static (SP) and dynamic (DP) head poses, covering 14 participants (11 males and 3 females).

Since raw images in both datasets contain the upper torso and the provided data collection information indicates a horizontal camera position, we estimate the roll of the head pose in raw images as the actual roll of the head to eliminate any ambiguity arising from the camera pose. Fig. 2.5 presents the distribution given MPIIFaceGaze [64] and EYEDIAP [59] (SP and DP, respectively). We can notice that in EYEDIAP (DP), the distribution of the roll of the head is wider compared to MPIIFaceGaze. On the other hand, EYEDIAP (SP) exhibits the smallest range of roll of the head.

Generated Synthetic Eye Images. Synthetic eye images are normalized from the synthetic face images generated by ST-ED [17]. ST-ED utilizes the 'face' gaze instead of the 'eye'

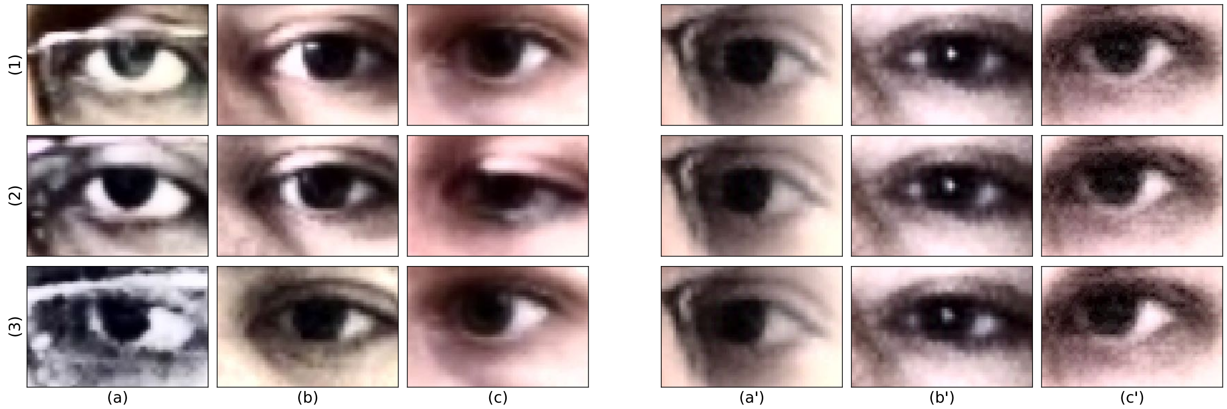


Figure 2.6: Comparison between real (left three columns) and synthetic (right three columns) eye images. Columns (a) to (c) are eye images from different persons in MPIIFaceGaze [64]. Columns (a') to (c') are synthetic images generated from the three previous columns, respectively. Rows (1) to (3) represent the gaze direction with the same pitch (-5 degrees) and the changed yaw of 5, 10, 15 degrees, respectively.

gaze during gaze redirection. The gaze directions are defined by the gazing target point and source point. The main difference between the 'face' and 'eye' gaze comes from the source point's 3D locations. The source point of the 'face' gaze is the midpoint between two eye centers. Thus we only have one 'face' gaze direction for each face image. As for the 'eye' gaze, the corresponding source point is the center of the eye, which means we have two 'eye' gaze directions for each face image.

To make the redirection of the 'face' gaze consistent with our 'eye' gaze estimation task, we calculate the gazing target location instead of the gaze direction during the preprocessing and redirection processes. To be specific, we first normalize face images given the preprocessing requirements of ST-ED. After normalization, in addition to saving the normalized (rotated) gaze directions, we also keep the normalized gazing target location. During the process of redirecting the gaze directions, we assume that the gazing target maintains the same distance to the source point. Then we normalize [50] the redirected face images given the 'repositioned' gazing target to acquire normalized eye images with the 'eye' gaze directions. Fig. 2.6 shows several normalized real and synthetic eye images with the dataset provided (left three columns) or assigned 'eye' gaze directions (right three columns).

Gaps between Real and Synthetic Images. The gap between real and synthetic data shown in Figure 2.4 does not provide conclusive evidence that the unobserved Kappa Angle is the cause. To investigate further, we mixed the real and synthetic data in training Diff-NN to determine if the unobserved person-dependent component was eliminated. However, as illustrated in Section 2.4.4, the mixture of real and synthetic data performs even worse than real data alone, providing further evidence of the absence of the Kappa Angle of the synthetic data.

2.4.2 Implementation Details

Network Architecture. The network only contains one single branch built with three convolutional layers and three fully connected layers. The convolutional part's structure inherits from the Diff-NN [51]. Each time, we feed only one eye image $\boldsymbol{I}^e \in \mathbb{R}^{H \times W \times C}$ into the network where $(H, W, C) = (48, 72, 3)$. The extracted features from the convolutional part are fed into the fully connected part. The corresponding head pose is concatenated with the output from the first fully connected layer. Then the last two fully connected layers are applied to the concatenated output to estimate the optical axis direction with yaw and pitch components.

The proposed network architecture has several advantages. The first point is the simple network structure. Instead of accompanying several inputs and branches for one output, our proposed network only takes one input but still has comparable prediction performance. The second point is overfitting avoidance. The differential method needs a dropout layer to avoid overfitting, which can cause bad performance on the new participant data. However, the proposed KAComp-Net could proactively prevent this obstacle by eliminating dropout layers.

Training Parameters. We train KAComp-Net with 12 epochs and a batch size of 128. The initial learning rate is set as 0.01. After each epoch, the learning rate is divided by 2. The optimizer is Adam [78], with a weight decay coefficient of 0.09. We use the default momentum value of $\beta_1 = 0.9, \beta_2 = 0.999$.

Evaluation Protocol. We cross-validated the methods' performance within the published datasets. In detail, we utilized the ‘leave-one-subject-out’ protocol when we evaluated the models within MPIIFaceGaze or EYEDIAP. Each time we select one subject’s data as the test set, and the rest was viewed as the training set. Note that only real data was utilized as the test set, and the synthetic data generated from the test subject’s data was not included in the training set in case of data leakage. At test time, we needed to choose several eye images for calibration. In order to alleviate the bias from some calibrated samples, we repeated testing the same trained model 200 times with random combinations of samples and calculated the mean angular errors of the predicted gazes and the standard deviations as the corresponding trained models’ performance on the test subject’s data. We looped all subjects’ data as the test set one by one and reported the average of mean angular errors and the standard deviations. Since the proposed KAComp-Net aims at predicting the single-eye gaze, we trained and evaluated the models on left and right eye images separately.

Gaze Inference. In the testing phase, we randomly pick a certain number (M) of calibrated samples F^e with the gaze directions. We first feed these samples into the KAComp-Net to estimate their optical axis directions. Then given the provided gazes and the OCR response, we can derive several $\{\widehat{\underline{\kappa}}_i, i \in [1, M]\}$ from M calibrated images. We then utilize their average to represent the estimated Kappa Angle,

$$\widehat{\underline{\kappa}} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{R}_{OCR,i}^{-1} \cdot \left[\boldsymbol{g}^{gt}(\boldsymbol{F}_i^e) - \psi(\boldsymbol{F}_i^e)\right] . \tag{2.11}$$

Given the estimated Kappa Angle from calibrated samples, we can predict the gaze. The predicted gaze error of eye image $\boldsymbol{I}_j^e$ is

$$\boldsymbol{g}^{err}(\boldsymbol{I}_j^e) = \boldsymbol{g}^{gt}(\boldsymbol{I}_j^e) - \left[\psi(\boldsymbol{I}_j^e) + \boldsymbol{R}_{OCR,j} \cdot \widehat{\underline{\kappa}}\right] , \tag{2.12}$$

where the $\widehat{\underline{\kappa}}$ is derived from Eq. (2.11).

2.4.3 Comparison with Eye-image-based Methods

We listed several state-of-the-art eye-image-based gaze estimation methods in Table 2.1, which were categorized into two groups according to the needs for calibration. There were two kinds of outputs: *eye gazes* and *face gazes*. The *eye gaze* represents the direction from the eye center to the gazing target, usually used for single-eye gaze estimation. The *face gaze* represents the direction from the center of two eye centers to the gazing target, which requires more inputs (e.g. left and right eye images).

Effectiveness of Calibration. It was straightforward to notice that with the assistance of a few ($M = 9$) calibration samples from the test set, the methods achieved significant improvements even if the networks were much simpler. However, better performance and lower calculation complexity were at the cost of the need for several calibrated samples and labeled gazes, which required extra efforts (e.g., calibration before use) under practical scenarios.

Methods with Calibration. We first compared two shallow networks' (Diff-Net[51], KAComp-Net) performance, which were designed for real-time purposes. Then we replaced the proposed KAComp-Net backbone from three-layer CNN to the VGG-16 backbone for fair comparisons with other calibration-needed state-of-the-art methods.

Compared with Diff-NN, the proposed KAComp-Net worked better with only around half of the multiply-accumulate (MAC) operations. The KAComp-Net performed a $0.51^\circ(10.81\%)$ boost on the MPIIFaceGaze and a $0.62^\circ(13.75\%)$ boost on the EYEDIAP. Apart from decreasing the mean angular error, KAComp-Net maintained a more stable performance given different calibration sets. The standard deviation of KAComp-Net was reduced by $0.12^\circ(30.00\%)$ and $0.27^\circ(51.92\%)$ compared to those of Diff-NN in MPIIFaceGaze and EYEDIAP, respectively. These results fully demonstrated that the estimated Kappa Angle with considering OCR is a more unified and robust characteristic than the gaze difference from the same eye when the network depth (learning ability) is limited. The differential-based method assumed that the pitch and yaw of the Kappa Angle were invariant concerning different head poses if the input eye images were normalized, which violated OCR and introduced the undesired person-dependent bias when calculating gaze differences. KAComp-Net considered OCR and used it to derive the invariant pitch and yaw of the Kappa Angle for the optic axis direction estimation, which essentially removed the bias caused by OCR and guided the network to learn a more unified feature from the eye images. The lower standard deviation meant a lower dependence on the calibrated samples, whose impact was further discussed in Section 2.4.5. In order to make the proposed KAComp-Net competitive compared to other state-of-the-art methods, we replaced the three-layer CNN with the pre-trained VGG-16 backbone for better feature extraction ability. The training parameters remained identical after we changed the backbone. KAComp-Net-VGG achieved 8.98%, 6.41% and 3.95% improvements on MPIIFaceGaze compared with RedFTAdap [15], FAZE [16] and Diff-NN-VGG [51], respectively.

2.4.4 Impacts of Synthetic Images

We discussed the effects from synthetic images based on experiments of Diff-NN and KAComp-Net in this section. Synthetic data were generated from real training data only.

Evaluation with Diff-NN. We utilized Diff-NN to investigate synthetic data impacts on the differential-based network. Since Diff-NN needed pairs of eye images from the same subject for training, we implemented three experiments according to the source of paired images: 1) real samples only; 2) separated real or synthetic samples within pairs; 3) Mixture of real and synthetic samples within pairs. The inference process was taken only on real data with 200 repeated evaluations and the number of calibration samples $M = 9$. The performance on real data only (RO), real and synthetic data independently (RS_I) and the mixture of real and synthetic data (RS_M) were $4.51 \pm 0.52°$, $4.27 \pm 0.50°$ and $5.99 \pm 0.68°$ on the EYEDIAP, respectively. RO and RS_M performance were similar, which meant that the synthetic data maintained the same gaze difference property as the real data. The mixture of them achieved worse performance than the other two, which further demonstrated that the Kappa Angle variation of the synthetic data

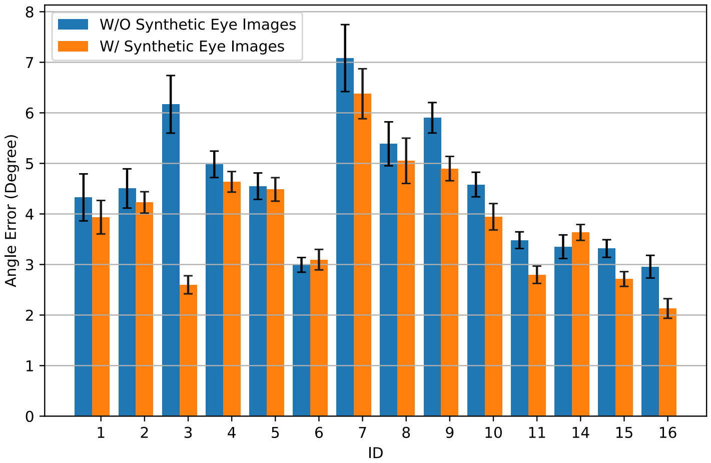


Figure 2.7: Comparison of mean angular errors and standard deviations (200 repeated experiments) of the gaze by KAComp-Net with and without synthetic data according to the leave-one-subject-out protocol in EYEDIAP.

was no longer kept as the real data did.

Evaluation with KAComp-Net. We did the experiments on KAComp-Net with or without synthetic data. Fig. 2.7 elaborates on the impacts of synthetic samples on KAComp-Net. The mean angle error was $4.54 \pm 0.32^\circ$ without synthetic data and $3.89 \pm 0.25^\circ$ with synthetic data in EYEDIAP. Synthetic data played an important role during the training of the KAComp-Net because it helped supervise the network learning a unified characteristic and further improved the accuracy for the Kappa Angle regression.

2.4.5 Impacts of Calibrated Samples

Fig. 2.8 illustrates the impact of the number of calibrated samples in EYEDIAP. The evaluation protocol is illustrated in Section 2.4.2. When the number of calibrated samples was less than three, Diff-NN had similar performance compared with no-calibration-needed methods. Especially when the number of calibrated samples $M = 1$, Diff-NN achieved 1.03° worse than GazeNet [53], mainly due to large gaze differences between limited calibrated samples and target

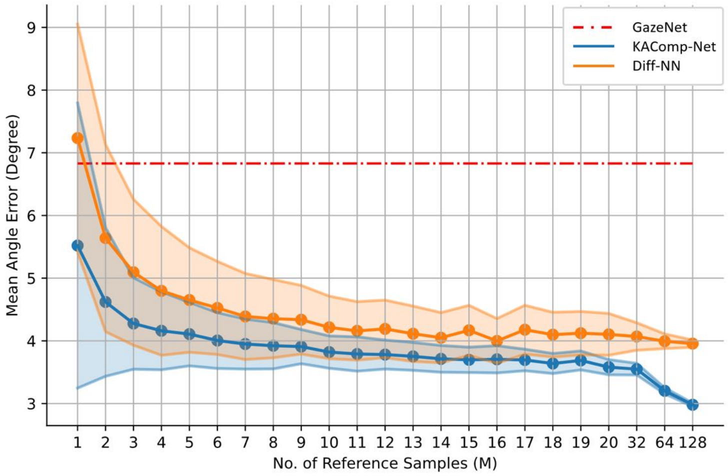
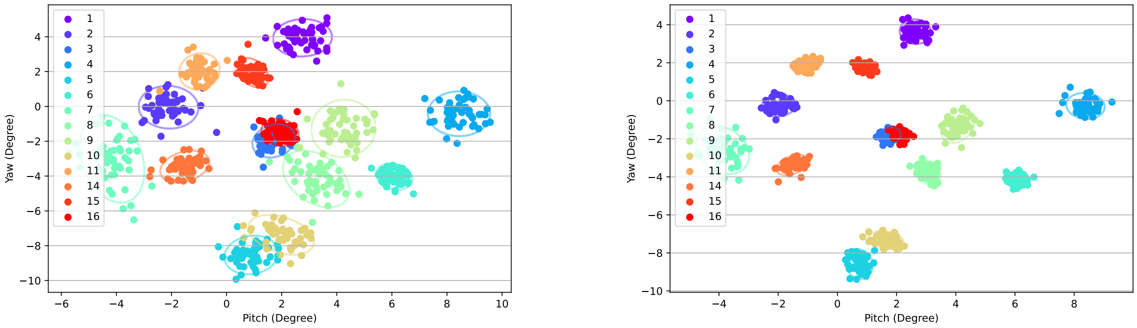


Figure 2.8: Comparison of mean angular errors and standard deviations (200 repeated experiments) among the state-of-the-art methods given different numbers of calibrated samples in EYEDIAP.

ones, and the number of network layers. As the number of calibrated samples increased, the prediction errors and the standard deviations of Diff-NN dropped significantly because more calibrated samples with similar gaze directions to target ones were acquired. Given the same number of calibrated samples, KAComp-Net achieved more accurate and more stable results than Diff-NN, proving the higher tolerance to the calibrated samples. Even with only one calibrated sample, KAComp-Net can achieve $1.32°(19.33\%)$ improvement compared with GazeNet. When the number of calibrated samples was larger than 64, KAComp-Net can further improve the estimation accuracy, unlike the plateauing performance of Diff-NN, shown in Fig. 2.8. The main reason for this phenomenon was given the estimation accuracy of the Kappa Angle from calibrated samples, which was discussed in detail in Section 2.4.6.

2.4.6 Estimated Kappa Angle Distribution

During the inference, we first calculated the Kappa Angles from the calibrated samples of the test subject. The estimated Kappa Angle distribution maps with different numbers (9 and 64) of calibrated samples were shown in Fig. 2.9 based on 50 repeated experiments. Note that with more calibrated samples for calibration, the estimated Kappa Angles had smaller standard



(a) $M = 9$

(b) $M = 64$

Figure 2.9: Distribution maps of the estimated Kappa Angles given different subjects with KAComp-Net given M calibrated samples. The legend number means subject ID in EYEDIAP.

deviations, which yielded smaller angular errors, shown in Fig. 2.7. An obvious comparison was found by the distribution maps between the subjects with ID 7 and 16. The estimated Kappa Angle range of the ID 7 subject was over $6^\circ \times 2^\circ$, and the corresponding predicted angle error was 6.38°, which was 64% higher than the mean angular error over all subjects. However, the distribution map of the ID 16 subject had less than a $2^\circ \times 2^\circ$ area, which achieved 2.13° angular error (45% lower than the mean angular error).

2.4.7 Impacts of Head Pose Variations

KAComp-Net is designed to remove the variance caused by OCR, but it doesn't depend on various head poses (or rolls) to trigger OCR for estimating the Kappa Angle. This is because OCR only affects whether it is needed to compensate for the redistribution of the pitch and yaw of the Kappa Angle before regressing this anatomical variable within each subject's data. Table 2.2 shows consistent improvements compared with Diff-NN under different levels of head pose variations in EYEDIAP, which also proves the importance of considering OCR.

Table 2.2: Estimated mean angle errors given static (SP) and dynamic (DP) head pose data in EYEDIAP.

Methods	Mean Angle Error (Degree)	
	SP	DP
Diff-NN	3.46±0.40	4.76±0.41
KAComp-Net	**3.16±0.26**	**4.37±0.26**

Table 2.3: Complexity Comparison between the Differential Method and Kappa Angle Compensation Method

	Diff-NN	KAComp-Net
Params (M)	42.015	**5.044**
MACs (M)	89.148	**28.581**

2.4.8 Algorithm Complexity

Diff-Net and KAComp-Net share the same three-convolutional-layer backbone, which aims at achieving real-time gaze estimation. Table 2.3 compares the size of the network and the number of multiply–accumulate (MAC) operations. We observe that KAComp-Net reduced 87.99% (67.94%) parameters (MACs) compared with Diff-NN.

2.5 Discussion

2.5.1 Ambiguity from the Camera Pose

Head poses in images can be modified due to different camera poses, even if the subject's head pose remains invariant. When using benchmark datasets, we can assume that the camera was placed horizontally based on data collection settings and clues from upper torsos, as discussed in Section 2.4.1. To ensure accurate estimation of head roll motion in practical scenarios, it is crucial to determine the camera's roll pose with respect to the horizontal level. If the camera can be placed statically, it can be manually calibrated to ensure it is positioned horizontally.

A possible alternative is capturing high-resolution iris images. Then the OCR response can be directly estimated from these images without relying on the derivation from head roll motion, as demonstrated in [79].

2.5.2 Why Rotation

In the normalization step, when the roll of the head is normalized to an upright status, the ground truth gaze is transformed by a rotation matrix instead of an affine transformation matrix, as illustrated in [50]. This process is similar to our OCR compensation process. When OCR occurs, the eyeball has an undesired roll after normalization, which redistributes the pitch and yaw of the Kappa Angle. To counteract this redistribution caused by various roll statuses within the same subject's data, we apply rotation matrices to compensate, as shown in Eq. 2.4.

2.5.3 Listing's Law and OCR

Ocular counter-roll (OCR) is a vestibulo-ocular reflex characterized by torsional rotations of the eye in response to lateral tilt of the head [73]. Listing's law states that when the head is fixed, there is an eye position called primary position, such that the eye assumes only those orientations that can be reached from primary position by a single rotation about an axis in a plane called Listing's plane [80]. Listing's law holds during fixation, saccades, smooth pursuit, and vergence, but fails during sleep and vestibulo-ocular reflex [81], including OCR.

When the head tilts to the side, OCR occurs, causing the eye to rotate around the roll axis that is out of Listing's plane. This means that the orientation of Listing's plane changes when the head is tilted, as shown in [82]. However, if the head maintains a static tilted posture, Listing's Law still applies, and eye rotation vectors are still confined to a plane. This plane is shifted along the torsional axis in relation to the upright position, proportional to the roll-tilt angle [83]. In this case, the eye orientation can still be represented by pitch and yaw components with a constant torsional bias. Our proposed KAComp-Net considers the bias caused by the OCR, and the remaining estimation processes are consistent with the cases where OCR doesn't happen.

2.5.4 Limitations

Our proposed method has several limitations, both from the structural design perspective and the data perspective. These limitations are viewed as research directions for future work.

Synthetic Data. KAComp-Net requires synthetic data to aid in the learning process of the optical axis direction. Compared with the real data, synthetic data has less unobserved person dependent components of gaze directions, as shown in Section 2.3.2 and Section 2.4.4. Although we took advantage of this property regardless of the gap, we still need to get rid of the dependence on synthetic data. In the future, the network can learn unified features directly from real data without Siamese learning between data from different domains. This could potentially improve the estimation accuracy.

Static / Dynamic OCR. KAComp-Net only considers static OCR response, which is related to the roll of the head. However, during head tilt, dynamic OCR occurs with slow phases away from and quick phases toward the head tilt [84]. With a sustained head tilt, the static OCR occurs, resulting in a static change in torsional eye position in the direction away from the head tilt [73]. In future work, if we have access to consecutive frames, we can model the process by considering both static and dynamic OCR.

High-Resolution Iris Images. In the KAComp-Net pipeline, OCR needs to be derived from the roll motion of the head, which is normally abandoned after normalization due to the low resolution of eye images. However, if we have high-resolution eye images, we don't need to derive the OCR response. Instead, we can measure the OCR directly given the high-resolution iris images [79] for a more accurate gaze estimation.

2.6 Conclusion

In this work, we derived and proposed a pipeline to regress the pitch and yaw of the Kappa Angle under the head coordinate system given the ocular counter-rolling response. This person-dependent Kappa Angle regression works with an eye-image-based person-independent gaze estimator trained with real and synthetic eye images for person-dependent calibration with a few samples. Several experiments on the benchmark datasets showed the effectiveness and robustness of the proposed methods with limited calibration samples.

Chapter 2, in full, is a reprint of the material as it appears in the publication of “Kappa Angle Regression with Ocular Counter-Rolling Awareness for Gaze Estimation”, Shiwei Jin, Ji Dai, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 3

ReDirTrans: Latent-to-Latent Translation for Gaze and Head Redirection

3.1 Introduction

Gaze is a crucial non-verbal cue that conveys attention and awareness in interactions. Its potential applications include mental health assessment [85, 86], social attitudes analysis [87], human-computer interaction [88], automotive assistance [89], AR/VR [90, 91]. However, developing a robust unified learning-based gaze estimation model requires large amounts of data from multiple subjects with precise gaze annotations [17, 15]. Collecting and annotating such an appropriate dataset is complex and expensive. To overcome this challenge, several methods have been proposed to redirect gaze directions [92, 14, 18, 17, 15] in real images with assigned directional values to obtain and augment training data. Some works focused on generating eye images with new gaze directions by either 1) estimating warping maps [14, 15] to interpolate pixel values or 2) using encoder-generator pairs to generate redirected eye images [92, 18].

ST-ED [17] was the first work to extend high-accuracy gaze redirection from eye images to face images. By disentangling several attributes, including person-specific appearance, it can explicitly control gaze directions and head orientations. However, due to the design of the encoder-decoder structure and limited ability to maintain appearance features by a 1×1024 projected appearance embedding, ST-ED generates low-resolution (128×128) images with restricted face range (no hair area), which narrows the application ranges and scenarios of gaze redirection.

As for latent space manipulation for face editing tasks, large amounts of works [33, 30, 31, 34, 37, 32] were proposed to modify latent vectors in predefined latent spaces (W [93], W^+ [94] and S [95]). Latent vectors in these latent spaces can work with StyleGAN [93, 29] to generate high-quality and high-fidelity face images with desired attribute editing. Among these methods, Wu *et.al* [95] proposed the latent space S working with StyleGAN, which achieved only one degree-of-freedom gaze redirection by modifying a certain channel of latent vectors in S by an uninterpreted value instead of pitch and yaw values of gaze directions.

Considering these, we proposed a new method, called ReDirTrans, to achieve latent-to-latent translation for redirecting gaze directions and head orientations in high-resolution full-face images based on assigned directional values. Specifically, we designed a framework to project input latent vectors from a latent space into the aimed-attribute-only embedding space for an interpretable redirection process. This embedding space consists of estimated pseudo conditions and embeddings of aimed attributes, where conditions describe deviations from the canonical status and embeddings are the 'carriers' of the conditions. In this embedding space, all transformations are implemented by rotation matrices multiplication built from pitch and yaw values, which can make the redirection process more interpretable and consistent. After the redirection process, the original embeddings and redirected ones are both decoded back to the initial latent space as the residuals to modify the input latent vectors by subtraction and addition operations. These operations represent removing the old state and adding a new one, respectively. ReDirTrans only focuses on transforming embeddings of aimed attributes and achieves status replacement by the residuals outputted from weight-sharing deprojectors. ReDirTrans does not project or deproject other attributes with information loss; and it does not affect the distribution of input latent vectors. Thus ReDirTrans can also work in a predefined feature space with a fixed pretrained encoder-generator pair for the redirection task in desired-resolution images. In summary, our contributions are as follows:

- A latent-to-latent framework, *ReDirTrans*, which projects latent vectors to an embedding space for an interpretable redirection process on aimed attributes and maintains other attributes, including appearance, in initial latent space with no information loss caused by projection-deprojection processes.
- A portable framework that can seamlessly integrate into a pretrained GAN inversion pipeline for high-accuracy redirection of gaze directions and head orientations, without the need for any parameter tuning of the encoder-generator pairs.
- A layer-wise architecture with learnable parameters that works with the fixed pretrained StyleGAN and achieves redirection tasks in high-resolution full-face images through *ReDirTrans-GAN*.

3.2 Related Works

3.2.1 Gaze and Head Redirection

Methods for redirecting gaze directions can be broadly classified into two categories: warping-based methods and generator-based methods. Deepwarp [13, 96] presented a deep network to learn warping maps between pairs of eye images with different gaze directions, which required large amounts of data with annotations. Yu *et al.* [14] utilized a pretrained gaze estimator and synthetic eye images to reduce the reliance on annotated real data. Yu *et al.* [15] further extended the warping-based methods in an unsupervised manner by adding a gaze representation learning network. As for the generator-based methods, He *et al.* [18] developed a GAN-based network for generating eye images with new gaze directions. FAZE [16] proposed an encoder-decoder architecture to transform eye images into latent vectors for redirection with rotation matrix multiplication, and then decode the edited ones back to the synthetic images with new gaze directions. ST-ED [17] further extended the encoder-decoder pipeline from gaze redirection only to both head and gaze redirection over full face images by disentangling latent vectors, and achieving precise redirection performance. However, ST-ED generates images with a restricted face range (no hair area) with a size of 128×128. We further improve the redirection task by covering the full face range with 1024×1024 resolution.

3.2.2 Latent Space Manipulation

Numerous methods investigated the latent space working with StyleGAN [93, 29] to achieve semantic editing in image space due to its meaningful and highly disentangled properties. As for the supervised methods, InterFaceGAN [33] determined hyperplanes for the corresponding facial attribute editing based on provided labels. StyleFlow [34] proposed mapping a sample from a prior distribution to a latent distribution conditioned on the target attributes estimated by pretrained attribute classifiers. Given the unsupervised methods, GANSpace [35], SeFa [36] and TensorGAN [37] leveraged principal components analysis, eigenvector decomposition and higher-order singular value decomposition to discover semantic directions in latent space, respectively. Other self-supervised methods proposed mixing of latent codes from other samples for local editing [38, 39], or incorporating the language model CLIP [40] for text-driven editing [41].

3.2.3 Domain Adaptation for Gaze Estimation

Domain gaps among different datasets restrict the application range of pretrained gaze estimation models. To narrow the gaps, a few domain adaptation approaches [57, 97] were proposed for the generic regression task. SimGAN [98] proposed an unsupervised domain adaptation method for narrowing the gaps between real and synthetic eye images. HGM [99] designed a unified 3D eyeball model for eye image synthesis and cross-dataset gaze estimation. PnP-GA [68] presented a gaze adaptation framework for generalizing gaze estimation in new domains based on collaborative learning. Qin *et al.* [100] utilized 3D face reconstruction to rotate head orientations together with changed eye gaze accordingly to enlarge overlapping gaze distributions among datasets. These adaptation methods typically rely on restricted face or eye images to alleviate interference from untargeted attributes. Our work incorporates the redirection task in a predefined meaningful feature space with controllable attributes to achieve high-resolution and full-face redirection.

3.3 Method

3.3.1 Problem Statements

Our goal is to train a conditional latent-to-latent translation module for face editing with physical meaning, and it can work either with a trainable or fixed encoder-generator pair. This editing module first transforms input latent vectors from encoded feature space $\mathcal{F}$ to an embedding space $\mathcal{Z}$ for redirection in an interpretable manner. Then it deprojects the original and redirected embeddings back to the initial feature space $\mathcal{F}$ for editing input latent vectors. The edited latent vectors are fed into a generator for image synthesis with the desired status of aimed facial attributes. The previous GAN-based work [33, 37, 101] achieved a certain facial attribute editing with a global latent residual multiplied by a scalar without physical meaning to describe the relative deviation from the original status. To make the whole process interpretable and achieve redirection directly based on the new gaze directions or head poses, we follow the assumption proposed by [17], where the dimension of an embedding is decided by the corresponding attribute's degree of the freedom (DoF) and redirection process is achieved by the rotation matrices multiplication. Thus the transformation equivariant mappings can be achieved between the embedding space $\mathcal{Z}$ and image space. To be specific, normalized gazes or head poses

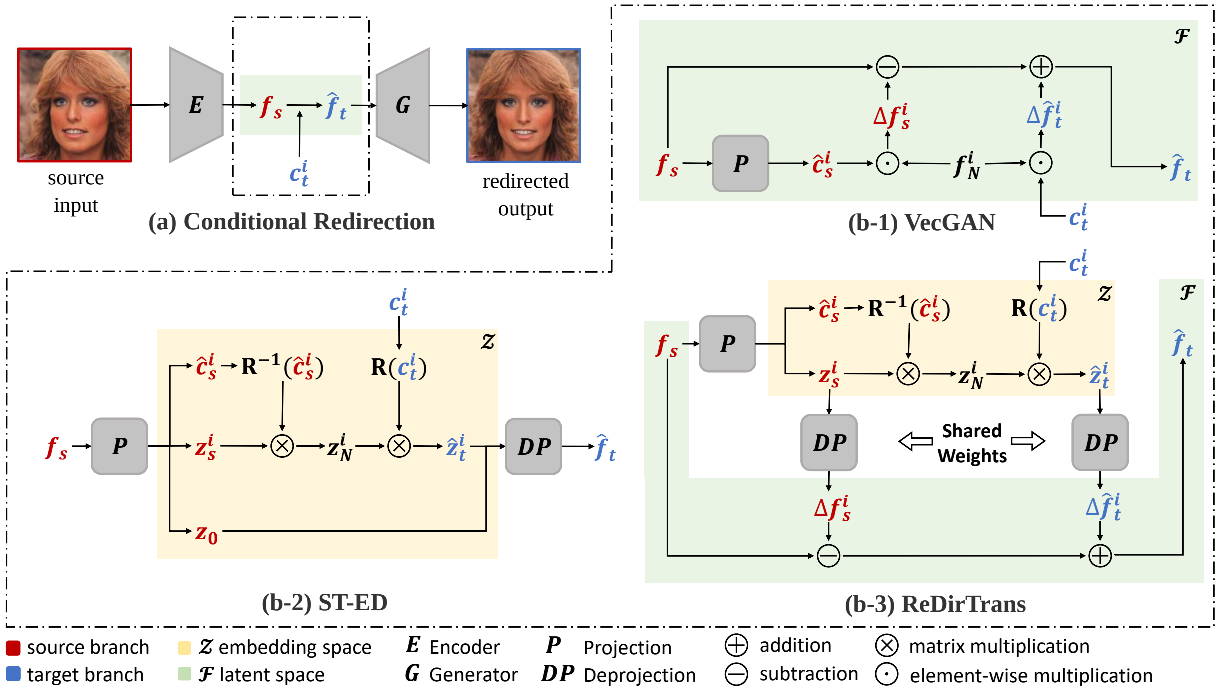


Figure 3.1: Conditional redirection pipeline and comparison among different redirectors. (a) We first encode the input image into the latent vector. Given the provided conditions, we modify the latent vector and send it to a generator for image synthesis with only aimed attribute redirection. (b) We compared our proposed redirector with two state-of-the-art methods: (b-1) VecGAN achieves editing in feature space $\mathcal{F}$ given projected conditions from latent vectors with a global direction f_N^i. (b-2) ST-ED projects the latent vector into conditions and embeddings of aimed attributes, and one appearance-related high dimensional embedding z_0 in embedding space $\mathcal{Z}$. After interpretable redirection process in space $\mathcal{Z}$, all embeddings are concatenated and projected back to space $\mathcal{F}$. (b-3) Our proposed ReDirTrans projects the latent vector into conditions and embeddings of aimed attributes only. After an interpretable redirection process, both original and redirected embeddings are deprojected back to initial space $\mathcal{F}$ as residuals. These residuals modify the input latent vector by subtraction and addition operations, which represent the initial status removal and the new status addition, respectively. This approach efficiently reduces effects on other attributes (especially the appearance related information) with fewer parameters than ST-ED.

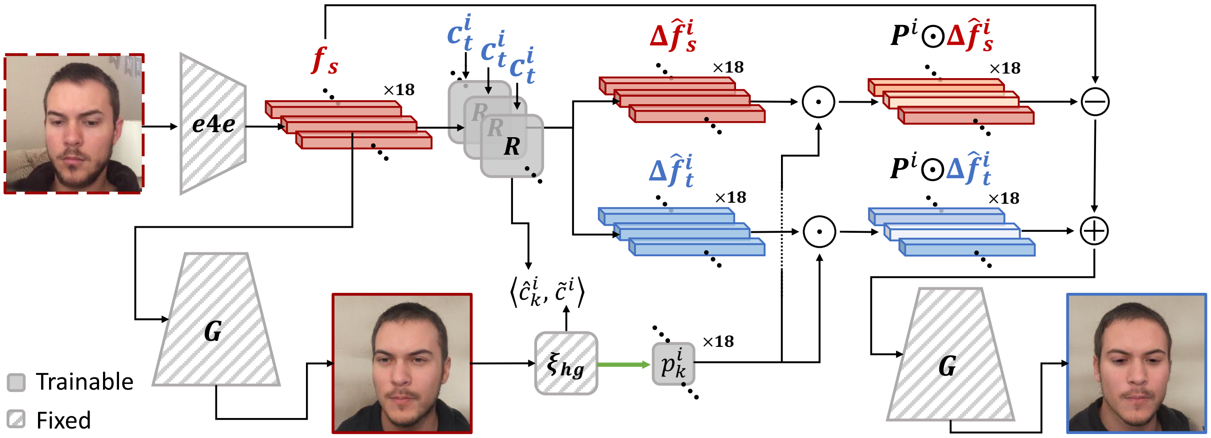


Figure 3.2: Structure of layer-wise ReDirTrans with fixed e4e and StyleGAN. Given the multi-layer representation of latent vectors in $W^+ \subseteq \mathbb{R}^{18\times512}$, we feed each layer into an independent ReDirTrans for redirection task given the provided target condition c_t^i, where i represents a certain attribute. We calculate errors between estimated conditions $\hat{c}_k^i, k \in [1,18]$ from multiple ReDirTrans and the pseudo condition $\tilde{c}^i$ estimated from the inverted image for supervising the trainable weights learning (green arrow) based on the Layer-wise Weights Loss described in Eq. 3.10 to decide which layers should contribute more to a certain attribute redirection. Given the estimated weights and initial latent vectors f_s, we can acquire the final disentangled latent vector $\hat{f}_t'$ based on Eq. 3.2 for redirected samples synthesis.

can be represented by a two-dimensional embedding with the pitch and yaw as the controllable conditions. The embeddings can be edited (multiplied) by the rotation matrices built from the pitch and yaw for achieving redirection (rotation) of aimed attributes in image space accordingly through our proposed redirector.

3.3.2 Redirector Architecture

ST-ED is one of the state-of-the-art architectures for gaze and head poses redirection over face images [17] shown in Fig. 3.1 (b-2). ST-ED projects the input latent vector f to non-varying embeddings z^0 and M varying ones with corresponding estimated conditions $\{(z^i, \hat{c}^i)|i \in [1,M]\}$, where $\hat{c}^i$ describes the estimated amount of deviation from the canonical status of the attribute i, and it can be compared with the ground truth c^i for the learning of conditions from latent vectors. The non-varying embedding z^0 defines subject's appearance, whose dimension is much larger (over twenty times larger in ST-ED) than other varying embeddings. It is inefficient to project input latent vectors into a high-dimensional embedding to maintain non-varying information such as identity, hairstyle, etc. Thus, we propose a new redirector architecture, called *ReDirTrans*, shown in Fig. 3.1 (b-3), which transforms the source latent vector f_s to the embeddings of aimed attributes through the projector $\boldsymbol{P}$ and redirects them given the newly provided target conditions c_t. Then we deproject both original embeddings $\mathbf{z_s}$ and redirected embeddings $\hat{\mathbf{z}}_\mathbf{t}$ back to the feature space $\mathcal{F}$ through the weights-sharing deprojectors $\boldsymbol{DP}$ to acquire latent residuals. These residuals contain source and target status of aimed attributes, denoted as Δf_s^i and $\Delta \hat{f}_t^i$, respectively. Inspired by addition and subtraction [33, 101] for face editing in feature space $\mathcal{F}$, the edited latent vector is

$$\hat{f}_t = f_s + \sum_{i=1}^{M} (-\Delta f_s^i + \Delta \hat{f}_t^i), i \in [1, M], \tag{3.1}$$

where the subtraction means removing source status and the addition indicates bringing in new status. The projector $\boldsymbol{P}$ ensures that the dimension of embeddings can be customized based on the degrees of freedom of desired attributes, and the transformations can be interpretable with physical meanings. The deprojector $\boldsymbol{DP}$ enables the original and edited features in the same feature space, allowing ReDirTrans to be compatible with pretrained encoder-generator pairs that are typically trained together without intermediate (editing) modules. ReDirTrans reduces parameters by skipping projection (compression) and deprojection (decompression) of the features that are not relevant to the desired attributes, but vital for final image synthesis.

3.3.3 Predefined Feature Space

Except for the *trainable* encoder-decoder (or -generator) pair to learn a specific feature space for redirection task as ST-ED did, ReDirTrans can also work in the predefined feature space to coordinate with *fixed*, *pretrained* encoder-generator pairs. For our implementation, we chose the $W^+ \in \mathbb{R}^{18 \times 512}$ feature space [94], which allows us to utilize StyleGAN [29] for generating high-quality, high-fidelity face images. We refer to this implementation as *ReDirTrans-GAN*. Considering multi-layer representation of the latent vector [94] and its semantic disentangled property between different layers [34, 35] in W^+ space, we proposed layer-wise redirectors, shown in Fig. 3.2, rather than using a single ReDirTrans to process all (18) layers of the latent vector. To largely reduce the interference between different layers during redirection, we assume that if one attribute's condition can be estimated from certain layers with less errors than the others, then we can 'modify' these certain layers with higher weights $p_k^i, k \in [1,18]$ than others to achieve redirection of the corresponding attribute i only. $\boldsymbol{P}^i = [p_1^i, \cdots, p_{18}^i]^T \in \mathbb{R}^{18\times 1}$, as part of network parameters, is trained given the loss function described in Eq. 3.10. The final disentangled latent vectors after redirection is

$$\hat{f}_{t,d} = f_s + \sum_{i=1}^{M} \boldsymbol{P}^i \odot (-\Delta f_s^i + \Delta \hat{f}_t^i), i \in [1, M], \tag{3.2}$$

where $\odot$ means element-wise multiplication and $(-\Delta f_s^i + \Delta \hat{f}_t^i) \in \mathbb{R}^{18\times 512}$. One **challenge** regarding the predefined feature space comes from the inversion quality. There exist attribute differences between input images and inverted results, shown in Fig. 3.4 and 3.6, which means that the conditions in source images cannot be estimated from source latent vectors. To solve this, instead of using conditions from source images, we utilized estimated conditions from the inverted images, which ensures the correctness and consistence of conditions learning from latent vectors.

3.3.4 Training Pipeline

Given a pair of source and target face images, I_s and I_t from the same person, we utilize an encoder to first transform I_s into the feature space $\mathcal{F}$, denoted as f_s. We further disentangle f_s into the gaze-direction-related embedding z_s^1 and the head-orientation-related embedding z_s^2 with corresponding estimated conditions: $\hat{c}_s^1$ and $\hat{c}_s^2$ by the projector $\boldsymbol{P}$. Then we build rotation matrices using the pitch and yaw from estimated conditions $(\hat{c}_s^1, \hat{c}_s^2)$ and target conditions (c_t^1, c_t^2) to normalize embeddings and redirect them to the new status, respectively:

$$\begin{aligned} &\text{Normalization: } z_N^i = \mathbf{R}^{-1}(\hat{c}_s^i) \cdot z_s^i, \\ &\text{Redirection: } \quad \hat{z}_t^i = \mathbf{R}(c_t^i) \cdot z_N^i, \end{aligned} \tag{3.3}$$

where $i \in \{1,2\}$, representing gaze directions and head orientations, respectively, and z_N^i denotes the normalized embedding of the corresponding attribute. We feed the original embedding z_s^i and the modified embedding $\hat{z}_t^i$ into the weights-sharing deprojectors ***DP*** to transform these embeddings back to the feature space $\mathcal{F}$ as the residuals. Given these residuals, we implement subtraction and addition operations over f_s as described in Eq. 3.1 (or Eq. 3.2) to acquire the edited latent vector $\hat{f}_t$ (or $\hat{f}_{t,d}$), which is sent to a generator for synthesizing redirected face image $\hat{I}_t$. $\hat{I}_t$ should have the same gaze direction and head orientation as I_t.

3.3.5 Learning Objectives

We supervise the relationship between the generated image $\hat{I}_t$ and the target image I_t with several loss functions: pixel-wise reconstruction loss, LPIPS metric [102] and attributes loss by a task-related pretrained model.

$$\mathcal{L}_{rec}(\hat{I}_t, I_t) = ||\hat{I}_t - I_t||_2, \tag{3.4}$$

$$\mathcal{L}_{LPIPS}(\hat{I}_t, I_t) = ||\psi(\hat{I}_t) - \psi(I_t)||_2, \tag{3.5}$$

$$\mathcal{L}_{att}(\hat{I}_t, I_t) = \langle \xi_{hg}(\hat{I}_t), \xi_{hg}(I_t) \rangle, \tag{3.6}$$

where $\psi(\cdot)$ denotes the perceptual feature extractor [102], $\xi_{hg}(\cdot)$ denotes the HeadGazeNet [17] to estimate the gaze and head pose from images and $\langle u, v \rangle = \arccos \frac{u \cdot v}{||u|| \cdot ||v||}$.

Identity Loss. Identity preservation after redirection is critical for the face editing task. Considering this, we calculate the cosine similarity of the identity-related features between the source image and the redirected image:

$$\mathcal{L}_{ID}(\hat{I}_t, I_s) = 1 - \langle \phi(\hat{I}_t), \phi(I_s) \rangle, \tag{3.7}$$

where $\phi(\cdot)$ denotes the pretrained ArcFace [103] model.

Label Loss. We have ground truth of gaze directions and head orientations, which can guide the conditions learning from the input latent vectors for the normalization step:

$$\mathcal{L}_{lab}(\hat{c}^i_s, c^i_s) = \langle \hat{c}^i_s, c^i_s \rangle, \;\; i \in \{1, 2\}. \tag{3.8}$$

Embedding Loss. The normalized embeddings only contain the canonical status of the corresponding attribute after the inverse rotation applied to the original estimated embeddings, shown in Fig. 3.1. Thus the normalized embeddings given a certain attribute across different samples within batch B should be consistent. To reduce the number of possible pairs within a batch, we utilize the first normalized embedding $z^i_{N,1}$ as the basis:

$$\mathcal{L}_{emb} = \frac{1}{B-1} \sum_{j=2}^{B} \langle z^i_{N,1}, z^i_{N,j} \rangle, \;\; i \in \{1, 2\}. \tag{3.9}$$

Layer-wise Weights Loss. This loss is specifically designed for the W^+ space to decide the weights p_i of which layer should contribute more to the aimed attributes editing. Firstly, we calculate the layer-wise estimated conditions $\hat{c}^i_k$ and calculate estimated pseudo labels $\tilde{c}^i$. Secondly, we have layer-wise estimated label errors by $\langle \hat{c}^i_k, \tilde{c}^i \rangle$. Lastly, we calculate the cosine similarity between the reciprocal of label errors and weights of layers as the loss:

$$\mathcal{L}_{prob} = \langle \{p_k\}, \{\frac{1}{\langle \hat{c}^i_k, \tilde{c}^i \rangle}\} \rangle, k \in [1, K], i \in \{1, 2\}, \tag{3.10}$$

where K is the number of layers for editing.

Full Loss. The combined loss function for supervising the redirection process is:

$$\begin{aligned} \mathcal{L} = \lambda_r \mathcal{L}_{rec} + \lambda_L \mathcal{L}_{LPIPS} + \lambda_{ID} \mathcal{L}_{ID} + \lambda_a \mathcal{L}_{att} \\ + \lambda_l \mathcal{L}_{lab} + \lambda_e \mathcal{L}_{emb} + \lambda_p \mathcal{L}_{prob}, \end{aligned} \tag{3.11}$$

where $\mathcal{L}_{LPIPS}$ and $\mathcal{L}_{prob}$ are utilized only when the pretrained StyleGAN is used as the generator.

3.4 Experiments

3.4.1 Datasets

GazeCapture [60] is the largest public gaze-related full-face dataset including $1,474$ participants with over two million frames taken under unconstrained scenarios. We utilize its training subset to train the redirector and evaluate the redirection precision with its test subset. MPIIFaceGaze [64] is a widely used benchmark dataset for the in-the-wild gaze estimation task. It includes $37,667$ full-face images captured from 15 participants with varied head orientations, multiple gaze directions and different illuminations. We utilize this dataset to evaluate the cross-dataset redirection performance. CelebA-HQ [104] is a high-quality version of CelebA [105] that consists of $30,000$ images at 1024×1024 resolution. We utilize this dataset for evaluating the cross-dataset qualitative redirection performance.

3.4.2 Implementation Details

Preprocessing Steps.

1) *ReDirTrans*: We preprocessed the image data to acquire a 128×128 restricted range of face images aligned with key points of the nose and eyes, followed by [17].

2) *ReDirTrans-GAN*: We preprocessed the image data to acquire 256×256 full-face images aligned with key points of the mouth and eyes. We utilized reflective padding to the blank

Table 3.1: The architecture of the projector and deprojector in **ReDirTrans**. **P** denotes projector and **DP** denotes deprojector.

ReDirTrans		Layers/Blocks

P	Pseudo Label Branch	FC(3072, 96, w/bias), LeakyReLU()
		FC(96, 4, w/bias), pi/2*Tanh()

	Embedding Branch	FC(3072, 3072, w/bias), LeakyReLU()
		FC(3072, 96, w/bias)

DP		FC(96, 1024, w/bias), LeakyReLU()
		FC(1024, 3072, w/bias)

Table 3.2: The architecture of the projector and deprojector in **ReDirTrans-GAN**. **P** denotes projector and **DP** denotes deprojector.

ReDirTrans-GAN		Layers/Blocks

P	Pseudo Label Branch	FC(512, 64, w/bias), LeakyReLU()
		FC(64, 4, w/bias), pi/2*Tanh()

	Embedding Branch	FC(512, 128, w/bias), LeakyReLU()
		FC(128, 96, w/bias)

DP		FC(96, 256, w/bias), LeakyReLU()
		FC(256, 512, w/bias)

areas after alignment and then covered these areas with Gaussian blur, followed by the work in [104].

Projector-Deprojector. Since the inputs to the projector have already been the decoded latent vectors from images, we utilized several fully connected modules as the architectures of the projector-deprojector.

1) *ReDirTrans*: Unlike ST-ED projecting the input latent vector into nine attribute embeddings, our proposed ReDirTrans only projected it into the aimed attribute (gaze directions and head orientations) embeddings. The size of the estimated label and embedding of one attribute are 2 and 3×16, respectively. The details are illustrated in Table 3.1.

2) *ReDirTrans-GAN*: As for the ReDirTrans-GAN, the main difference comes from the size of latent vectors in latent space $\mathcal{F}$. The details are shown in Table 3.2.

Encoder-Generator and Loss Functions.

1) *ReDirTrans*: ST-ED proposed the architecture of encoder-decoder pair given the DenseNet [106] architecture, for 128×128 output. As for the decoder, the convolutional layers were replaced by the transposed convolutional layers and the average-pooling layers. The detailed encoder-decoder structure is illustrated in [17]. To ensure the generation quality, we utilized a PatchGAN [107] discriminator with corresponding adversarial loss as proposed in ST-ED during the training.

2) *ReDirTrans-GAN*: Since both e4e and StyleGAN were pretrained and fixed during the training, the image discriminator mentioned above was no longer used. Instead, to maintain the perceptual quality and editability of latent codes after redirection as the original latent codes encoded by e4e, we kept utilizing the e4e proposed delta-regularization loss $\mathcal{L}_{d-reg}$ and the adversarial loss $\mathcal{L}_{adv}$ by a latent discriminator [108]. Noted that we applied these two loss functions to the modified latent vectors after redirection to maintain the editability of e4e encoded latent vectors.

Gaze and Head Pose Estimation Network. During training, we need a pretrained gaze and head pose estimation network $\xi_{hg}(\cdot)$ as the estimator to supervise the redirection process. During the evaluation, we require another different external pretrained gaze and head pose estimation network $\xi'_{hg}(\cdot)$, which is unseen during training, to evaluate the consistency of the aimed attributes between redirected and target samples. We followed the pipeline proposed by ST-ED, which utilized a VGG-16-based $\xi_{hg}(\cdot)$ [109] and a ResNet50-based $\xi'_{hg}(\cdot)$ [110].

1) *ReDirTrans*: We retrained and employed the VGG-16-based $\xi_{hg}(\cdot)$ and ResNet50-based $\xi'_{hg}(\cdot)$ as the gaze and head pose estimators, given the architectures and training parameters illustrated in [17].

2) *ReDirTrans-GAN*: To fit the different sizes of input (256×256) and output images (1024×1024) with the full face range when training and evaluating ReDirTrans-GAN, we downsampled the output images to 256×256. The fully connected modules after the convolutional

Table 3.3: Architecture of the VGG-16-based gaze direction and head orientation estimation network, $\xi_{hg}(\cdot)$.

Nr.	**layers / blocks**
0	VGG-16 Conv layers
1	AvgPool2d(size=4, stride=4)
2	FC(2048, 128, w/bias), LeakyReLU()
3	FC(128, 64, w/bias), LeakyReLU()
4	FC(64, 4, w/bias), $0.5\pi \cdot \tanh()$

Table 3.4: Architecture of the ResNet50-based gaze direction and head orientation estimation network, $\xi'_{hg}(\cdot)$.

Nr.	**layers / blocks**
0	ResNet-50 Conv layers, stride of MaxPool2d=1
1	FC(2048, 4, w/bias)

part of $\xi_{hg}(\cdot)$ and $\xi'_{hg}(\cdot)$ were modified accordingly for different input sizes compared with the ST-ED version. The detailed structures of $\xi_{hg}(\cdot)$ and $\xi'_{hg}(\cdot)$ are shown in Table 3.3 and Table 3.4, respectively.

3.4.3 Training Hyperparameters

1) *ReDirTrans*: We trained ReDirTrans and the encoder-decoder pair with the same hyperparameters as ST-ED [17] by using over 1.4×10^6 full-face images from GazeCapture Training subset.

2) *ReDirTrans-GAN*: We randomly chose $10,000$ images from the GazeCapture training subset to train ReDirTrans-GAN since both the encoder and generator are fixed and pretrained. The number of epochs is 2 with a batch size of 2. The initial learning rate is 10^{-4} and is decayed by 0.8 every $3,000$ iterations. The optimizer is Adam [78] with the default momentum value of $\beta_1 = 0.9, \beta_2 = 0.999$.

The loss weights are $\lambda_r = 8$, $\lambda_L = 8$, $\lambda_{ID} = 5$, $\lambda_a = 1$, $\lambda_l = 5$, $\lambda_e = 2$, $\lambda_p = 10$, $\lambda_{d-reg} = 0.0002$, $\lambda_{adv} = 2$, where λ_{d-reg} and λ_{adv} are the weights of delta-regularization loss $\mathcal{L}_{d-reg}$ and the adversarial loss $\mathcal{L}_{adv}$, respectively.

3.4.4 Redirection Step

We applied rotation matrices built by the pitch and yaw to the estimated embeddings for redirection purposes.

$$\begin{bmatrix} \cos\phi^i & 0 & \sin\phi^i \\ 0 & 1 & 0 \\ -sin\phi^i & 0 & \cos\phi^i \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta^i & -\sin\theta^i \\ 0 & \sin\theta^i & cos\theta^i \end{bmatrix}, i \in \{1,2\} \tag{3.12}$$

where ϕ represents yaw and θ represents pitch, and index i represents gaze directions and head orientations, respectively.

3.4.5 Evaluation Criteria

We follow metrics utilized by ST-ED [17] to evaluate different redirectors' performance.

Redirection Error. We measure the redirection accuracy in image space by a pre-trained ResNet-50 based [110] head pose and gaze estimator ξ'_{hg}, which is unseen during the training. Given the target image I_t and the generated one $\hat{I}_t$ redirected by conditions of I_t, we report the angular error between $\xi'_{hg}(\hat{I}_t)$ and $\xi'_{hg}(I_t)$ as the redirection errors.

Disentanglement Error. We quantify the disentanglement error by the condition's fluctuation range of one attribute when we redirect the other one. The redirection angle ε follows $\mathcal{U}(-0.1\pi, 0.1\pi)$. For example, when we redirect the head pose of the generated image $\hat{I}_t$ by ε and generate a new one $\hat{I}'_t$, we calculate the angular error of the estimated gaze directions between $\xi'_{hg}(\hat{I}_t)$ and $\xi'_{hg}(I'_t)$.

LPIPS. LPIPS is able to measure the distortion [102] and image similarity in gaze directions [18] between images, which is applied to evaluate the redirection performance.

Table 3.5: Within-dataset quantitative comparison (GazeCapture test subset) between different methods for redirecting head orientations and gaze directions. (Lower is better). **Head (Gaze) Redir** denotes the redirection accuracy in degree between the redirected image and the target image given head orientations (gaze directions). **The Head (Gaze) Effect** denotes the effects in degree on one attribute when we redirect the other. † denotes copied results from [17]. Other methods are retrained given previous papers.

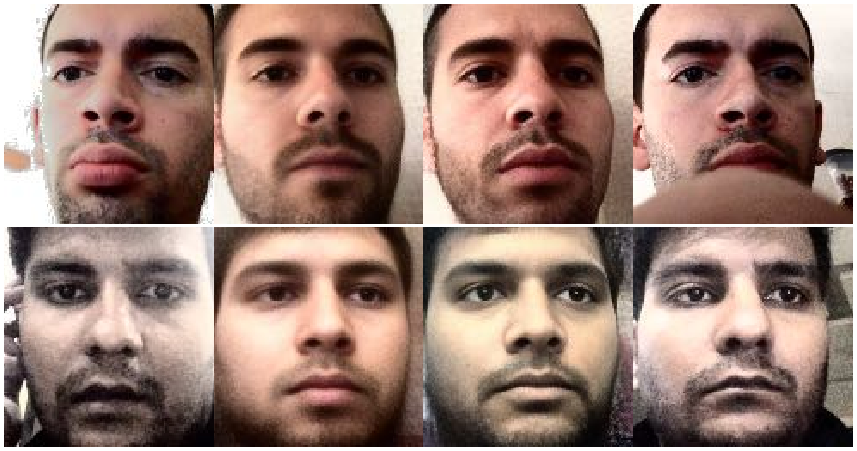
	Gaze Redir	Head Redir	Gaze Effect	Head Effect	LPIPS
StarGAN † [111]	4.602	3.989	0.755	3.067	0.257
He *et al.* † [18]	4.617	1.392	0.560	3.925	0.223
VecGAN [101]	2.282	0.824	0.401	2.205	**0.197**
ST-ED [17]	2.385	0.800	**0.384**	2.187	0.208
ReDirTrans	**2.163**	**0.753**	0.429	**2.155**	**0.197**

Table 3.6: Cross-dataset quantitative comparison (MPIIFaceGaze) between different methods for redirecting head orientations and gaze directions. (Lower is better). Notations are the same as them in the Table 3.5. † denotes copied results from [17]. Other methods are retrained given previous papers.

	Gaze Redir	Head Redir	Gaze Effect	Head Effect	LPIPS
StarGAN † [111]	4.488	3.031	0.786	2.783	0.260
He *et al.* † [18]	5.092	1.372	0.684	3.411	0.241
VecGAN [101]	2.670	1.242	0.391	1.941	0.207
ST-ED [17]	**2.380**	1.085	**0.371**	**1.782**	0.212
ReDirTrans	**2.380**	**0.985**	0.391	**1.782**	**0.202**

3.4.6 Redirectors in Learnable Latent Space

We compared quantitative performance of different redirectors, which were trained along with the trainable encoder-decoder pair designed by ST-ED on 128 × 128 images with restricted face ranges, given the criteria proposed in Sec. 3.4.5. Table 3.5 and Table 3.6 present within-dataset and cross-dataset performance, respectively. From these tables, we observe that our proposed ReDirTrans achieved more accurate redirection and better LPIPS compared with other state-of-the-art methods by considering the extra embedding space $\mathcal{Z}$ for redirecting embeddings of aimed attributes only and maintaining other attributes including the appearance-related information in the original latent space $\mathcal{F}$. ST-ED [17] projected input latent vectors into nine



(a) Input (b) ST-ED (c) ReDirTrans (d) Target

Figure 3.3: Qualitative comparison of ReDirTrans and ST-ED in GazeCapture. ReDirTrans preserves more facial attributes, such as lip thickness and sharpness of the beard.

embeddings including the non-varying embedding z^0. This appearance-related high dimensional embedding z^0 requires more parameters than ReDirTrans during projection. After redirecting the embeddings of aimed attributes, ST-ED deprojected a stack of z^0, redirected embeddings, and rest unvaried embeddings of other attributes back to the feature space for decoding. This projection-deprojection process of non-varying embedding z^0 results in loss of appearance and worse LPIPS, as depicted in Fig. 3.3. VecGAN [101] was proposed to edit the attributes only within the feature space by addition and subtraction operations. Since there is no projection-deprojection process, given the original latent code, LPIPS performance is better than ST-ED. However, as no extra embedding space was built for the aimed attributes editing, both redirection accuracy and the disentanglement process were affected.

3.4.7 Redirectors in Predefined Latent Space

Except for using the trainable encoder-decoder pair of ST-ED, we also implemented our proposed ReDirTrans within a predefined feature space W^+ to achieve redirection task in full face images with desired resolution. We utilized e4e [30] as the pre-trained encoder, which can transform input images into latent vectors in W^+, and we chose StyleGAN2 [29]

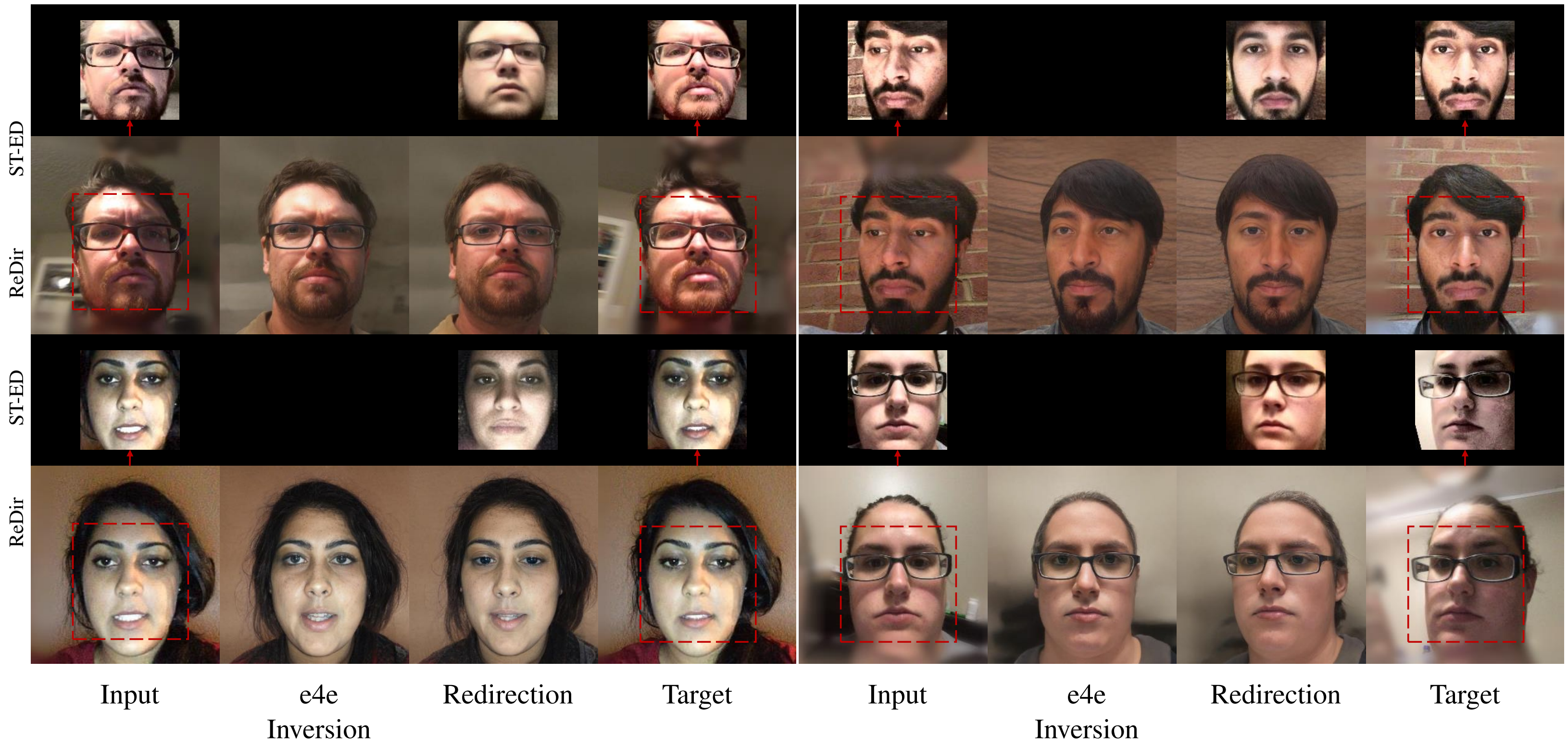


Figure 3.4: Qualitatively comparisons between ST-ED and ReDirTrans-GAN. Red boxes represent different face covering ranges. ‘ReDir’ denotes ReDirTrans-GAN.

Table 3.7: Learning-based gaze estimation errors (in degrees) in GazeCapture and MPIIFaceGaze with or without redirected data augmentation. *Q%* represents percent of labeled data in 10,000 images for training ReDirTrans. 'Raw' or 'Aug' mean training the gaze estimator with real data only or with real and redirected data.

Q%	GazeCapture		MPIIFaceGaze	
	Raw ↓	Aug ↓	Raw ↓	Aug ↓
25	5.875	**5.238**	8.607	**7.096**
50	4.741	**4.506**	6.787	**6.113**
75	4.308	**4.200**	6.165	**5.767**

as the pre-trained generator to build ReDirTrans-GAN. Fig. 3.4 shows qualitative comparison between ST-ED and ReDirTrans-GAN in the GazeCapture test subset with providing target images from the same subject. ReDirTrans-GAN successfully redirected gaze directions and head orientations to the status provided by target images while maintaining the same appearance patterns with 1024×1024 full face images. Due to the design of ReDirTrans, which maintains unrelated attributes and appearance information in the initial latent space instead of going through the projection-deprojection process, ReDirTrans-GAN keeps more facial attributes such as expressions, mustaches, bangs compared with ST-ED. Fig. 3.5 presents qualitative results with assigned conditions (pitch and yaw of gaze directions and head orientations) in CelebA-HQ [104]. ReDirTrans-GAN can achieve out-of-domain redirection tasks in predefined feature space while maintaining other facial attributes.

3.4.8 Data Augmentation

To solve data scarcity of the downstream task: learning-based gaze estimation, we utilized redirected samples with assigned gaze directions and head orientations to augment training data. We randomly chose $10,000$ images from the GazeCapture training subset to retrain ReDirTrans-GAN with using only $Q\%$ ground-truth labels of them. The HeadGazeNet $\xi_{hg}(\cdot)$ was also retrained given the same $Q\%$ labeled data and $Q \in \{25, 50, 75\}$. Then we utilized ReDirTrans-

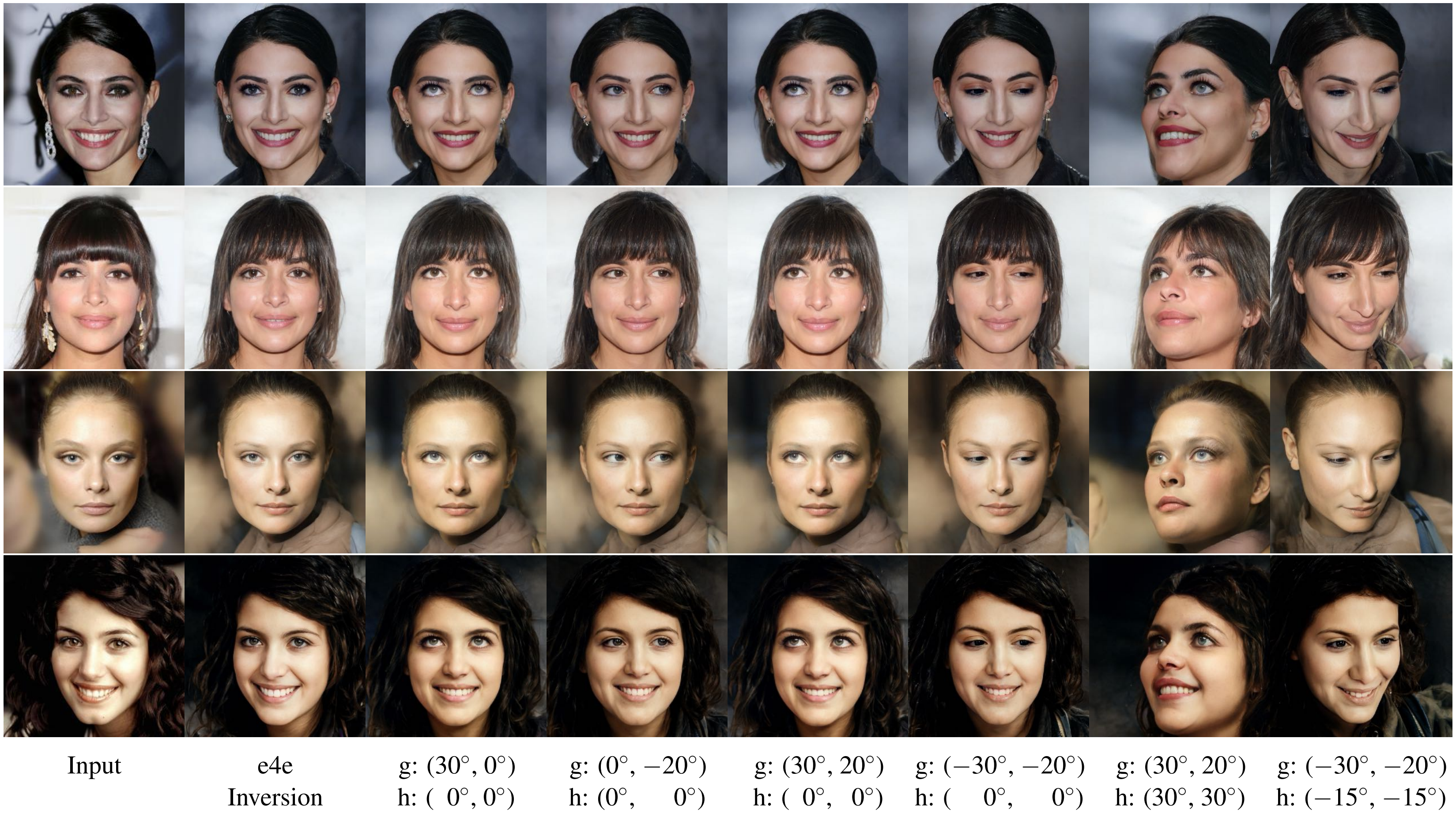


Figure 3.5: Redirection results given assigned (pitch, yaw) conditions of gaze directions (g) and head orientations (h). The first two columns are input and inversion results with e4e [30] and StyleGAN [29]. The following columns are redirected samples with assigned redirection values based on the latent code estimated from e4e.

GAN to generate redirected samples given provided conditions over $Q\%$ labeled real data and combined the real and redirected data as an augmented dataset with size $2 \times 10,000 \times Q\%$ for training a gaze estimator. Table 3.7 presented within-dataset and cross-dataset performance and demonstrated consistent improvements for the downstream task given redirected samples as data augmentation.

3.4.9 Challenge in Predefined Feature Space

One challenge for redirection tasks in predefined feature space comes from inconsistency between input and inverted images, mentioned in Sec. 3.3.3. We can observe that the existing gaze differences between input and inverted images in Fig. 3.4. In some cases, the gaze directions are changed after GAN inversion, which means that the encoded latent codes do not necessarily keep the original gaze directions. Thus, instead of using provided gaze directions of input images during the training, we utilized estimated gaze directions from inverted results to correctly normalize the gaze and head pose to the canonical status. This process ensures correctness when further new directions are added, making the training process more consistent.

3.4.10 Gaze Correction

ReDirTrans can correct gaze directions of inverted results by viewing input images as the target ones. e4e guarantees high editability, which is at the cost of inversion performance [30]. Fig. 3.6 shows several samples which failed to maintain input images' gaze directions even by the ReStyle encoder [31], which iteratively updates the latent codes given the differences between the input and inverted results.

Table 3.8 and Table 3.9 further present within- and cross-dataset evaluation performance for gaze correction tasks. e4e inversion results can maintain gaze directions and head orientations better in CelebA-HQ than GazeCapture since samples in CelebA-HQ have much less varied

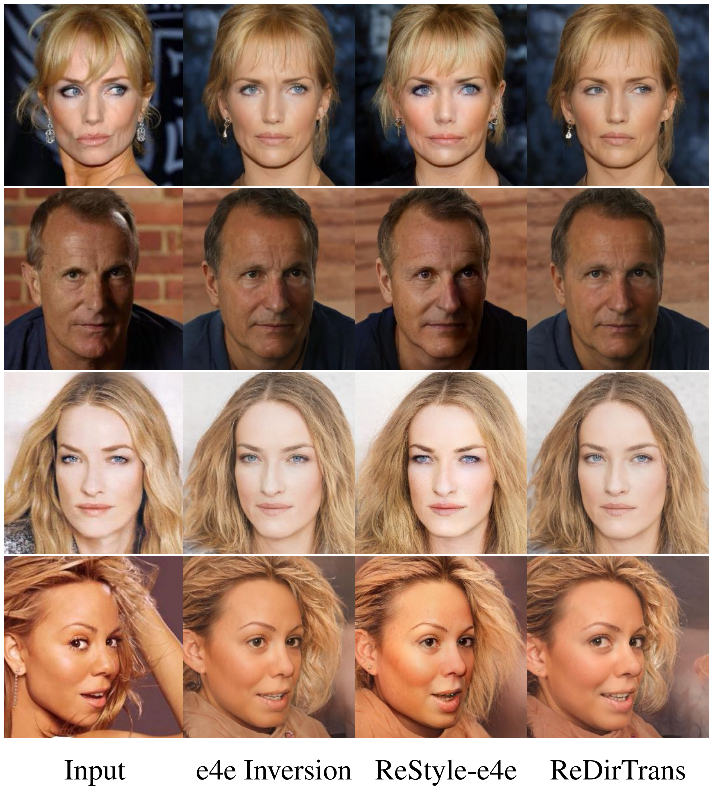


Figure 3.6: Gaze correction results of CelebA-HQ by viewing the same image as both the input and target.

gaze directions and head orientations. However, after we included ReDirTrans in the inversion pipeline as ReDirTrans-GAN, we can successfully maintain gaze directions and head orientations without affecting identity information (ID), which was measured by a pretrained ArcFace model [103]. Fig. 3.7 shows more examples. From both qualitative and quantitative evaluation, we can successfully correct the wrong gaze directions based on inverted results from e4e with ReDirTrans-GAN.

3.4.11 Redirection Accuracy of ReDirTrans-GAN

Table 3.10 presents the redirection accuracy of ReDirTrans-GAN in the GazeCapture test subset. We can observe that ReDirTrans-GAN cannot achieve as accurate redirection performance as ReDirTrans, which worked with the trainable encoder-decoder pair. There exists a **trade-off** between redirection accuracy and the following considerations:

Table 3.8: Within-dataset gaze correction performance given the input latent vectors encoded by e4e in the GazeCapture test subset. As for LPIPS and ID similarity, we compared the redirected image with the real target image (I_t) and its inverted image ($\hat{I}_t$), respectively. 'ReDir' denotes ReDirTrans-GAN

	Gaze Redir ↓	Head Redir ↓	LPIPS (I_t) ↓	ID (I_t) ↓	LPIPS ($\hat{I}_t$) ↓	ID ($\hat{I}_t$) ↓
e4e	11.302	4.13	0.334	0.377	–	–
ReDir	**2.505**	**1.020**	0.353	0.388	0.117	0.128

Table 3.9: Corss-dataset gaze correction performance given the input latent vectors encoded by e4e in CelebA-HQ. As for LPIPS and ID similarity, we compared the redirected image with the real target image (I_t) and its inverted image ($\hat{I}_t$), respectively. 'ReDir' denotes ReDirTrans-GAN.

	Gaze Redir ↓	Head Redir ↓	LPIPS (I_t) ↓	ID (I_t) ↓	LPIPS ($\hat{I}_t$) ↓	ID ($\hat{I}_t$) ↓
e4e	4.448	2.586	0.211	0.286	–	–
ReDir	**3.157**	**2.257**	0.228	0.314	0.087	0.099

- We utilized fixed encoder and generator parameters during the redirector training to ensure no modification to the predefined latent space;
- e4e encoded latent vectors in W^+ have limitations to understanding gaze in Section 3.4.10. e4e was trained with the FFHQ dataset, which does not include samples with as varied gaze directions and head orientations as the samples in GazeCapture. Given some cases with large gaze directions or head orientations, e4e cannot invert them very well;
- We kept the redirected latent codes within the 'high editability space' proposed by e4e to allow for further editing with other face editing techniques, sacrificing some quality (redirection accuracy);
- Extended face covering ranges and down-sampling of high-resolution generated images could cause the performance drop.
- The deprojector learned that the redirected latent vectors after addition and subtraction operations would not deviate away from the original input latent vectors. Thus ReDirTrans-GAN cannot redirect some extreme cases as well as ReDirTrans did, especially for head orientations.

Table 3.10: Within-dataset gaze correction performance given the input latent vectors encoded by e4e in the GazeCapture test subset. As for LPIPS and ID similarity, we compared the redirected image with the real target image (I_t) and its inverted image ($\hat{I}_t$), respectively. 'ReDir' denotes ReDirTrans-GAN.

	Gaze Redir ↓	Head Redir ↓	LPIPS (I_t) ↓	ID (I_t) ↓	LPIPS ($\hat{I}_t$) ↓	ID ($\hat{I}_t$) ↓
ReDir	2.648	1.863	0.448	0.212	0.223	0.130

- Predefined face alignments (four eyes corners and two mouth corners) restricts both the encoder and generator's ability for extreme head pose synthesis.

In summary, we made a deliberate choice to use a fixed encoder-generator pair, preserve edited latent codes in W^+, and edit within the 'high editability space' to maintain compatibility with continuing facial attribute editing by other methods. ReDirTrans-GAN provides a solution to edit attributes in predefined feature spaces that have limited abilities to depict those attributes. It also addresses the face editing task of redirecting or correcting gaze from the latent code perspective. Fig. 3.8 and Fig. 3.9 show redirected samples with modifying gaze directions and head orientations separately.

3.4.12 Layer-wise Weights

Given the layer-wise representation of latent vectors in W^+ space, we proposed layer-wise weights loss to measure the contribution of each layer for the corresponding attribute redirection. We compared the redirected samples with and without considering the layer-wise weights loss, shown in Fig. 3.10. We observed that gaze directions and head orientations become entangled without this loss and the network tends to learn a specific combination of gaze directions and head orientations. However, when we utilized this loss with the estimated layer-wise weights to modify each layer's output residuals further. In that case, gaze directions and head orientations can be disentangled and redirected independently.

3.5 Conclusions

We introduce ReDirTrans, a novel architecture working in either learnable or predefined latent space for high-accuracy redirection of gaze directions and head orientations. ReDirTrans projects input latent vectors into aimed-attribute pseudo labels and embeddings for redirection in an interpretable manner. Both the original and redirected embeddings of aimed attributes are deprojected to the initial latent space for modifying the input latent vectors by subtraction and addition. This pipeline ensures no compression loss to other facial attributes, including appearance information, which essentially reduces effects on the distribution of input latent vectors in initial latent space. Thus we successfully implemented ReDirTrans-GAN in the predefined feature space working with fixed StyleGAN to achieve redirection in high-resolution full-face images, either by assigned values or estimated conditions from target images while maintaining other facial attributes. The redirected samples with assigned conditions can be utilized as data augmentation for further improving learning-based gaze estimation performance. In future work, instead of a pure 2D solution, 3D data can be included for further improvements.

Chapter 3, in full, is a reprint of the material as it appears in the publication of “ReDirTrans: Latent-to-Latent Translation for Gaze and Head Redirection”, Shiwei Jin, Zhen Wang, Lei Wang, Ning Bi, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2023. The dissertation author was the primary investigator and author of this paper.

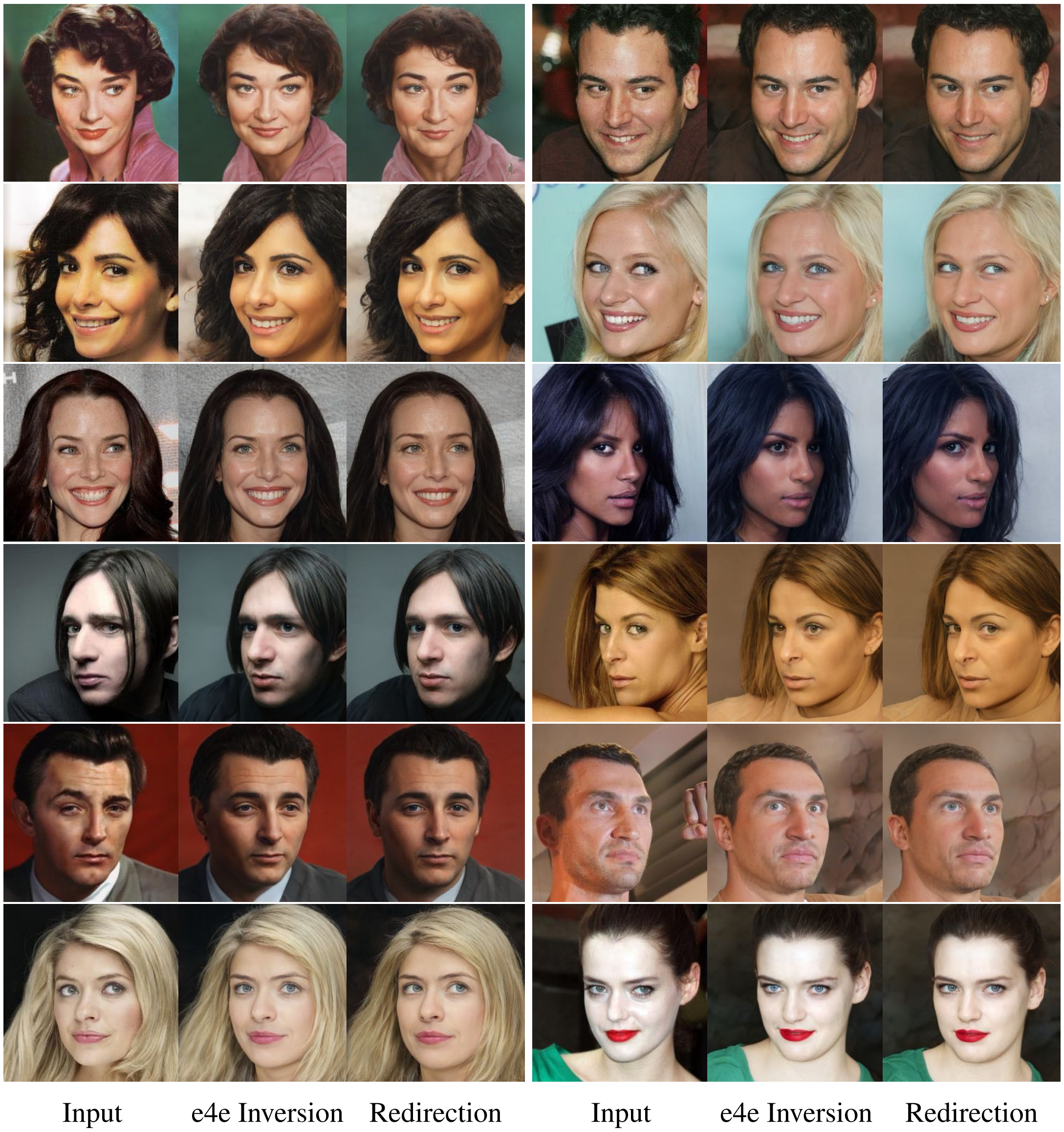


Figure 3.7: Gaze correction samples in CelebA-HQ. We set the same image as input and target to redirect the wrong gaze directions and head orientations given e4e inversion results.

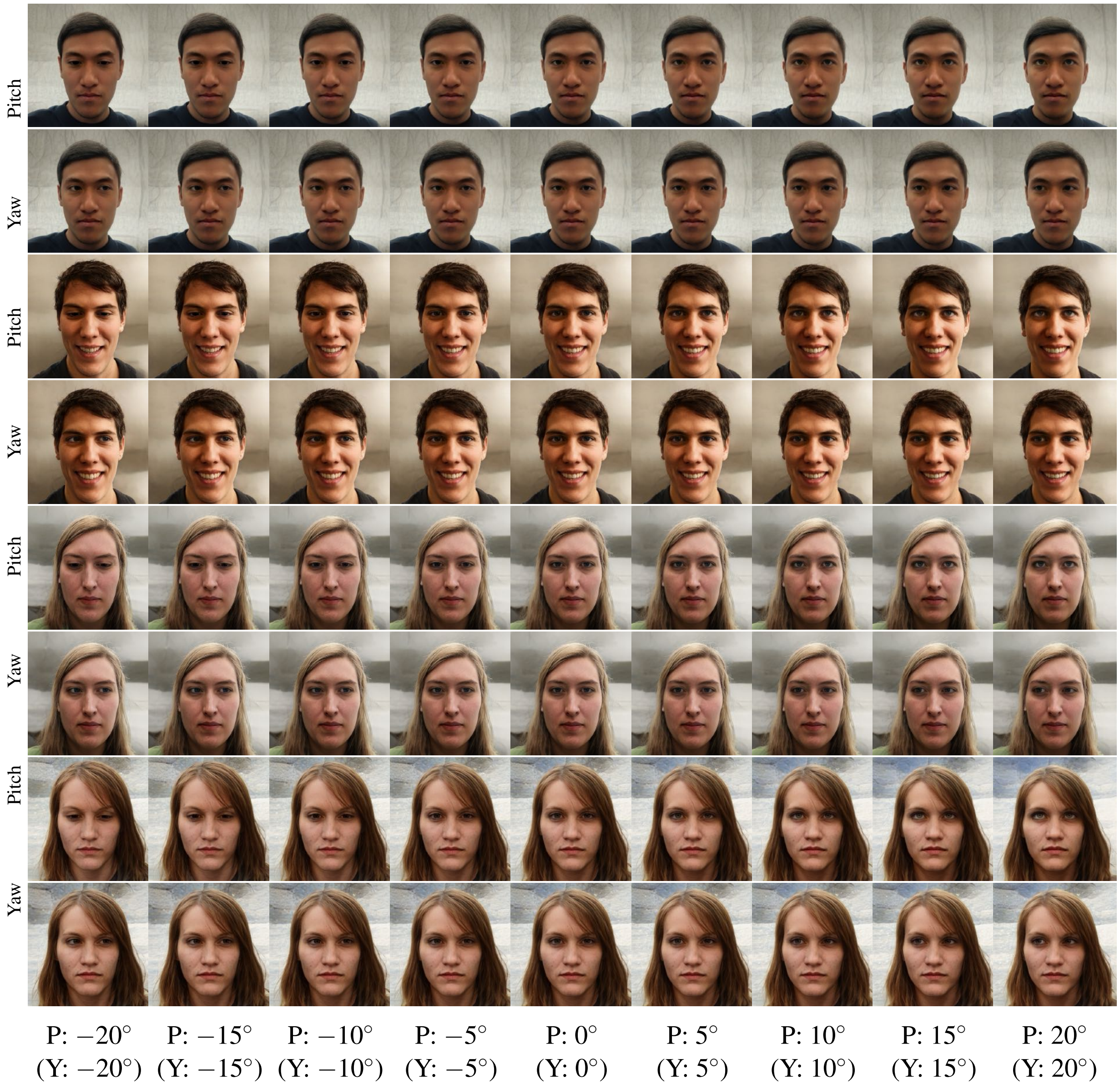


Figure 3.8: Gaze redirection results. The head orientations are all set as $(0^\circ, 0^\circ)$. 'Pitch' (P) means that we only redirect the pitch component of gaze directions and set yaw as 0°. 'Yaw' (Y) means that we only redirect the yaw component of gaze directions and set pitch as 0°. The redirected angles are listed at the bottom of the figure.

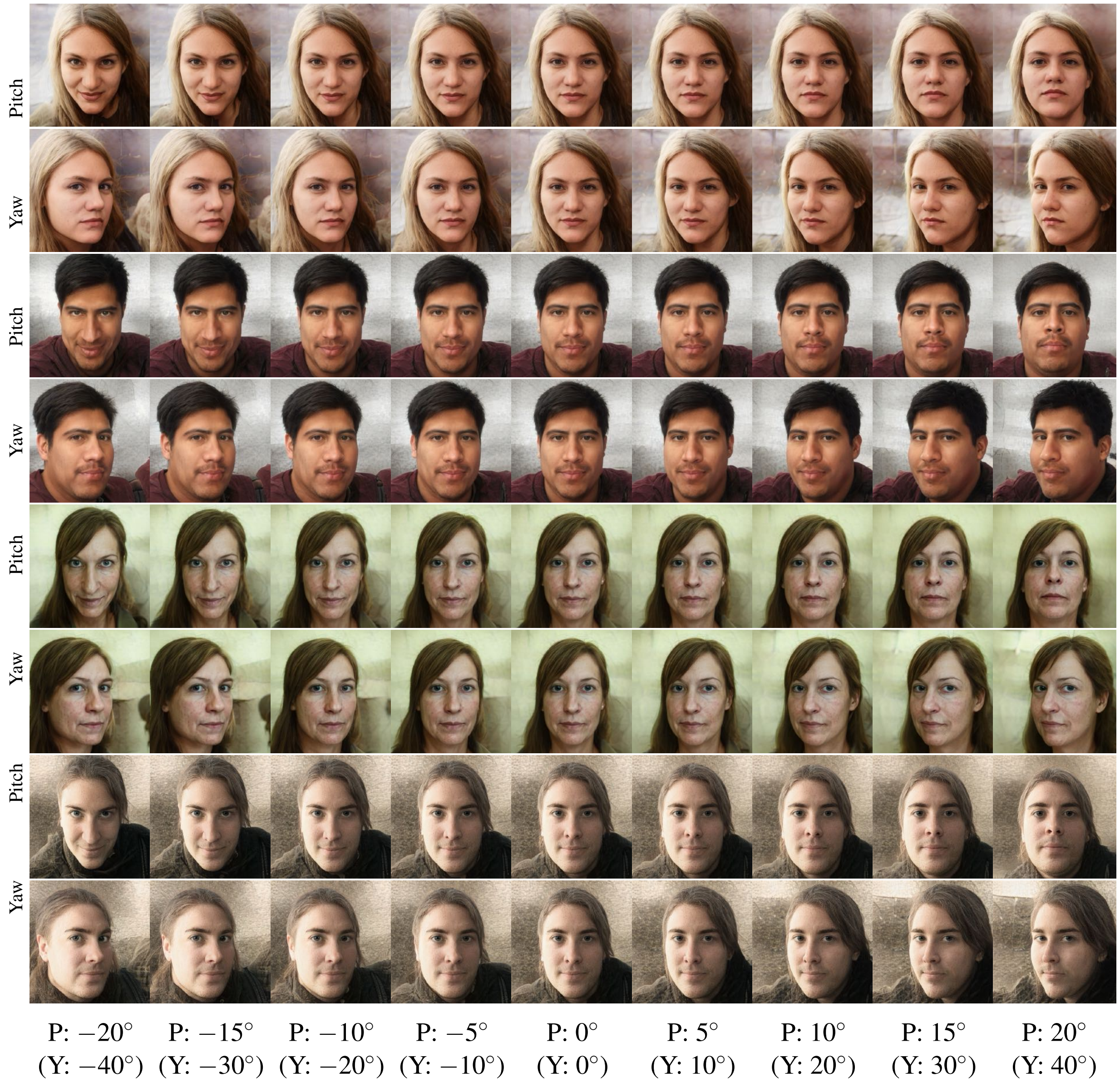


Figure 3.9: Head Redirection Results. The gaze directions are all set as $(0^\circ, 0^\circ)$. 'Pitch' (P) means that we only redirect the pitch component of head orientations and set yaw as 0°. 'Yaw' (Y) means that we only redirect the yaw component of head orientations and set pitch as 0°. The redirected angles are listed at the bottom of the figure.

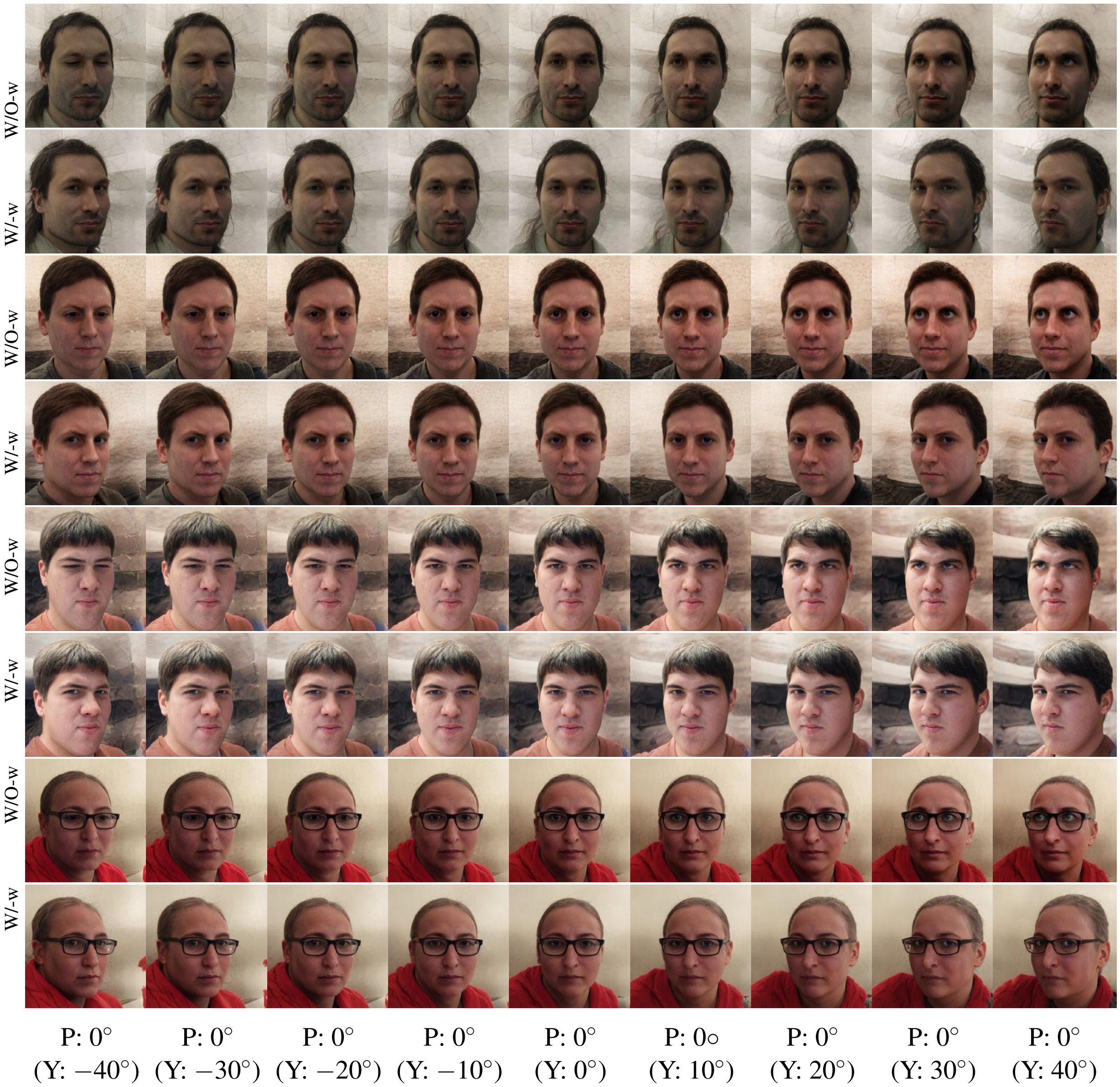


Figure 3.10: Comparison of redirected samples with or without using layer-wise weights loss in ReDirTrans-GAN. The gaze directions are all set as $(0^\circ, 0^\circ)$. We only redirect the head orientations given the provided pitch and yaw values below the figure. 'W/-w' denotes the redirected samples with the layer-wise weights loss. 'W/O-w' denotes the redirected samples without the layer-wise weights loss. 'P' denotes Pitch and 'Y' denotes Yaw.

Chapter 4

AUEditNet: Dual-Branch Facial Action Unit Intensity Manipulation with Implicit Disentanglement

4.1 Introduction

Facial action units (AUs), serving as anatomical indicators of facial muscle movements, have been effectively utilized as conditions for fine-grained facial expression editing in images [24, 26]. The manipulation of AU intensities offers advantages such as objective quantification on a six-integer-level ordinal scale defined by the Facial Action Coding System (FACS) [19], the ability to generate over 7000 combinations in observed facial expressions with a small number of AUs (30) [112], and the potentials for continuous intensity manipulation, instead of the category-based expression editing [113]. However, public datasets containing intensity annotations for over 10 AUs are constrained by limited subject counts, and frame-level AU intensity annotation requires expert involvement and extensive works. As a result, current AU intensity manipulation methods [24, 26, 27] often resort to the pretrained AU intensity estimator [20] to obtain predicted annotations for datasets with larger subject pools, sidestepping the reliance on expert-labeled datasets with a restricted number of subjects.

On the other hand, the semantic richness in latent space and high-quality generation capability of StyleGAN [29] have facilitated the development of facial attribute editing methods [35, 33, 101, 114, 115] that enable targeted modifications without affecting other attributes and identity. However, searching unified editing directions in the latent space for attribute editing typically requires substantial data from numerous subjects to disentangle the target attributes from others and identity. Limited number of subjects may lead to overfitting issues and poor generalization to new faces.

Considering these, it is challenging to search disentangled editing directions for manipulating intensities of multiple AUs based on the data from limited subjects. To address this, we propose a method to manipulate intensities of 12 AUs within the W^+ latent space [94] of StyleGAN [29] for high-resolution face image synthesis using only 18 subjects' data. Specifically, we introduce a novel pipeline designed to enforce disentanglement within the network, even with a dataset containing a limited number of subjects compared to the number of target facial attributes we aim to edit. This approach offers a potential solution to achieve desired facial attribute editing despite the dataset's limited subject count. To summarize, our contributions are as follows:

- Achieve accurate AU intensity manipulation in high-resolution synthesized face images conditioned by AU intensity values or target images without requiring network retraining or extra AU estimators.

- Introduce an architecture designed to disentangle target attributes from others and identity, even when working with data containing very few subjects compared to the number of target facial attributes we aim to edit.

- Propose the encoding of labels to match the level-wise disentangled structure of latent vectors in W^+ to avoid entangled labels as conditions for editing.

- Demonstrate the ability to manipulate float or negative AU intensities while generating consistent results, despite the training set labels encompassing six levels.

4.2 Related Work

4.2.1 AU Intensity Manipulation

GANimation [24] is an early work that utilizes AU intensities as conditions for facial expression manipulation. However, it suffers from attention mechanism issues that could result in overlap artifacts in regions where facial deformations occur [25]. Ling *et al.* [26] propose using the relative AU intensities between the source and target images as conditions, avoiding the direct addition of new attributes onto the existing expression [27]. Alternatively, ICface [27] introduces a two-stage editing pipeline. The initial stage transforms the input image into a neutral one with all AU intensities set to zero, and the second stage maps this neutral status to the final output, depicting the desired driving attributes with two independent generators. However, the architecture of ICface is redundant and resource-intensive. FACEGAN [116] utilizes AU representations to construct facial landmarks for expression transfer, reducing the potential of identity leakage from the target image. These methods place greater emphasis on facial expressions compared to AUs, both in terms of their editing goals and evaluation criteria.

4.2.2 Image Editing in Latent Space

The latent space working with StyleGAN2 [29] is well-known of its meaningful and highly disentangled properties. Several unsupervised methods [35, 117, 118, 36] search editing directions in the latent space without the need for attributes labels. For instance, GANSpace [35] employs principal component analysis to identify semantic editing directions in the latent space. In contrast, supervised methods [33, 101, 119, 114] typically rely on pretrained attribute estimators or attribute labels. InterFaceGAN [33], for example, utilizes a binary support vector machine [120] to estimate hyperplanes for the corresponding attribute editing. Furthermore, some methods [41, 121, 115] use the CLIP loss [40] to enable text-driven image manipulation. These methods usually handle identity information effortlessly since commonly used datasets contain a much larger number of subjects compared to the attributes involved in the editing process. However, in certain cases with a limited number of subjects included, the identity issue becomes significant. Therefore, in our work, we introduce a novel architecture designed to implicitly disentangle identity information from multiple attributes, even when dealing with a restricted number of subjects.

4.3 Proposed Method

4.3.1 Problem Setting

Our objective is to develop an intermediate module within the pretrained GAN inversion pipeline that enables the modification of specific facial attributes in input face images based on target conditions, while preserving the individual's identity and leaving other attributes unaffected. A crucial aspect of achieving this lies in effectively disentangling the target facial attributes from others and from identity. Prior works [35, 33] focused on identifying global editing directions in latent space for desired facial attributes by analyzing data from thousands of subjects. The data includes a significantly larger number of subjects than the number of facial attributes aiming to edit. Consequently, it is common for different subjects in the dataset to share the same facial attributes. This characteristic naturally facilitates the disentanglement of identity-related influences from the identified global editing directions for the corresponding attribute editing.

However, while data availability from numerous subjects is abundant, obtaining fine-grained labels poses challenges. The significant tradeoff between data collection and annotation efforts, particularly when expert annotation is necessary, can hinder the inclusion of detailed labels.

In specific face image editing tasks, such as AU intensity manipulation, multi-level intensity labels offer advantages over binary labels (activated or not). Yet, datasets with intensity labels covers more AUs often comprise fewer subjects. The limited subject pool in facial attribute editing may blend identity features with the target attributes, complicating disentanglement processes. To address this, we propose a novel framework named *AUEditNet*. This architecture enables seamless intensity adjustments across 12 AUs in face images, even when trained on a restricted dataset containing only 18 subjects.

4.3.2 Pipeline Overview

Consistent with previous works [101, 119, 115], we use a GAN inversion pair that consists of an encoder $\mathcal{E}$ and a generator $\mathcal{G}$ to achieve the transformation between the image space and the latent space. All editing occurs in the latent space. During training, we use a pair of images I_{src}, I_{tar} from one subject, while an additional face image I_{rnd} is randomly chosen from another subject's data. The processes are visually depicted in Fig. 4.1 with detailed descriptions.

Feature Space for Target AUs. Initially, we encode the input source image I_{src} into the latent vectors $W_{src} = \mathcal{E}(I_{src})$. To achieve explicit control over facial attributes using conditions associated with physical interpretation, we perform additional encoding of the latent vector W^j_{src}, one level from the multi-level vectors W_{src}, through a trainable encoder Φ^j_{enc}. Here, j corresponds to the level index, taking into account the disentangled level-wise structure of W_{src}, as outlined in Sec. 4.3.3. For the purpose of this subsection, we can disregard this index. The outcome of this encoding process is as follows:

$$\Phi^j_{enc}(W^j_{src}) = \left\{\hat{c}^{i,j}_{src}, \hat{a}^{i,j}_{src}, z^j_{src}\right\}, i \in [1,N], \tag{4.1}$$

where N represents the number of facial attributes included in the editing task. In this Eq. 4.1, $\hat{c}^{i,j}_{src}$ denotes whether the i-th facial attribute exists or not (AU is activated or not); $\hat{a}^{i,j}_{src}$ is

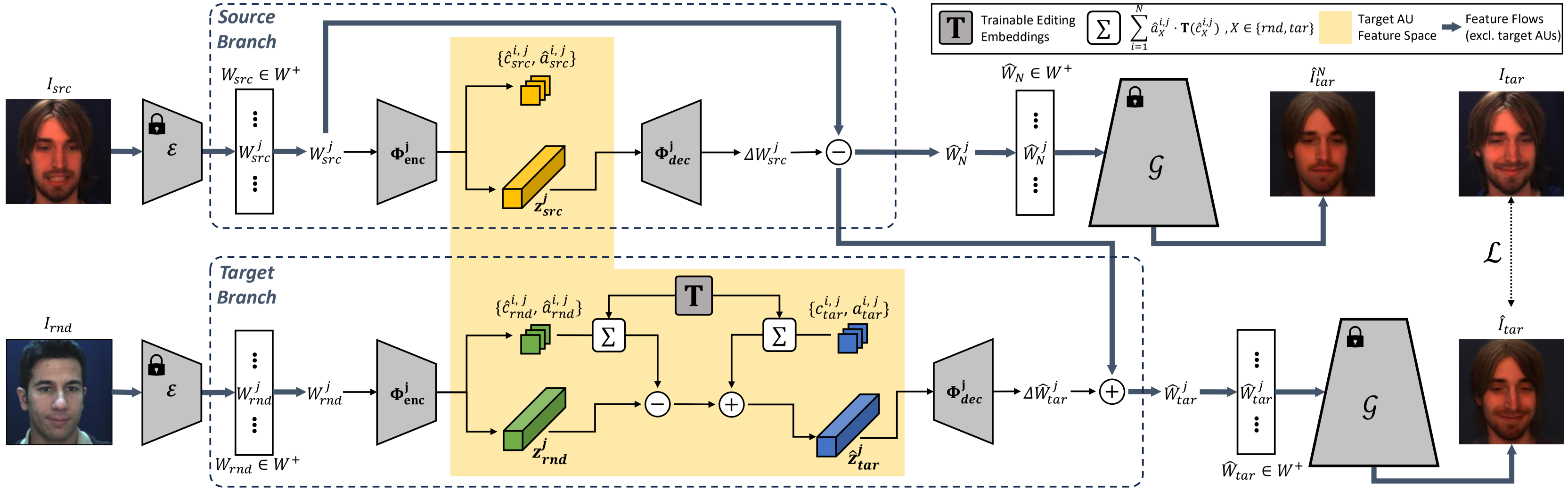


Figure 4.1: Overall scheme of the proposed AUEditNet. AUEditNet has a dual-branch structure that separately addresses source attribute removal (*Source Branch*) and target attribute addition (*Target Branch*). The *Source Branch* aims at removing the original status in I_{src}, maintaining other attributes and identity while keeping them distinct from the feature space of target facial attributes (highlighted in *yellow*). The *Target Branch* focuses on determining an edited direction $\Delta\hat{W}_{tar}^j$ for the new status of the target facial attribute, ensuring its independence from identity and other facial attributes. Instead of applying this branch directly to I_{src}, we randomly select another image I_{rnd}, facilitating implicit disentanglement of attributes and identity. The *blue bold arrows* present feature flows excluding the target facial attributes. In this configuration, AUEditNet guarantees that these flows remain outside the embedding space of the target facial attributes.

the corresponding estimated detailed labels (AU intensities); z_{src}^{j} is the embedding which acts as a medium for delivering information pertaining to the target facial attributes in this newly encoded space. $\hat{c}_{src}^{i,j}$ would select an editing direction from a globally trainable matrix $\mathbf{T}$ if the i-th facial attribute exists. $\mathbf{T}$ contains N editing directions, each possessing the same dimension as the embeddings. When a specific editing direction $\mathbf{T}(\hat{c}_{src}^{i,j})$ is chosen, we scale it with the estimated labels $\hat{a}_{src}^{i,j}$ to serve as an intensity control. This yields a normalized embedding $z_{N}^{j} = z_{src}^{j} - \sum_{i=1}^{N} \hat{a}_{src}^{i,j} \cdot \mathbf{T}(\hat{c}_{src}^{i,j})$. Ideally, z_{N}^{j} exclusively represents a canonical status of the target facial attribute, free from any person-specific information. While it seems feasible to continue incorporating new target conditions into this normalized embedding for subsequent generation with edited attributes [17, 119], this approach has limitations.

- It cannot ensure the complete exclusion of other attributes or identity features from the normalized embedding.

- Achieving optimal disentanglement of identity from target attributes requires training data that ideally encompasses as many subjects as possible to attain the desired normalized embedding.

- A loss function is necessary to enforce normalized embeddings identical within a batch, which heavily relies on the batch size and can be resource-intensive.

Given these limitations, instead of directly adding target conditions to the source embedding, our approach adopts a dual-branch structure to physically prevent irrelevant attribute or identity features (indicated by *blue bold arrows*) from infiltrating the feature space of target facial attributes (highlighted in *yellow*), as illustrated in Fig. 4.1.

We introduce I_{rnd} through the same processing steps with the shared-weights modules and build a normalized embedding instead of using the source one to compel the network to retain only the target-attribute related information within this encoded space during training. Finally, we introduce new conditions (the existence of the i-th attribute $c_{tar}^{i,j}$ and the corresponding detailed labels $a_{tar}^{i,j}$). This yields the edited embedding $\hat{z}_{tar}^{j} = z_{rnd}^{j} - \sum_{i=1}^{N} \hat{a}_{rnd}^{i,j} \cdot \mathbf{T}(\hat{c}_{rnd}^{i,j}) + \sum_{i=1}^{N} a_{tar}^{i,j} \cdot \mathbf{T}(c_{tar}^{i,j})$. During testing, we directly use source normalized embedding for efficiency considerations.

Source Latent Vectors Editing. For all other facial attributes and identity information, our goal is to preserve them within the original latent space [119]. We input the source embedding z_{src}^{j} and the edited target embedding $\hat{z}_{tar}^{j}$ into the decoder Φ_{dec}^{j} to obtain the residuals ΔW_{src}^{j} and $\Delta \hat{W}_{tar}^{j}$ respectively, which are used for editing W_{src}^{j}. The purpose of ΔW_{src}^{j} is to capture the source status of the target facial attributes in the input image, while $\Delta \hat{W}_{tar}^{j}$ stores the new status. Rather than solely assessing the result with the new status, we propose to supervise both outcomes through the following expressions:

$$\begin{cases} \hat{W}_{N}^{j} = W_{src}^{j} - \Delta W_{src}^{j}, \\ \hat{W}_{tar}^{j} = \hat{W}_{N}^{j} + \Delta \hat{W}_{tar}^{j}, \end{cases} \tag{4.2}$$

where $\hat{W}_{N}^{j}$ represents the intermediate editing resulting from the removal of the source status of the aimed facial attributes, and $\hat{W}_{tar}^{j}$ is the outcome achieved by incorporating the target conditions based on $\hat{W}_{N}^{j}$. After replacing the latent vector at the index j in W_{src} with $\hat{W}_{N}^{j}$ (or $\hat{W}_{tar}^{j}$), we obtain the final edited latent vectors $\hat{W}_{N}$ (or $\hat{W}_{tar}$) for image generation. $\hat{I}_{tar}^{N} = \mathcal{G}(\hat{W}_{N})$ represents a synthesized face image with zero intensities (deactivation) for all AUs, while $\hat{I}_{tar} = \mathcal{G}(\hat{W}_{tar})$ is generated based on the target intensities.

4.3.3 Multi-Level Architecture

The latent space used for editing is the W^{+} space [94], compatible with StyleGAN [29]. Latent vectors in W^{+} exhibit a multi-level structure, allowing them to control different semantic levels of images [122]. The level is indexed by j and $j \in [1, M]$, where $M \leq 18$ due to the dimension of W^{+}. Rather than reintegrating disentangled level-wise features in W^{+} using a single editing module, we opt for multiple independent editing modules $\{\mathcal{P}^{j}(\cdot) \mid j \in [1, M]\}$, each responsible for editing a specific level of the latent vectors, shown in Fig. 4.2. Here, M

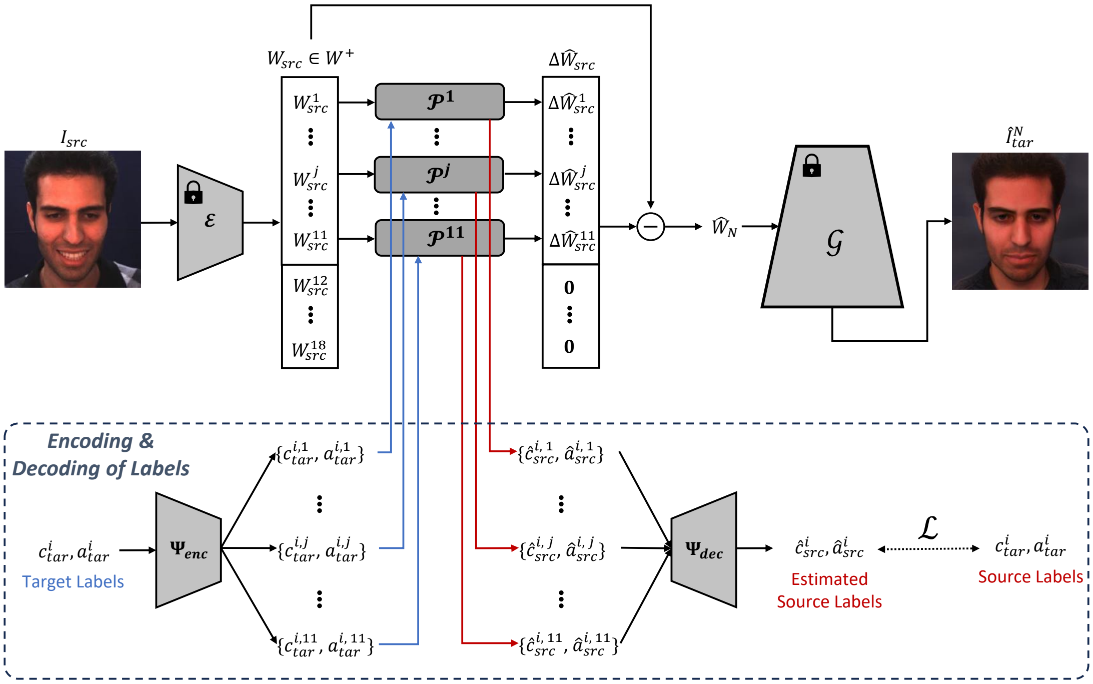

I_{src}
$\mathcal{E}$
$W_{src} \in W^+$
W^1_{src}
W^j_{src}
W^{11}_{src}
W^{12}_{src}
W^{18}_{src}
$\mathcal{P}^1$
$\mathcal{P}^j$
$\mathcal{P}^{11}$
$\Delta\widehat{W}_{src}$
$\Delta\widehat{W}^1_{src}$
$\Delta\widehat{W}^j_{src}$
$\Delta\widehat{W}^{11}_{src}$
0
0
$\widehat{W}_N$
$\mathcal{G}$
$\hat{I}^N_{tar}$
Encoding & Decoding of Labels
c^i_{tar}, a^i_{tar}
Target Labels
Ψ_{enc}
$\{c^{i,1}_{tar}, a^{i,1}_{tar}\}$
$\{c^{i,j}_{tar}, a^{i,j}_{tar}\}$
$\{c^{i,11}_{tar}, a^{i,11}_{tar}\}$
$\{\hat{c}^{i,1}_{src}, \hat{a}^{i,1}_{src}\}$
$\{\hat{c}^{i,j}_{src}, \hat{a}^{i,j}_{src}\}$
$\{\hat{c}^{i,11}_{src}, \hat{a}^{i,11}_{src}\}$
Ψ_{dec}
$\hat{c}^i_{src}, \hat{a}^i_{src}$
Estimated Source Labels
$\mathcal{L}$
c^i_{tar}, a^i_{tar}
Source Labels

Figure 4.2: Multi-level architecture of AUEditNet. We only focus on editing the first 11 levels of latent vectors in W^+. Each level has one corresponding editing module $\mathcal{P}^j$, whose detailed structure is described in Fig. 4.1. Given a sequence of target labels for 12 AUs, we first use Ψ_{enc} to encode them into embeddings and feed these embeddings into each $\mathcal{P}^j$ for editing purposes. Meanwhile, each $\mathcal{P}^j$ estimates the label embeddings from the source latent vectors. Subsequently, we use Ψ_{dec} to decode these estimated embeddings back to the label space and compare them with the actual source labels for supervision. For simplicity, we only include one target attribute with the index i. In the real implementation, the input target labels should include labels for all 12 AUs. We only include the *source branch* in this figure for better description. The pipeline is the same and the weights are shared in the *target branch*.

denotes the number of levels we aim to edit, set to 11 in our task. The rest of latent vectors maintain invariant during editing.

4.3.4 Encoding and Decoding of Labels

Various works explored incorporating input conditions into multi-level latent vectors within the W^+ space for editing purposes. StyleFlow [34] empirically found optimal level index ranges linked to specific facial attributes, like expression $(4-5)$, yaw $(0-3)$, and gender $(0-7)$. However, their focus was primarily on smiling expressions, which didn't satisfy our requirements for editing multiple AUs. Moreover, searching such optimal index ranges demands substantial datasets. ReDirTrans [119] proposed to apply the same conditions universally across levels and use error-based weights to determine each level's contribution to the target facial attribute. However, they assumed that their aimed attribute (gaze directions) could be estimated from a single level of the latent vectors in W^+, which might not suit other attribute manipulations.

Given these limitations, instead of focusing on which level (or levels) controls the target attribute, we propose encoding labels to align with the multi-level structure. This approach avoids mixing multiple facial attribute labels when inputted into individual levels. Specifically, we propose to first encode the target labels of multiple facial attributes ($\{\, (c^i_{tar}, a^i_{tar}) \mid 1 \in [1,N] \,\}$) into multi-level embeddings for fitting the multi-level structure. Then, we feed the j-th level embedding ($\{\, (\hat{c}^{i,j}_{tar}, \hat{a}^{i,j}_{tar}) \mid 1 \in [1,N] \,\}$) into the corresponding editing module $\mathcal{P}^j$ to perform editing. Given the level-wise estimated label embeddings from the source image, we decode them back to the original label space to get the estimated source labels $\{\, (\hat{c}^i_{src}, \hat{a}^i_{src}) \mid 1 \in [1,N] \,\}$. The entire process can be summarized as follows:

$$\begin{cases} \Psi_{enc}(c^i_{tar}, a^i_{tar}) = c^{i,j}_{tar}, a^{i,j}_{tar}, \\ \Psi_{dec}(\hat{c}^{i,j}_{src}, \hat{a}^{i,j}_{src}) = \hat{c}^i_{src}, \hat{a}^i_{src}, \end{cases} \tag{4.3}$$

where $i \in [1,N]$, $j \in [1,M]$, and the subscripts '*src*' and '*tar*'can be interchanged if we switch the roles of the source and target images during training. The proposed encoding-decoding pipeline for labels doesn't restrict the estimation of aimed attributes to a single level of latent vectors. Fig. 4.2 presents the overall multi-level architecture of AUEditNet. The encoder-decoder pair, ψ_{enc} and ψ_{dec} are trained based on the **Label Loss** introduced in Sec. 4.3.6.

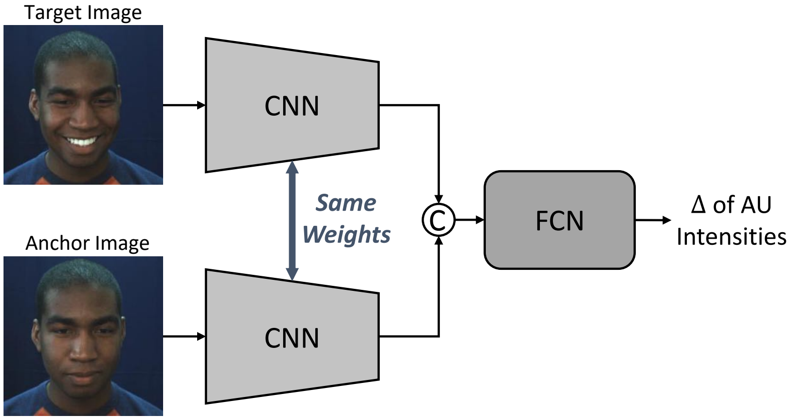


Figure 4.3: Structure of the AU intensity estimator. This Siamese network takes a pair of images from the same subject as inputs and estimates the difference of AU intensities between these two images (the target and anchor images). We use convolutional neural network (CNN) to extract features. After concatenating two features, we use fully-connected network (FCN) to regress the final output.

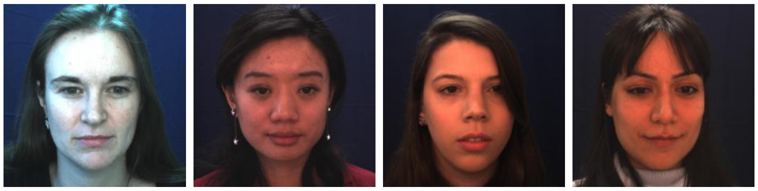


Figure 4.4: Comparison of eyebrow positions and shapes on the DISFA dataset. All of these four images have deactivated (zero-intensity) AU 1 (Inner Brow Raiser), AU 2 (Outer Brow Raiser) and AU 4 (Brow Lowerer). We can observe that the different eyebrow positions and shapes could affect the performance given a unified AU intensity estimator.

4.3.5 AU Intensity Estimator

In our work, pretrained AU intensity estimators are required at two stages: when utilizing the *Pretrained Function Loss* in Sec. 4.3.6 during training and when evaluating manipulation performance quantitatively during inference.

Estimator Structure. We utilize a Siamese network for AU intensity estimation, shown in Fig. 4.3. The input is a pair of images from the same subject. One is viewed as the target image, and the other one is viewed as the anchor image. The output is the difference of AU intensities between the target image and the anchor image. This design could help to reduce personal facial attributes' influences, such as eyebrow positions and shapes affecting the eyebrow-related AU movements, illustrated in Fig. 4.4. If all AU intensities in the anchor image are at zero, the output represents the absolute intensities of AUs in the target image.

Estimator in Training. During training, the pretrained convolutional part of VGG-16 [123] serves as the backbone in the AU intensity estimator, trained on the DISFA training subset. It functions as F_{pre} to detect AU intensities in synthesized images during AUEditNet's training. The anchor image is randomly chosen from the same subject's data with all AUs deactivated (zero intensity).

Estimator in Inference. During testing, we use another external AU intensity estimator to quantify the manipulation performance, which is unseen during training. We use the pretrained convolutional part of ResNet-50 [110] as the backbone to build the AU intensity estimator H_{est}, trained on the DISFA training subset.

4.3.6 Objectives

During training, AUEditNet requires source and target images from the same subject. And the random image can be randomly picked from other subjects. We train AUEditNet by minimizing the following loss:

$$\mathcal{L} = \lambda_R \mathcal{L}_R + \lambda_P \mathcal{L}_P + \lambda_F \mathcal{L}_F + \lambda_{ID} \mathcal{L}_{ID} + \lambda_L \mathcal{L}_L. \tag{4.4}$$

Pixel-wise and Perceptual Losses. We minimize both the pixel-wise loss $\mathcal{L}_R$ and the perceptual loss $\mathcal{L}_P$ [124] between the edited image $\hat{I}_{tar}$, which is generated based on the provided target conditions, and the actual target image I_{tar}.

$$\begin{aligned} \mathcal{L}_R &= \|\hat{I}_{tar} - I_{tar}\|_2, \\ \mathcal{L}_P &= \|F_{pcept}(\hat{I}_{tar}) - F_{pcept}(I_{tar})\|_2, \end{aligned} \tag{4.5}$$

where $F_{pcept}(\cdot)$ denotes the perceptual feature extractor.

Pretrained Function Loss. Following the prior works [17, 119], the pretrained function loss $\mathcal{L}_F$ focuses on task-relevant inconsistencies between $\hat{I}_{tar}$ and I_{tar}. The inconsistencies include both intermediate activation feature maps $\{f_k, k \in [1, K]\}$ and estimation results derived from a network $F_{pre}(\cdot)$, which is pretrained on the specific task (e.g. AU intensity estimation).

$$\begin{aligned} \mathcal{L}_F =& \frac{1}{K} \sum_{k=1}^{K} \|f_k(\hat{I}_{tar}) - f_k(I_{tar})\|_2 \\ &+ \frac{1}{N} \|F_{pre}(\hat{I}_{tar}) - F_{pre}(I_{tar})\|_2, \end{aligned} \tag{4.6}$$

where K is the number of chosen layers from F_{pre}.

Identity Loss. We restrict the ID similarity between the real image I_{tar} and two generated images $\hat{I}_{tar}; \hat{I}^N_{tar}$ based on the pretrained ArcFace network [103], denoted as F_{id}.

$$\mathcal{L}_{ID} = \sum^{\hat{I} \in \{\hat{I}_{tar}, \hat{I}^N_{tar}\}} 1 - \langle F_{id}(\hat{I}), F_{id}(I_{tar}) \rangle \tag{4.7}$$

Label Loss. We propose the label loss to supervise the transformation process in the 'latent space' for labels through Ψ_{enc} and Ψ_{dec} as mentioned in Sec. 4.3.3. Let's assume we use the source image's labels (c^i_{src} and a^i_{src}) as the target conditions for the i-th facial attribute. In this scenario, the generated image should be identical to the source image. This implies that the estimated source embedding z^j_{src} should match the edited embedding $\hat{z}^j_{tar}$ at each level. In other words, the removal of the source status and the addition of the target status should be entirely consistent. As a result, the level-wise conditions $(c^{i,j}_{src}, a^{i,j}_{src})$ in the label's latent space, encoded from the source image's labels (c^i_{src}, a^i_{src}), should align with the estimated conditions $(\hat{c}^{i,j}_{src}, \hat{a}^{i,j}_{src})$. This corresponds to the second part of Eq. 4.8, which supervises the learning of the encoder Ψ_{enc}.

To further ensure that the level-wise estimation retains the information about the labels of the source image, we utilize the label decoder Ψ_{dec} to guarantee that the estimated results $(\hat{c}^i_{src}, \hat{a}^i_{src})$ decoded from the level-wise conditions $(\hat{c}^{i,j}_{src}, \hat{a}^{i,j}_{src})$ are consistent with the source image's labels. Thus, we build the loss as follows:

$$\mathcal{L}_L = \sum^{\xi \in \{c,a\}} \frac{1}{N} \sum_{i=1}^{N} \left(\|\hat{\xi}_{src}^{i} - \xi_{src}^{i}\|_2 + \frac{1}{M} \sum_{j=1}^{M} \|\hat{\xi}_{src}^{i,j} - \xi_{src}^{i,j}\|_2 \right). \tag{4.8}$$

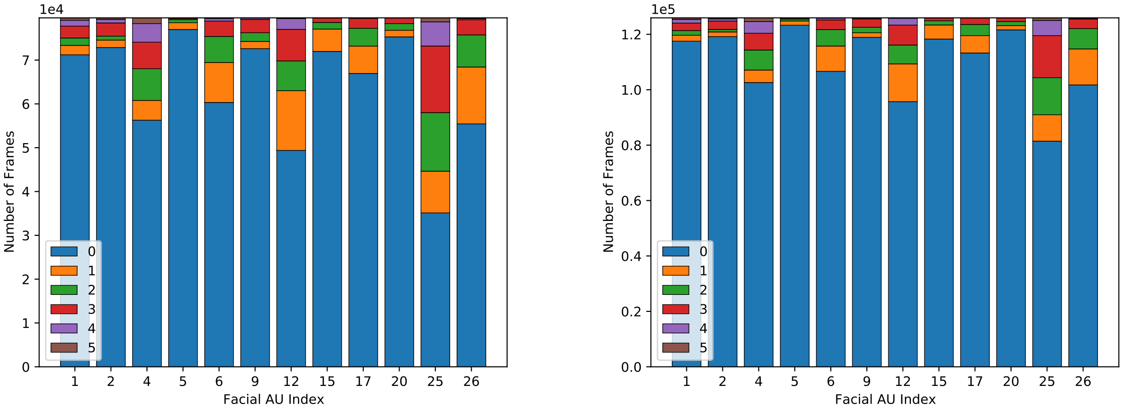
In summary, the first three terms in Eq. 4.4 ensure the generated image's similarity to the target image. The fourth term enforces identity consistency, and the final term supervises the learning of level-wise pseudo-labels for avoiding entangled labels as conditions and improving the attribute editing performance. The hyperparameters $\lambda_R, \lambda_P, \lambda_F, \lambda_{ID}, \lambda_L$ enable a balanced learning from these various losses.

4.4 Experiments

4.4.1 Implementation Details

We employed e4e [30] and StyleGAN2 [29] as the GAN inversion pair. We designed a Siamese network for the external AU intensity estimation for the pretrained function loss in Eq. 4.6. This network takes a pair of face images from the same subject as the input and estimates the intensity difference of AUs between these two images. This design reduces the impact of subject-specific facial attributes. During training, we used the convolutional part of VGG-16 [123] as the backbone to build the AU intensity estimation network F_{pre}. During test, we used a separate estimator H_{est}, which has the same architecture with ResNet-50 [110] as the backbone. Importantly, this estimator H_{est} was never exposed to the training phase but was trained on the same training dataset.

We trained AUEditNet using the DISFA training subset [125, 126]. DISFA comprises of 27 subjects and provides multi-level integral intensities for 12 AUs, offering annotations for the largest number of AUs among publicly available datasets for AU intensity estimation. The DISFA dataset [126, 125] is the only public dataset that contains intensity labels for 12 action



(a)

(b)

Figure 4.5: Distribution of AU intensities in DISFA. (a) Including samples with at least one non-zero AU intensity. (b) Whole DISFA dataset. As shown in (a), the distribution remains highly imbalanced after filtering out samples with zero intensities for all AUs.

units (AUs). It serves as the benchmark for AU intensity estimation tasks [127, 128]. Current AU intensity manipulation methods [24, 26, 27] often rely on large public datasets with predicted AU intensities as ground truth. This preference arises due to DISFA's limitations: it comprises only 27 subjects, notably fewer than the extensive subject pools of 337, 98, and over 1000 subjects used in these methods [24, 26, 27], respectively. Additionally, the intensity distribution within DISFA is highly imbalanced, as depicted in Fig. 4.5. Nevertheless, to the best of our knowledge, we are the first work to leverage such imbalanced datasets with limited subject counts for achieving AU intensity manipulation.

To assess AUEditNet, we used the DISFA test subset to evaluate its accuracy in manipulating AU intensities while preserving other attributes. We used 18 subjects for training and 9 subjects for testing, following the data split used in [129, 130]. Furthermore, we expanded our evaluation to encompass facial expressions, beyond AUs alone, by using the BU-4DFE dataset [131]. Our evaluation involved tasks related to expression transfer and data augmentation for AU intensity estimation. For further assessment of out-of-domain editing performance, we incorporated CelebA-HQ [104] and FFHQ [93], which both are the benchmarks for the high-quality human face image datasets.

4.5 Training Details

To expedite training and mitigate the influence of numerous samples with zero intensities of all AUs, shown in Fig. 4.5, we always use one sample with at least one non-zero AU intensity as the source image. The target and random images are chosen randomly from the rest data without any special requirements. We utilize the cycle pipeline [132] to input the generated target image with source image conditions back to the network to achieve cycled image reconstruction.

We opt for a batch size of 2, utilizing Adam optimizer [78] with default momentum values ($\beta_1 = 0.9, \beta_2 = 0.999$). The training process, consuming around $18,123$ MiB on a single NVIDIA RTX 3090, iterates for $30,000$ iterations. The loss weights in Eq. 4.4 are set as $\lambda_R = 8, \lambda_P = 1, \lambda_F = 125, \lambda_{ID} = 20, \lambda_L = 20$.

4.5.1 Evaluation Criteria

We assess the performance of AUEditNet by examining the comparison between the generated image $\hat{I}_{tar}$ and the target image I_{tar} from four perspectives: the accuracy of intensity editing in AUs, identity preservation, image similarity, and smile expression manipulation (illustrated in Sec. 4.5.3).

Accuracy of AU Intensity Manipulation. We quantify the AU intensity manipulation performance in edited images by using the external pretrained ResNet-50 based estimator H_{est}, which is unseen during training. We report the Intra-Class Correlation (specifically ICC(3,1) [133]) and mean squared error (MSE), both calculated for 12 AUs, between the estimated values $H_{est}(\hat{I}_{tar})$ or $H_{est}(\hat{I}^N_{tar})$ and their intended target values.

Identity Preservation. A well-trained image editor should consistently maintain the identity given various provided conditions. To assess the similarity of identity, we measure the distance of embeddings between $\hat{I}_{tar}$ and I_{tar} to assess the similarity of identity, where the embedding is extracted by a pretrained face recognition model [134].

Image Similarity. We employ two metrics: pixel-wise mean squared error and the Learned Perceptual Image Patch Similarity (LPIPS) [102] to measure the image similarity between $\hat{I}_{tar}$ and I_{tar}.

4.5.2 Qualitative Evaluation

Within-Dataset Evaluation Fig. 4.6 illustrates a qualitative comparison of AU intensity manipulation based on provided target conditions. Both ReDirTrans [119] and our proposed AUEditNet employ a two-step editing process to prevent potential attribute status mixing. After the source status removal, the generated images should exhibit all AU intensities set to zero, serving as benchmarks when all AUs are deactivated. ReDirTrans and AUEditNet demonstrate the ability to learn the desired AU movements, under both cases when deactivating all AUs or assigning new target intensities. However, ReDirTrans fails to preserve identity information in intermediate and final generated images. Additionally, ReDirTrans attempts to address color discrepancy between real and inverted images during AU editing, resulting in undesired color distortion in images. In contrast, AUEditNet focuses only on editing the aimed AUs' intensities, devoid of unrelated information, which is achieved through the dual-branch architecture. On the other hand, DeltaEdit [115] excels in maintaining identity information and other facial attributes. However, it is limited to learning noticeable AU movements and may ignore subtle motions such as eyebrow, cheek, and lip corner movements, potentially causing significant changes in the entire facial expression. AUEditNet successfully achieves accurate AU intensity editing under this two-phase editing process while maintaining identity.

Cross-Dataset Evaluation Fig. 4.7 presents the cross-dataset results involving single AU editing with multiple intensity levels on the CelebA-HQ dataset [104]. AUEditNet exhibits

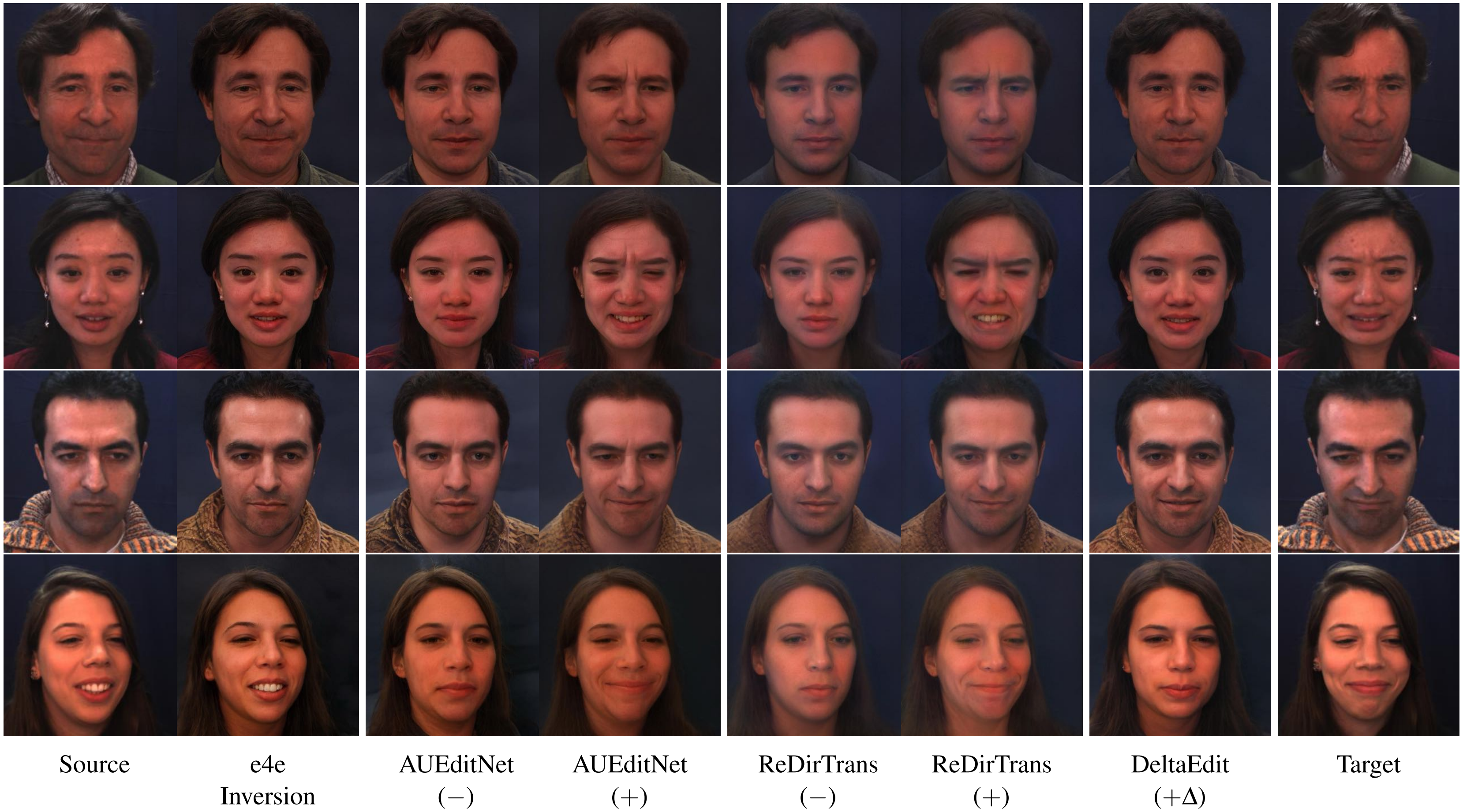


Figure 4.6: Comparison of AU intensity manipulation using target AU intensities in DISFA. AUEditNet, ReDirTrans generate editing results that involve the removal (−) of source attributes and the addition (+) of target attributes. DeltaEdit uses intensity differences between source and target images for attribute addition (+Δ). The removal (−) process yields 'neutral-like' face images with all AU intensities set to zero.

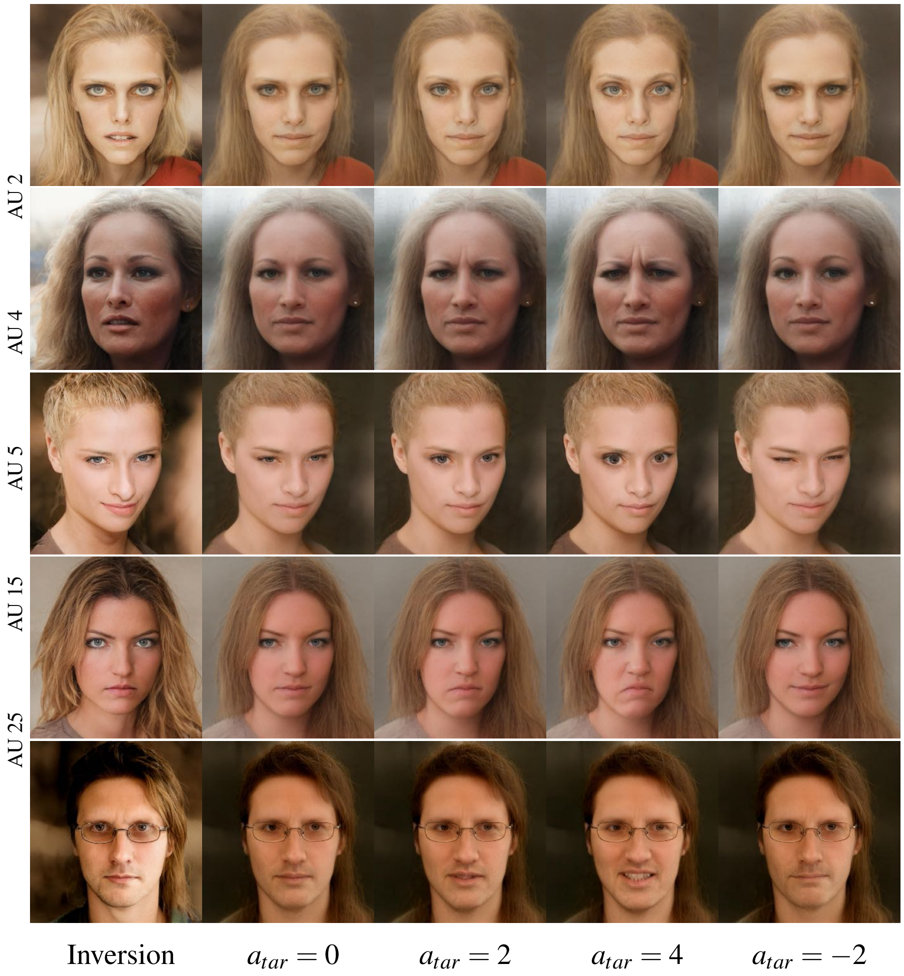


Figure 4.7: Cross-dataset evaluation of single AU intensity manipulation in CelebA-HQ. The descriptions of AUs (from top to bottom) are Outer Brow Raiser, Brow Lowerer, Upper Lid Raiser, Lip Corner Depressor, and Lips Part. a_{tar} represents the target intensity.

the capability to achieve consecutive AU intensity manipulation. Notably, even in the absence of negative intensities during training, AUEditNet produces reasonable editing outcomes. For instance, applying negative intensity to AU 5 (Upper Lid Raiser) results in a generated image with partial eye closure. Regarding AU 25 (lips part), where intensity indicates mouth openness, providing negative intensity still maintains the closed configuration, aligning with the case of zero intensity instead of creating unrealistic results.

4.5.3 Quantitative Evaluation

Accuracy of AU Intensity Editing. Table 4.1 presents measurements of ICC and MSE for comparing the estimated AU intensities against ground truth. We categorize the methods under each evaluation metric based on their research directions, whether they focus on the estimation or editing of AU intensities in images. Among the editing methods, our proposed AUEditNet surpasses state-of-the-art facial attribute editing methods, especially in terms of the average performance across all 12 AUs. When it comes to the performance of deactivating all AUs, AUEditNet achieves a substantial 38.92% improvement in MSE compared to ReDirTrans [119]. This illustrates the complete and accurate attribute removal process, which, in turn, contributes to enhanced final performance since attribute removal and addition are entirely reversible processes with shared trainable parameters.

Furthermore, we expand our comparison to include both editing and estimation methods because the editing performance is also assessed using the same AU intensity estimation process. Moreover, the external estimator H_{est} is trained on the same data as AU intensity estimation methods. We still observe that the estimation performance, when evaluated with our edited face images, surpasses that of state-of-the-art AU intensity estimation methods on the DISFA test subset [125, 126]. This finding further solidifies the high level of consistency between the provided target intensities and the edited images generated by AUEditNet.

Identity Preservation and Image Similarity. Table 4.2 summarizes the performance of identity preservation and image similarity given image editing results. In addition to comparing the edited images with the real target images, we also conduct a comprehensive comparison using GAN-inverted images as the target. All three editing methods focus on the latent code editing, without adjusting the image encoder and generator. From the identity perspective, DeltaEdit [115] achieves the best performance, nearly matching the GAN inversion performance. However, this is at the cost of AU intensity manipulation accuracy, resulting in a decline of 71.50% in ICC and 98.23% in MSE compared to AUEditNet. Comparing our AUEditNet with ReDirTrans [119],

Table 4.1: Comparison to the state-of-the-art action unit (AU) intensity estimation and editing methods in DISFA [125]. The ‘Method’ column under each metric is categorized into two parts: 1. Upper part: AU intensity estimation methods; 2. Lower part: AU intensity editing methods. In the estimation task, we evaluate the performance by comparing the estimated intensities of the input image to the ground truth. For the editing task, the procedure begins with the editing of the input image based on the target conditions. Then we acquire the estimated AU intensities from the edited image via the external pretrained estimator H_{est}. Finally, we compare these estimated intensities with the provided target conditions. ‘(*N*)’ denotes the results obtained after the source attribute removal, where all AU intensities are set to zero. Because the desired output under this case should exhibit no AU movements, resulting in a single, uniform outcome, these results are only compared within this group. The best performance is indicated within brackets and in bold, while the second best is highlighted only in bold.

	Method	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	Avg
ICC(3, 1) (↑)	2DC [135]	.70	.55	.69	.05	**.59**	**.57**	**.88**	.32	.10	.08	.90	.50	.50
	HR [127]	.56	.52	.75	.42	.51	.55	.82	**[.55]**	.37	.21	**.93**	**.62**	.57
	Aps [136]	.35	.19	.78	**[.73]**	.52	**[.65]**	.81	**.49**	**.61**	.28	.92	**[.67]**	.58
	DeltaEdit [115]	.091	.058	.114	.034	.383	.065	.694	.008	.004	.041	.581	.166	.179
	ReDirTrans [119]	**[.856]**	**[.631]**	**.851**	.436	**[.634]**	.278	.862	.364	.602	**.481**	.927	.480	**.617**
	AUEditNet	**.848**	**.559**	**[.874]**	**.600**	.577	.230	**[.890]**	.276	**[.669]**	**[.511]**	**[.950]**	.548	**[.628]**
MSE (↓)	2DC [135]	.32	**.39**	.53	.26	**[.43]**	**.30**	**.25**	.27	.61	**.18**	.37	.55	.37
	HR [127]	.41	**[.37]**	.70	.08	**.44**	**.30**	.29	.14	.26	**[.16]**	**.24**	.39	.32
	Aps [136]	.68	.59	.40	**.03**	.49	**[.15]**	.26	.13	**.22**	.20	.35	**[.17]**	.30
	DeltaEdit [115]	.605	.686	1.311	.031	.513	.485	.570	**.080**	.424	.454	1.157	.420	.561
	ReDirTrans [119]	**[.181]**	.397	**.341**	.034	.453	.552	.286	**[.070]**	.225	.333	.247	**.367**	**.290**
	AUEditNet	**.191**	.445	**[.309]**	**[.029]**	.492	.579	**[.228]**	**.080**	**[.188]**	.322	**[.169]**	**.367**	**[.283]**
	ReDirTrans (*N*)	**[.045]**	.117	**[.025]**	**[.019]**	**[.024]**	.009	.300	.032	.177	**[.032]**	.803	.427	.167
	AUEditNet (*N*)	.069	**[.101]**	.098	.024	.036	**[.006]**	**[.227]**	**[.004]**	**[.014]**	.063	**[.351]**	**[.228]**	**[.102]**

Table 4.2: Comparison of identity preservation and image similarity in facial attribute editing methods. 'GAN Inversion' as a baseline illustrates that the accuracy of action unit intensity editing cannot be reflected in the performance of the Image Similarity criteria. The best performance is indicated within brackets and in bold, while the second best is highlighted only in bold.

Method	Target Image	Identity Preservation	Image Similarity	
		Distance (↓)	L2 (↓)	LPIPS (↓)
GAN Inversion [30]	Real	.368	.025	.173
	Inverted	.278	.011	.065
DeltaEdit [115]	Real	**[.396]**	**[.022]**	**[.165]**
	Inverted	**[.309]**	**[.011]**	**[.074]**
ReDirTrans [119]	Real	.505	**.024**	.175
	Inverted	.479	.018	.153
AUEditNet	Real	**.468**	.026	**.174**
	Inverted	**.435**	**.016**	**.126**

we observe the identity preservation improvements of 7.33% and 9.19% considering real and inverted images, respectively. These results further validate the effectiveness of our method's ability to achieve disentanglement and preserve identity during intensity manipulation.

Regarding image similarity, DeltaEdit [115] continues to outperform the other two editing methods. However, when using the GAN inversion as the baseline to compare the inverted source image with the real or inverted target images separately, we find that the image similarity criteria still maintain good performance, even when dealing with different AU intensities between source and target images. In other words, the difference in AU intensities is not reflected over the image similarity. When compared to ReDirTrans [119], AUEditNet achieves comparable performance with the real target image and achieves better performance with the inverted one. These results further demonstrate AUEditNet's disentanglement ability when achieving AU intensity editing.

Smile Manipulation. We evaluate smile attribute manipulation using metrics proposed in [114] on the FFHQ dataset [93]. Specifically, we modify the intensities of AU 6 (Cheek Raiser) and AU 12 (Lip Corner Puller) across eight levels simultaneously to enable smile intensity editing

Table 4.3: Comparison of smile intensity manipulation performance on the FFHQ test dataset. AUEditNet achieves the best performance given identity preservation (E_d) and manipulation efficiency (ρ).

Method	Smile Attribute	
	E_d (↓)	ρ (↑)
Talk-to-Edit [137]	0.212	40.9
StyleFlow [34]	**0.099**	88.9
Do *et al.* [114] (W/ StyleGAN2)	0.103	96.9
AUEditNet	**0.099**	**121.3**

[138]. Table 4.3 provides comparisons based on identity preservation (E_d) and manipulation efficiency (ρ). The result indicates that AUEditNet better preserves identity when an attribute undergoes the same quantity of change than others.

4.5.4 Expression Transfer

Setting individual AU values is a cumbersome process and requires expertise for achieving desired expression synthesis [139]. In contrast, our proposed AUEditNet demonstrates the capability to directly transfer facial expressions from target images without the need for retraining the network. The process involves inputting the target image with the desired expression into the target branch in Fig. 4.1. Instead of employing removal and addition processes, we directly feed the estimated embeddings of the target image into the decoder Φ^j_{dec}, similar to the procedure in the source branch, to acquire editing residuals with target facial expressions. Fig. 4.8 shows expression transfer results on the BU-4DFE dataset [131]. The edited images demonstrate the contributions of AU intensity manipulation to the facial expression reenactment.

4.6 Smile Attribute Manipulation

To further validate AUEditNet's effectiveness, we assess the facial expression editing performance by manipulating intensities over some AUs. We modify the intensities of AU 6 (Cheek

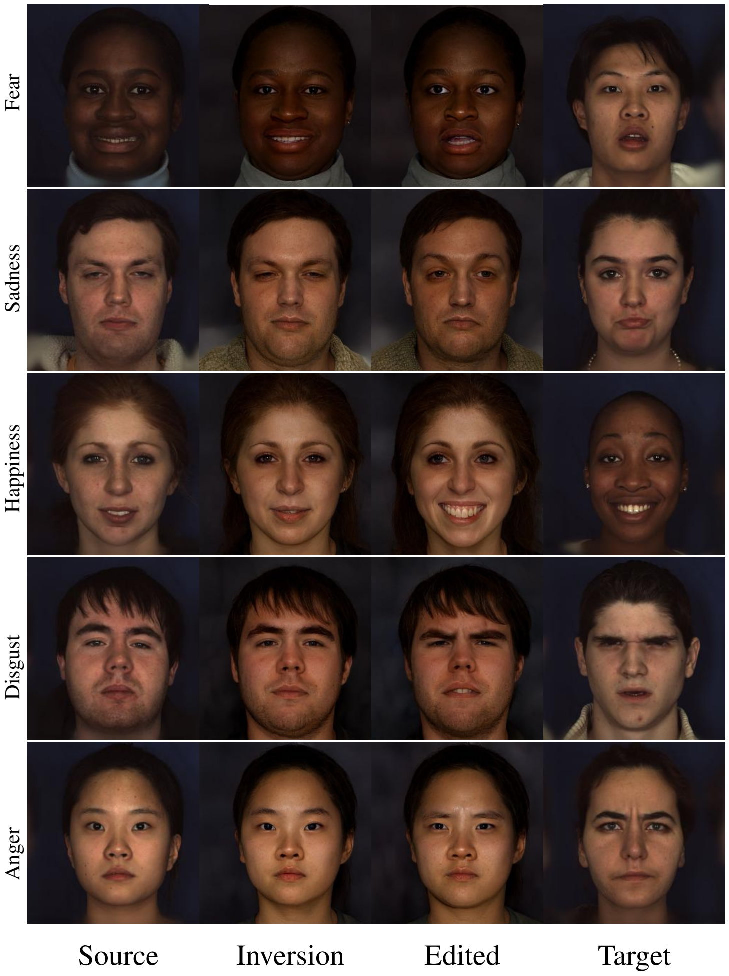


Figure 4.8: AU intensity manipulation conditioned on target images to achieve facial expression transfer on the BU-4DFE dataset. The fine-grained facial expressions, such as AU 17 (Chin Raiser) in ‘Sadness’ and AU 25 (Lips Part) in ‘Disgust’, are transferred accurately.

Raiser) and AU 12 (Lip Corner Puller) across eight levels (shown in Fig. 4.9) simultaneously to enable smile intensity editing [138]. Following the evaluation proposed in [114], we utilize a pretrained face recognition model [134] for identity preservation assessment and utilize Face++ [140] to evaluate the smile attribute intensity values in generated images.

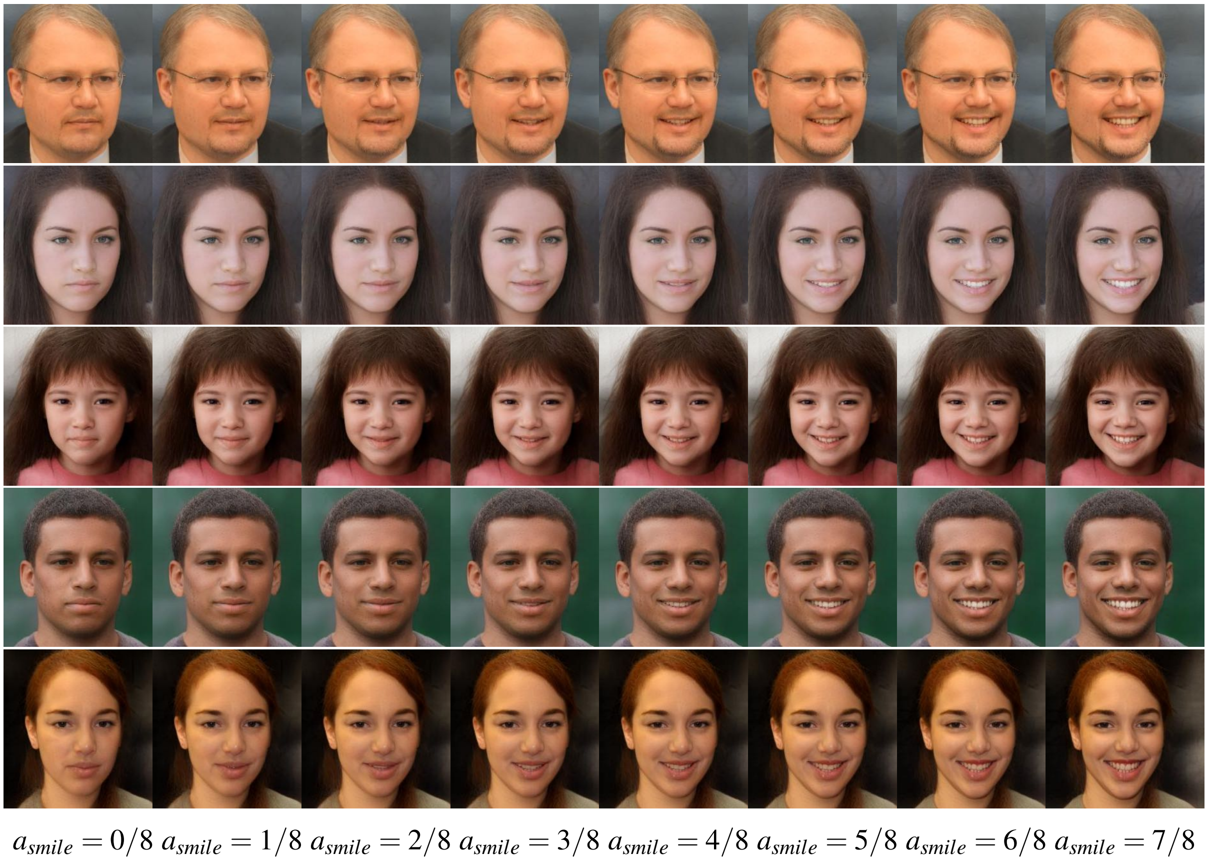


Figure 4.9: Smile attribute manipulation achieved by the AU intensity manipulation. a_{smile} denotes the target smile intensity, ranged $[0, 1]$

4.6.1 Synthetic Data Augmentation

We employ supervised AU intensity estimation as a downstream task to assess the consistency between the generated images and real images. Initially, we randomly pick a certain number of real samples with labels from the dataset. Then, we use these real images to generate an equal number of synthetic images, assigning them the provided target conditions as pseudo labels. Combining these two groups forms the training set for an AU intensity estimator. We compare the evaluation performance of this estimator trained with both real and synthetic data against one trained solely on real data. The results, depicted in Fig. 4.10, consistently demonstrate improvements with including synthetic data during training.

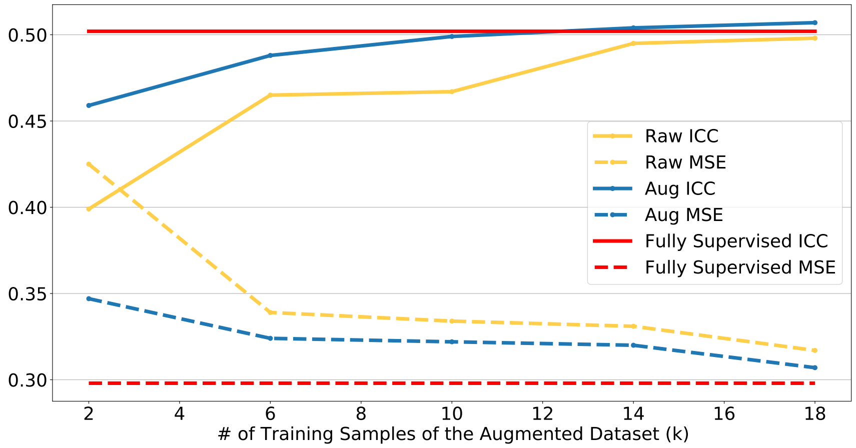


Figure 4.10: Synthetic data augmentation via AUEditNet for the AU intensity estimation task. 'Raw' denotes training with real data only. 'Aug' denotes training with both real and synthetic data. 'Fully Supervised' denotes the baseline with using all training samples of the DISFA training dataset.

4.6.2 Ablation Study

Table 4.4 shows the results of ablation studies for AUEditNet. In **module design**, the integration of dual branch (+ Dual.) leads to improvements in both AU manipulation accuracy (MSE, ICC) and ID preservation (ID). Notably, in the 'Removal' case for neutral face generation, MSE and ID get 33.5% and 26.8% improvements, respectively. The evaluation in this removal-only process is valuable for assessing whether unrelated information is introduced into the target AU space during editing, which is often invisible when 'Removal and Addition' processes are implemented. Level-wise label mapping can further improve manipulation accuracy. Regarding **training loss**, a well-trained AU intensity estimator ($\mathcal{L}_F$) plays a more crucial role than a paired target image ($\mathcal{L}_R$ & $\mathcal{L}_P$). This observation aligns with the fact that pixel-wise MSE and perceptual loss may not effectively capture AU motions. The absence of ID loss leads to a performance drop in ID. However, it also loosens constraints on latent code editing, resulting in more accurate AU manipulation.

When evaluating the performance of AUEditNet, the target image for the AU intensity estimator is the generated image with the provided target conditions. The anchor image can be

Table 4.4: Ablation Study for AUEditNet.

	Model	Target (Removal & Addition)			Neutral (Removal)	
		MSE ↓	ICC ↑	ID ↓	MSE ↓	ID ↓
Training	w/o $\mathcal{L}_R$ & $\mathcal{L}_P$	0.388	0.584	0.502	0.253	0.440
	w/o $\mathcal{L}_F$	0.507	0.356	0.480	0.467	0.454
	w/o $\mathcal{L}_{ID}$	0.288	0.619	0.533	0.115	0.515
Design	Sngl.	0.317	0.598	0.545	0.621	0.724
	+ Encoder, Decoder	0.290	0.617	0.505	0.167	0.600
	+ Dual.	0.288	0.617	0.471	0.111	0.439
	+ Label Mapping	**0.283**	**0.628**	**0.468**	**0.102**	**0.426**

Table 4.5: Comparison of AU intensity manipulation performance when using different types of anchor images. '(*Syn*)' means using synthetic face images with deactivating all AUs as the anchor image. '(*Real*)' means using real images with zero intensities of all AUs from the test subject as the anchor image. The results under the 'Real' case are copied from Table 4.1.

	Type	AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26	Avg
ICC	Real	**.848**	**.559**	.874	**.600**	.577	.230	**.890**	.276	.669	.511	**.950**	**.548**	.628
	Syn	.853	.551	**.885**	**.600**	**.586**	**.235**	.888	**.283**	**.685**	**.514**	.948	.533	**.631**
MSE	Real	.191	**.445**	.309	**.029**	.492	.579	**.228**	**.080**	.188	.322	**.169**	**.367**	.283
	Syn	**.186**	.452	**.291**	.030	**.483**	**.574**	.230	**.080**	**.181**	**.321**	.171	.377	**.281**

either a real image with zero intensities of all AUs from the test subject or the generated one with deactivating all AUs. To fully evaluate the quality of the generated images, we further implemented the comparison using real or generated images with deactivated AUs as the anchor images, as shown in Table 4.5. When using synthetic images as the anchor images, the final performance is further improved even if the external AU intensity estimator H_{est} is only trained with the real images in the training subset. Additionally, it proves AUEditNet's effectiveness in AU intensity manipulation when deactivating all AUs.

4.7 Conclusion

In this work, we achieved accurate AU intensity manipulation in high-resolution synthetic face images. Our method allows conditioning manipulation on intensity values or target images without retraining the network or requiring extra estimators. This pipeline presents a promising solution for editing facial attributes despite the dataset's limited subject count. We validated our method both qualitatively and quantitatively through extensive experiments. The performance boost with synthetic augmented data confirms the quality of generated samples paired with target conditions as pseudo labels, mitigating the challenge of data scarcity. In the future, we aim to explore weakly-supervised or self-supervised methods to further advance AU intensity manipulation.

Chapter 4, in full, is a reprint of the material as it appears in the publication of "AUEditNet: Dual-Branch Facial Action Unit Intensity Manipulation with Implicit Disentanglement,", Shiwei Jin, Zhen Wang, Lei Wang, Peng Liu, Ning Bi, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2024. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Facial attributes encompass a diverse range of features on a person's face, playing crucial roles in fields like surveillance, healthcare, and entertainment. As the development of deep neural networks, there has been a growing interest in learning-based facial attribute analysis. This learning-based analysis can be broadly classified into two main tasks: facial attribute estimation and manipulation. Facial attribute estimation predicts attributes from images, while manipulation focuses on editing the aimed attributes in images without affecting others. These tasks are interrelated and mutually beneficial. Facial attribute manipulation has the potential to enhance estimation accuracy by distilling features that are related to the aimed attributes and using synthetic samples to augment raw datasets Conversely, a well-trained attribute estimator can serve as a benchmark to guide the training of the manipulator, ensuring that the targeted attributes are edited accurately at the image level. Moreover, with the advancement of generators and their ability to produce increasingly realistic images, facial attribute manipulation can extend its capabilities beyond data augmentation, such as video conferencing, character animation, avatar manipulation, and immersive experiences in virtual and augmented reality environments.

This dissertation focuses on three facial attributes: gaze directions, head orientations, and facial action units. We proposed several learning-based algorithms to estimate and manipulate these facial attributes given input images:

- In Chapter 2, we introduced a unified eye-image-based gaze estimation algorithm, integrating Kappa Angle regression to achieve person-dependent calibration during both training and evaluation phases. Our approach leverages synthetic eye images and accounts for ocular counter-rolling (OCR) response to predict the Kappa Angle. The proposed method achieves comparable estimation accuracy with significantly lower standard deviation and requires fewer network parameters compared to existing approaches on benchmark datasets. This indicates the effectiveness of our proposed gaze estimation pipeline with bringing in OCR-aware Kappa Angle compensation. This work [141] was accepted and presented at the GAZE Workshop during CVPR 2023 and was honored with the Best Poster Award.

- In Chapter 3, we introduced an innovative gaze direction and head orientation editing algorithm. This framework operates in a latent-to-latent manner, projecting latent codes into an embedding space for an interpretable redirection process on aimed attributes, while preserving others in the initial latent space without information loss from projection-deprojection procedures. This portable framework seamlessly integrates into a pretrained GAN inversion pipeline, enabling precise redirection of gaze directions and head orientations on high-resolution full-face images, without the need for any parameter tuning of the encoder-generator pairs. This work [119] was accepted by CVPR 2023 and was showcased during the poster session.

- In Chapter 4, we introduced a method for accurately manipulating Action Unit (AU) intensities in high-resolution synthesized face images, conditioned by either AU intensity values or target images. Our approach presented an architecture specifically designed to disentangle target attributes from other facial features and identity information. Notably, this disentanglement was achieved even with limited data containing very few subjects compared to the number of target facial attributes we aimed to edit. The effectiveness of our method was demonstrated by its ability to manipulate both float and negative AU intensities while consistently generating realistic results, despite the training set labels encompassing six levels. Moreover, synthesized face images with assigned conditions, working as the pair to augment the dataset and contributing to enhancing the overall performance of AU intensity estimation. This work [142] was accepted by CVPR 2024 and was showcased during the poster session.

Chapter 2, in full, is a reprint of the material as it appears in the publication of “Kappa Angle Regression with Ocular Counter-Rolling Awareness for Gaze Estimation”, Shiwei Jin, Ji Dai, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in the publication of “ReDirTrans: Latent-to-Latent Translation for Gaze and Head Redirection”, Shiwei Jin, Zhen Wang, Lei Wang, Ning Bi, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in the publication of “AUEditNet: Dual-Branch Facial Action Unit Intensity Manipulation with Implicit Disentanglement,”, Shiwei Jin, Zhen Wang, Lei Wang, Peng Liu, Ning Bi, and Truong Nguyen, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2024. The dissertation author was the primary investigator and author of this paper.

5.2 Future Work

In future work, there are several aspects in which the proposed research could be further extended and developed.

Self-Supervised Gaze and Head Redirection State-of-the-art redirection methods [16, 17] achieved high-accuracy gaze and head redirection with the need of off-the-shelf estimators, such as gaze and head estimators. These pretrained attribute estimators were utilized to ensure the specific attribute editing in the generated results. However, a robust pretrained estimator still requires a large amount of data with attribute-related annotations. Moreover, the editing accuracy is affected or even decided by these pretrained estimators. A few work investigated and proposed self-supervised redirection methods. Yu *et al.* [15] proposed estimating warping maps given the angle difference of gaze directions between source and target eye images. Qin *et al.* [100] proposed redirecting the head orientations based on the reconstructed 3D face model. In our future work, we plan to incorporate the warping-map idea into the redirection task. Given the source image and the target gaze direction (or head orientation), we can estimate the corresponding warping map as the pseudo ground truth. Then we estimate the warping map between the source and redirected images and compare it with the aforementioned pseudo ground truth to alleviate the dependence on the pretrained estimators.

3D Facial Gaussian Avatar Animation Both ReDirTrans-GAN [119] and AUEditNet [142] are 2D-based methods designed for facial attribute manipulation. However, pure 2D solutions struggle with extreme cases. For instance, when faced with large pitch (greater than 40°) and yaw (greater than 40°) values as new head orientations, ReDirTrans-GAN [119] tends to introduce undesired changes, such as altering face shapes, in the generated images. A similar phenomenon happens in facial expression editing tasks [37]. To tackle this challenge, we aim to integrate 3D information to maintain the structural rigidity of the face. Specifically, GaussianAvatars [143] introduced a method of representing 3D facial avatars using 3D Gaussian splats attached to a parametric face model (FLAME). This approach allows for complete control and animation of facial avatars, including adjustments to pose, expression, and viewpoint. In our future work, we intend to incorporate the Gaussian splatting process into our facial attribute editing pipeline.

Bibliography

[1] X. Zheng, Y. Guo, H. Huang, Y. Li, and R. He, "A survey of deep facial attribute analysis," International Journal of Computer Vision, vol. 128, pp. 2002–2034, 2020.

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

[3] K. Alberto Funes Mora and J.-M. Odobez, "Geometric generative gaze estimation (g3e) for remote rgb-d cameras," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1773–1780, 2014.

[4] C. Gou, Y. Wu, K. Wang, F.-Y. Wang, and Q. Ji, "Learning-by-synthesis for accurate eye detection," in 2016 23rd international conference on pattern recognition (ICPR), pp. 3362–3367, IEEE, 2016.

[5] C. Gou, Y. Wu, K. Wang, K. Wang, F.-Y. Wang, and Q. Ji, "A joint cascaded framework for simultaneous eye detection and eye state estimation," Pattern Recognition, vol. 67, pp. 23–31, 2017.

[6] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza, "Hybrid method based on topography for robust detection of iris center and eye corners," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 9, no. 4, pp. 1–20, 2013.

[7] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in Proceedings of the symposium on eye tracking research and applications, pp. 207–210, 2014.

[8] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Inferring human gaze from appearance via adaptive linear regression," in 2011 International Conference on Computer Vision, pp. 153–160, IEEE, 2011.

[9] B. Noris, J.-B. Keller, and A. Billard, "A wearable gaze tracking system for children in unconstrained environments," Computer Vision and Image Understanding, vol. 115, no. 4, pp. 476–486, 2011.

[10] F. Martinez, A. Carbone, and E. Pissaloux, “Gaze estimation using local features and non-linear regression,” in 2012 19th IEEE International Conference on Image Processing, pp. 1961–1964, IEEE, 2012.

[11] Y. Sugano, Y. Matsushita, and Y. Sato, “Learning-by-synthesis for appearance-based 3d gaze estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1821–1828, 2014.

[12] K. A. Funes-Mora and J.-M. Odobez, “Gaze estimation in the 3d space using rgb-d sensors,” International Journal of Computer Vision, vol. 118, no. 2, pp. 194–216, 2016.

[13] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky, “Deepwarp: Photorealistic image resynthesis for gaze manipulation,” in European conference on computer vision, pp. 311–326, Springer, 2016.

[14] Y. Yu, G. Liu, and J.-M. Odobez, “Improving few-shot user-specific gaze adaptation via gaze redirection synthesis,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11937–11946, 2019.

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

[16] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, “Few-shot adaptive gaze estimation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9368–9377, 2019.

[17] Y. Zheng, S. Park, X. Zhang, S. De Mello, and O. Hilliges, “Self-learning transformations for improving gaze and head redirection,” Advances in Neural Information Processing Systems, vol. 33, pp. 13127–13138, 2020.

[18] Z. He, A. Spurr, X. Zhang, and O. Hilliges, “Photo-realistic monocular gaze redirection using generative adversarial networks,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6932–6941, 2019.

[19] P. Ekman and W. V. Friesen, “Facial action coding system,” Environmental Psychology & Nonverbal Behavior, 1978.

[20] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, “Openface 2.0: Facial behavior analysis toolkit,” in 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018), pp. 59–66, IEEE, 2018.

[21] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, “Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5562–5570, 2016.

[22] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “Affectnet: A database for facial expression, valence, and arousal computing in the wild,” IEEE Transactions on Affective Computing, vol. 10, no. 1, pp. 18–31, 2017.

[23] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a large-scale speaker identification dataset,” arXiv preprint arXiv:1706.08612, 2017.

[24] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, “Ganimation: Anatomically-aware facial animation from a single image,” in Proceedings of the European conference on computer vision (ECCV), pp. 818–833, 2018.

[25] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, “Ganimation: One-shot anatomically consistent facial animation,” International Journal of Computer Vision, vol. 128, pp. 698–713, 2020.

[26] J. Ling, H. Xue, L. Song, S. Yang, R. Xie, and X. Gu, “Toward fine-grained facial expression manipulation,” in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16, pp. 37–53, Springer, 2020.

[27] S. Tripathy, J. Kannala, and E. Rahtu, “Icface: Interpretable and controllable face reenactment using gans,” in Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 3385–3394, 2020.

[28] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, “Encoding in style: a stylegan encoder for image-to-image translation,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2287–2296, 2021.

[29] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8110–8119, 2020.

[30] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, “Designing an encoder for stylegan image manipulation,” ACM Transactions on Graphics (TOG), vol. 40, no. 4, pp. 1–14, 2021.

[31] Y. Alaluf, O. Patashnik, and D. Cohen-Or, “Restyle: A residual-based stylegan encoder via iterative refinement,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6711–6720, 2021.

[32] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano, “Hyperstyle: Stylegan inversion with hypernetworks for real image editing,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18511–18521, 2022.

[33] Y. Shen, C. Yang, X. Tang, and B. Zhou, “Interfacegan: Interpreting the disentangled face representation learned by gans,” IEEE transactions on pattern analysis and machine intelligence, 2020.

[34] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, “Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows,” ACM Transactions on Graphics (ToG), vol. 40, no. 3, pp. 1–21, 2021.

[35] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, “Ganspace: Discovering interpretable gan controls,” Advances in Neural Information Processing Systems, vol. 33, pp. 9841–9850, 2020.

[36] Y. Shen and B. Zhou, “Closed-form factorization of latent semantics in gans,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1532–1540, 2021.

[37] R. Haas, S. Graßhof, and S. S. Brandt, “Tensor-based emotion editing in the stylegan latent space,” arXiv preprint arXiv:2205.06102, 2022.

[38] E. Collins, R. Bala, B. Price, and S. Susstrunk, “Editing in style: Uncovering the local semantics of gans,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5771–5780, 2020.

[39] M. J. Chong, W.-S. Chu, A. Kumar, and D. Forsyth, “Retrieve in style: Unsupervised facial feature transfer and retrieval,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3887–3896, 2021.

[40] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and S. Ilya, “Learning transferable visual models from natural language supervision,” in International Conference on Machine Learning, pp. 8748–8763, PMLR, 2021.

[41] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, “Styleclip: Text-driven manipulation of stylegan imagery,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2085–2094, 2021.

[42] R. Menges, C. Kumar, D. Müller, and K. Sengupta, “Gazetheweb: A gaze-controlled web browser,” in Proceedings of the 14th International Web for All Conference, pp. 1–2, 2017.

[43] L. Fridman, B. Reimer, B. Mehler, and W. T. Freeman, “Cognitive load estimation in the wild,” in Proceedings of the 2018 chi conference on human factors in computing systems, pp. 1–9, 2018.

[44] A. Grillini, D. Ombelet, R. S. Soans, and F. W. Cornelissen, “Towards using the spatio-temporal properties of eye movements to classify visual field defects,” in Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp. 1–5, 2018.

[45] A. Tawari, K. H. Chen, and M. M. Trivedi, "Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation," in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 988–994, IEEE, 2014.

[46] J. Schwehr and V. Willert, "Driver's gaze prediction in dynamic automotive scenes," in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–8, 2017.

[47] B. I. Outram, Y. S. Pai, T. Person, K. Minamizawa, and K. Kunze, "Anyorbit: Orbital navigation in virtual environments with eye-tracking," in Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp. 1–5, 2018.

[48] Y. Zhang, L. Zhang, W. Hamidouche, and O. Deforges, "A fixation-based 360° benchmark dataset for salient object detection," in 2020 IEEE International Conference on Image Processing (ICIP), pp. 3458–3462, IEEE, 2020.

[49] E. Lindén, J. Sjostrand, and A. Proutiere, "Learning to personalize in appearance-based gaze tracking," in Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0, 2019.

[50] X. Zhang, Y. Sugano, and A. Bulling, "Revisiting data normalization for appearance-based gaze estimation," in Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp. 1–9, 2018.

[51] G. Liu, Y. Yu, K. A. F. Mora, and J.-M. Odobez, "A differential approach for gaze estimation with calibration.," in BMVC, vol. 2, p. 6, 2018.

[52] K. W. Wright, "Anatomy and physiology of eye movements," in Pediatric Ophthalmology and Strabismus, pp. 125–143, Springer, 2003.

[53] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 1, pp. 162–175, 2017.

[54] S. G. Diamond and C. H. Markham, "Ocular counterrolling as an indicator of vestibular otolith function," Neurology, vol. 33, no. 11, pp. 1460–1460, 1983.

[55] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4511–4520, 2015.

[56] K. A. Funes Mora and J.-M. Odobez, "3d gaze tracking and automatic gaze coding from rgb-d cameras," in IEEE Conference in Computer Vision and Pattern Recognition, Vision Meets Cognition Workshop, 2014.

[57] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "A 3d morphable eye region model for gaze estimation," in European Conference on Computer Vision, pp. 297–313, Springer, 2016.

[58] K. Wang and Q. Ji, "Real time eye gaze tracking with kinect," in 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2752–2757, IEEE, 2016.

[59] K. A. Funes Mora, F. Monay, and J.-M. 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, pp. 255–258, 2014.

[60] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2176–2184, 2016.

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

[62] Y. Cheng, F. Lu, and X. Zhang, "Appearance-based gaze estimation via evaluation-guided asymmetric regression," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 100–115, 2018.

[63] Y. Cheng, S. Huang, F. Wang, C. Qian, and F. Lu, "A coarse-to-fine adaptive network for appearance-based gaze estimation," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 10623–10630, 2020.

[64] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 51–60, 2017.

[65] Y. Xiong, H. J. Kim, and V. Singh, "Mixed effects neural networks (menets) with applications to gaze estimation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7743–7752, 2019.

[66] Z. Chen and B. E. Shi, "Appearance-based gaze estimation using dilated-convolutions," in Asian Conference on Computer Vision, pp. 309–324, Springer, 2018.

[67] Z. Chen and B. Shi, "Offset calibration for appearance-based gaze estimation via gaze decomposition," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 270–279, 2020.

[68] Y. Liu, R. Liu, H. Wang, and F. Lu, "Generalizing gaze estimation with outlier-guided collaborative adaptation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3835–3844, 2021.

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

[70] Y. Wang, Y. Jiang, J. Li, B. Ni, W. Dai, C. Li, H. Xiong, and T. Li, "Contrastive regression for domain adaptation on gaze estimation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19376–19385, 2022.

[71] M. A. Strobl, F. Lipsmeier, L. R. Demenescu, C. Gossens, M. Lindemann, and M. De Vos, "Look me in the eye: evaluating the accuracy of smartphone-based eye tracking for potential application in autism spectrum disorder research," Biomedical engineering online, vol. 18, no. 1, pp. 1–12, 2019.

[72] D. A. Atchison, G. Smith, and G. Smith, Optics of the human eye, vol. 35. Butterworth-Heinemann Oxford, 2000.

[73] J. Otero-Millan, C. Treviño, A. Winnick, D. S. Zee, J. P. Carey, and A. Kheradmand, "The video ocular counter-roll (vocr): a clinical test to detect loss of otolith-ocular function," Acta oto-laryngologica, vol. 137, no. 6, pp. 593–597, 2017.

[74] M. F. Reschke, S. J. Wood, and G. Clément, "Ocular counter rolling in astronauts after short-and long-duration spaceflight," Scientific reports, vol. 8, no. 1, pp. 1–9, 2018.

[75] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in European conference on computer vision, pp. 499–515, Springer, 2016.

[76] M. L R D and P. Biswas, "Appearance-based gaze estimation using attention and difference mechanism," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3143–3152, 2021.

[77] Z. Guo, Z. Yuan, C. Zhang, W. Chi, Y. Ling, and S. Zhang, "Domain adaptation gaze estimation by embedding with prediction consistency," in Proceedings of the Asian Conference on Computer Vision, 2020.

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

[79] K.-K. Oh, B.-Y. Moon, H. G. Cho, S.-Y. Kim, and D.-S. Yu, "Measurement of ocular counter-roll using iris images during binocular fixation and head tilt," Journal of International Medical Research, vol. 49, no. 3, p. 0300060521997329, 2021.

[80] H. Von Helmholtz, Handbuch der physiologischen Optik, vol. 9. Voss, 1867.

[81] A. M. Wong, "Listing's law: clinical significance and implications for neural control," Survey of ophthalmology, vol. 49, no. 6, pp. 563–575, 2004.

[82] J. M. Furman and R. H. Schor, "Orientation of listing's plane during static tilt in young and older human subjects," Vision research, vol. 43, no. 1, pp. 67–76, 2003.

[83] T. Haslwanter, D. Straumann, B. Hess, and V. Henn, "Static roll and pitch in the monkey: shift and rotation of listing's plane," Vision research, vol. 32, no. 7, pp. 1341–1348, 1992.

[84] R. J. Leigh and D. S. Zee, The neurology of eye movements. Contemporary Neurology, 2015.

[85] S. Andrist, X. Z. Tan, M. Gleicher, and B. Mutlu, "Conversational gaze aversion for humanlike robots," in 2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 25–32, IEEE, 2014.

[86] M. X. Huang, J. Li, G. Ngai, and H. V. Leong, "Stressclick: Sensing stress from gaze-click patterns," in Proceedings of the 24th ACM international conference on Multimedia, pp. 1395–1404, 2016.

[87] R. E. Kaisler and H. Leder, "Trusting the looks of others: Gaze effects of faces in social settings," Perception, vol. 45, no. 8, pp. 875–892, 2016.

[88] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. 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, pp. 1118–1130, 2017.

[89] J. Schwehr and V. Willert, "Driver's gaze prediction in dynamic automotive scenes," in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–8, IEEE, 2017.

[90] A. Burova, J. Mäkelä, J. Hakulinen, T. Keskinen, H. Heinonen, S. Siltanen, and M. Turunen, "Utilizing vr and gaze tracking to develop ar solutions for industrial maintenance," in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–13, 2020.

[91] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in vr: How do people explore virtual environments?," IEEE transactions on visualization and computer graphics, vol. 24, no. 4, pp. 1633–1642, 2018.

[92] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Gazedirector: Fully articulated eye gaze redirection in video," in Computer Graphics Forum, vol. 37, pp. 217–225, Wiley Online Library, 2018.

[93] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401–4410, 2019.

[94] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?," in Proceedings of the IEEE/CVF international conference on computer vision, pp. 4432–4441, 2019.

[95] Z. Wu, D. Lischinski, and E. Shechtman, "Stylespace analysis: Disentangled controls for stylegan image generation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12863–12872, 2021.

[96] D. Kononenko, Y. Ganin, D. Sungatullina, and V. Lempitsky, “Photorealistic monocular gaze redirection using machine learning,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 11, pp. 2696–2710, 2017.

[97] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. 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, pp. 131–138, 2016.

[98] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2107–2116, 2017.

[99] K. Wang, R. Zhao, and Q. Ji, “A hierarchical generative model for eye image synthesis and eye gaze estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 440–448, 2018.

[100] J. Qin, T. Shimoyama, and Y. Sugano, “Learning-by-novel-view-synthesis for full-face appearance-based 3d gaze estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4981–4991, 2022.

[101] Y. Dalva, S. F. Altındiş, and A. Dundar, “Vecgan: Image-to-image translation with interpretable latent directions,” in Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI, pp. 153–169, Springer, 2022.

[102] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586–595, 2018.

[103] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4690–4699, 2019.

[104] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” arXiv preprint arXiv:1710.10196, 2017.

[105] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in Proceedings of the IEEE international conference on computer vision, pp. 3730–3738, 2015.

[106] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708, 2017.

[107] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134, 2017.

[108] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or, "Face identity disentanglement via latent space mapping," arXiv preprint arXiv:2005.07728, 2020.

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

[110] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[111] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. 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, pp. 8789–8797, 2018.

[112] K. R. Scherer, "Emotion as a process: Function, origin and regulation," 1982.

[113] H. Ding, K. Sricharan, and R. Chellappa, "Exprgan: Facial expression editing with controllable expression intensity," in Proceedings of the AAAI conference on artificial intelligence, vol. 32, 2018.

[114] H. Do, E. Yoo, T. Kim, C. Lee, and J. Y. Choi, "Quantitative manipulation of custom attributes on 3d-aware image synthesis," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8529–8538, 2023.

[115] Y. Lyu, T. Lin, F. Li, D. He, J. Dong, and T. Tan, "Deltaedit: Exploring text-free training for text-driven image manipulation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6894–6903, 2023.

[116] S. Tripathy, J. Kannala, and E. Rahtu, "Facegan: Facial attribute controllable reenactment gan," in Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 1329–1338, 2021.

[117] A. Voynov and A. Babenko, "Unsupervised discovery of interpretable directions in the gan latent space," in International conference on machine learning, pp. 9786–9796, PMLR, 2020.

[118] O. K. Yüksel, E. Simsar, E. G. Er, and P. Yanardag, "Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14263–14272, 2021.

[119] S. Jin, Z. Wang, L. Wang, N. Bi, and T. Nguyen, "Redirtrans: Latent-to-latent translation for gaze and head redirection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5547–5556, 2023.

[120] W. S. Noble, “What is a support vector machine?,” Nature biotechnology, vol. 24, no. 12, pp. 1565–1567, 2006.

[121] Y. Zhu, H. Liu, Y. Song, Z. Yuan, X. Han, C. Yuan, Q. Chen, and J. Wang, “One model to edit them all: Free-form text-driven image manipulation with semantic modulations,” Advances in Neural Information Processing Systems, vol. 35, pp. 25146–25159, 2022.

[122] C. Yang, Y. Shen, and B. Zhou, “Semantic hierarchy emerges in deep generative representations for scene synthesis,” International Journal of Computer Vision, vol. 129, pp. 1451–1466, 2021.

[123] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in 3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society, 2015.

[124] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14, pp. 694–711, Springer, 2016.

[125] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, “Disfa: A spontaneous facial action intensity database,” IEEE Transactions on Affective Computing, vol. 4, no. 2, pp. 151–160, 2013.

[126] S. M. Mavadati, M. H. Mahoor, K. Bartlett, and P. Trinh, “Automatic detection of non-posed facial action units,” in 2012 19th IEEE International Conference on Image Processing, pp. 1817–1820, IEEE, 2012.

[127] I. Ntinou, E. Sanchez, A. Bulat, M. Valstar, and Y. Tzimiropoulos, “A transfer learning approach to heatmap regression for action unit intensity estimation,” IEEE Transactions on Affective Computing, 2021.

[128] T. Song, Z. Cui, Y. Wang, W. Zheng, and Q. Ji, “Dynamic probabilistic graph convolution for facial action unit intensity estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4845–4854, 2021.

[129] W. Li, F. Abtahi, Z. Zhu, and L. Yin, “Eac-net: Deep nets with enhancing and cropping for facial action unit detection,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 11, pp. 2583–2596, 2018.

[130] Z. Shao, Z. Liu, J. Cai, and L. Ma, “Jaa-net: joint facial action unit detection and face alignment via adaptive attention,” International Journal of Computer Vision, vol. 129, pp. 321–340, 2021.

[131] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu, “A high-resolution spontaneous 3d dynamic facial expression database,” in 2013 10th IEEE

international conference and workshops on automatic face and gesture recognition (FG), pp. 1–6, IEEE, 2013.

[132] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, pp. 2223–2232, 2017.

[133] P. E. Shrout and J. L. Fleiss, "Intraclass correlations: uses in assessing rater reliability.," Psychological bulletin, vol. 86, no. 2, p. 420, 1979.

[134] A. Geitgey, "Github – face recognition." `https://github.com/ageitgey/face_recognition/`, 2021.

[135] D. Linh Tran, R. Walecki, O. Rudovic, S. Eleftheriadis, B. Schuller, and M. Pantic, "Deepcoder: Semi-parametric variational autoencoders for automatic facial action coding," in Proceedings of the IEEE International Conference on Computer Vision, pp. 3190–3199, 2017.

[136] E. Sanchez, M. K. Tellamekala, M. Valstar, and G. Tzimiropoulos, "Affective processes: stochastic modelling of temporal context for emotion and facial expression recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9074–9084, 2021.

[137] Y. Jiang, Z. Huang, X. Pan, C. C. Loy, and Z. Liu, "Talk-to-edit: Fine-grained facial editing via dialog," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13799–13808, 2021.

[138] J. M. Girard, G. Shandar, Z. Liu, J. F. Cohn, L. Yin, and L.-P. Morency, "Reconsidering the duchenne smile: indicator of positive emotion or artifact of smile intensity?," in 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), pp. 594–599, IEEE, 2019.

[139] F. P. Papantoniou, P. P. Filntisis, P. Maragos, and A. Roussos, "Neural emotion director: Speech-preserving semantic control of facial expressions in" in-the-wild" videos," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18781–18790, 2022.

[140] Face++, "Face detection api." `https://www.faceplusplus.com/`, Accessed: 2023-11-15.

[141] S. Jin, J. Dai, and T. Nguyen, "Kappa angle regression with ocular counter-rolling awareness for gaze estimation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2658–2667, 2023.

[142] S. Jin, Z. Wang, L. Wang, P. Liu, N. Bi, and T. Nguyen, "Aueditnet: Dual-branch facial action unit intensity manipulation with implicit disentanglement," arXiv preprint arXiv:2404.05063, 2024.

[143] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner, "Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians," arXiv preprint arXiv:2312.02069, 2023.