Unsupervised High-Resolution Portrait Gaze Correction and Animation

Jichao Zhang, Jingjing Chen, Hao Tang, Enver Sangineto,
Peng Wu, Yan Yan, Nicu Sebe, *Senior Member, IEEE,* Wei Wang

***Abstract*—This paper proposes a gaze correction and animation method for high-resolution, unconstrained portrait images, which can be trained without the gaze angle and the head pose annotations. Common gaze-correction methods usually require annotating training data with precise gaze, and head pose information. Solving this problem using an unsupervised method remains an open problem, especially for high-resolution face images in the wild, which are not easy to annotate with gaze and head pose labels. To address this issue, we first create two new portrait datasets: CelebGaze (256×256) and high-resolution CelebHQGaze (512×512). Second, we formulate the gaze correction task as an image inpainting problem, addressed using a Gaze Correction Module (GCM) and a Gaze Animation Module (GAM). Moreover, we propose an unsupervised training strategy, i.e., Synthesis-As-Training, to learn the correlation between the eye region features and the gaze angle. As a result, we can use the learned latent space for gaze animation with semantic interpolation in this space. Moreover, to alleviate both the memory and the computational costs in the training and the inference stage, we propose a Coarse-to-Fine Module (CFM) integrated with GCM and GAM. Extensive experiments validate the effectiveness of our method for both the gaze correction and the gaze animation tasks in both low and high-resolution face datasets in the wild and demonstrate the superiority of our method with respect to the state of the art.**

***Index Terms*—Generative Adversarial Networks (GANs), Facial Attribute Manipulation, Gaze Correction.**

I. Introduction

The goal of the gaze correction task is to manipulate the gaze direction of a face image with respect to a specific target direction. The main application of this task is altering the eye appearance so that the person's gaze is directed into the camera. For example, shooting a good portrait is challenging as the subjects may be too nervous to stare at the camera. Another scenario is videoconferencing, where eye contact is very important. The gaze can express attributes such as attentiveness and confidence. Unfortunately, eye contact is frequently lost during a video conference, as the participants look at the monitors and not directly into the camera. Moreover, some works use gaze redirection to improve few-shot gaze estimation task [1], [2].

Fig. 1. Left: 256×256 images and the corresponding gaze-corrected results generated by our method using samples of the CelebGaze dataset. Right: 512×512 high-resolution images and the gaze-corrected results using samples of our dataset CelebHQGaze. The first and second rows show the original images and eye-gaze corrected results, respectively.

Early works in gaze correction relied on special hardware, such as stereo cameras [3], [4], Kinect sensors [5] or transparent mirrors [6], [7]. Recently, a few methods based on machine learning showed a good quality synthetic for gaze correction. For instance, Kononenko and Lempitsky [8] propose to solve the problem of monocular gaze correction using decision forests. DeepWarp [9] uses a deep network to directly predict an image-warping flow field with a coarse-to-fine learning process. However, this method fails in generating photo-realistic images when the gaze redirection involves large angles. Moreover, it produces unnatural eye shapes because of the $L1$ loss, which is used to learn the flow field without any geometric-based regularization. To solve this problem, PRGAN [10] proposes to exploit adversarial learning with a cycle-consistent loss to generate more plausible gaze redirection results. However, these methods [8]–[10] fail in obtaining high-quality gaze redirection results in the wild when there are large variations in the head pose and the gaze angles. Recently, Marcel *et al.* [11] proposed a content-consistent model for realistic eye generation. However, their approach is based on semantic segmentation masks, which implies a great annotation effort. Another category of works is based on 3D models without training data, such as GazeDirector [12]. The main idea of GazeDirector is to model the eye region in 3D instead of predicting a flow field directly from an input image. However, modeling in 3D has strong assumptions that

This research was supported by the PRIN project CREATIVE (Prot. 2020ZSL9F9), and by the EU H2020 projects AI4Media (No. 951911) and SPRING (No. 871245).

Jichao Zhang, Nicu Sebe, and Wei Wang are with the Department of Information Engineering and Computer Science (DISI), University of Trento, Italy. E-mail: jichao.zhang@unitn.it.

Jingjing Chen and Peng Wu are from the Zhejiang University and the Shandong University of China, respectively.

Hao Tang is with the Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland. E-mail: hao.tang@vision.ee.ethz.ch.

Enver Sangineto is with the Department of Engineering (DIEF), University of Modena and Reggio Emilia, Modena, Italy. E-mail: enver.sangineto@unimore.it.

Yan Yan is with the Department of Computer Science of Illinois Institute of Technology, United States. E-mail: yyan34@iit.edu

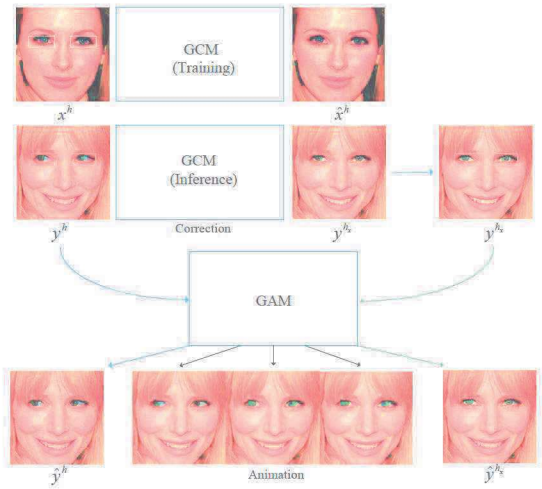

Fig. 2. Overview of the proposed architecture. We have two main modules: Gaze Correction Module for performing gaze correction (GCM) and Gaze Animation Module for performing gaze animation (GAM). Moreover, we propose to use the gaze-corrected samples from GCM to train GAM (Synthesis-as-Training). The trained GAM can achieve gaze animation by interpolating the latent feature. The white boxes are the eye mask to remove the eye region. The gray boxes represent the cropping of eye region

do not hold in non-laboratory scenarios.

The unsupervised method can avoid expensive annotations. Moreover, it has essential significance for image representation and semantic disentanglement. Thus, we recently proposed a novel gaze-correction method, GazeGAN [13], which is extended in this paper. In [13], we collected the CelebGaze dataset, which consists of two image domains: X, with eyes staring at the camera, and Y, with eyes looking somewhere else (see Fig. 1, *left*). Note that the CelebGaze images do not annotate the gaze angle or the head pose. Moreover, in [13] we propose an unsupervised learning method for gaze correction and animation, which consists of two main modules: the Gaze Correction Module (GCM) and the Gaze Animation Module (GAM). GCM is an inpainting model, trained on a domain X, which learns how to fill in the missing eye regions with a new content representing the gaze-corrected eyes. GAM is another inpainting model used for gaze animation, and it is trained on a domain Y. To generalize the gaze redirection to various angle directions (i.e., in "animations"), we propose a training method (Synthesis-As-Training) that uses synthetic data to train GAM and encourages the encoded features of the eye region to be correlated with the gaze angle. Then, gaze animation can be achieved by interpolating these features in the latent space.

In this paper, we extend GazeGAN [13] to work also with higher resolution portrait images. Specifically, we first create a new dataset, CelebHQGaze, containing 512×512 high-resolution portrait images, as shown in Fig. 1 (*right*). Second, we propose a novel GCM and GAM integrated with a coarse-to-fine module (CFM). In more detail, CFM first allows the inpainting model to be trained using low-resolution images for coarse-grained image generation. Then it uses a global nonparametric model, Laplacian Reconstruction, and a local parametric model, Local-Refinement Autoencoder, to compensate for the high-frequency information loss and to remove possible artifacts for the eye region. Utilizing this new architecture, we can avoid training each module using high-resolution images. CFM speeds up both the training and the inference process while obtaining high-quality results, comparable with directly training with high-resolution images.

Similar to GazeGAN [13], an autoencoder is pretrained using self-supervised mirror learning (PAM), where the bottleneck features are used as an extra input to the dual inpainting model to preserve the identity of the corrected results. Moreover, global and local discriminators are used to improve the visual quality of the generated samples. Finally, our qualitative and quantitative evaluations show that our method generates higher-quality results with respect to the state-of-the-arts in both the gaze correction and the gaze animation tasks.

We summarize below our main contributions:

1) Introduce an unsupervised inpainting architecture for high-resolution gaze correction and animation.
2) Propose a novel CFM module that can alleviate both the memory and the computational costs in the training and inference stages while achieving high-quality results comparable with training with high-resolution facial images.
3) Propose a gaze animation module and a Synthesis-As-Training method to generate gaze-correction results with variable angles.
4) Make publicly available the CelebHQGaze dataset for the research community interested in gaze correction and animation: https://github.com/zhangqianhui/GazeAnimationV2.

II. Related Work

Generative Adversarial Networks. Generative Adversarial Networks (GANs) [14] are powerful generative models which learn a distribution that mimics a given target distribution. They have been applied to many fields, such as low-level image processing tasks (e.g., image inpainting [15], [16], image super-resolution [17]–[19]), semantic and style transfer (e.g., image translation [20]–[26], image attribute manipulation [27]–[32], person image synthesis [33]–[35], image manipulation [36]).

Image Inpainting. Image Inpainting is an important task in computer vision and computer graphics, and it aims to fill in the missed/masked pixels of an image utilizing plausible synthesized content. Most of the previous methods can be split into two classes. The first is based on diffusion or patch-based approaches, which rely on handcrafted low-level features. For example, PatchMatch [37] is a fast nearest-neighbor field algorithm, which can perform real-time image inpainting. Generally speaking, this class of methods is based on low-level features. They are usually ineffective in filling in the missing part of an image when the underlying semantic structure is not trivial and cannot generate novel objects that cannot be found in other non-masked parts of the source image. The second class of methods is based on learning approaches. Recently, CNN-based and GAN-based methods have shown promising performance on image inpainting [38]–[41]. For instance, inpainting can be used for facial attributes manipulation such as hair, mouth, and eyes [42]–[44]. We also adopt an inpainting approach, differently from previous

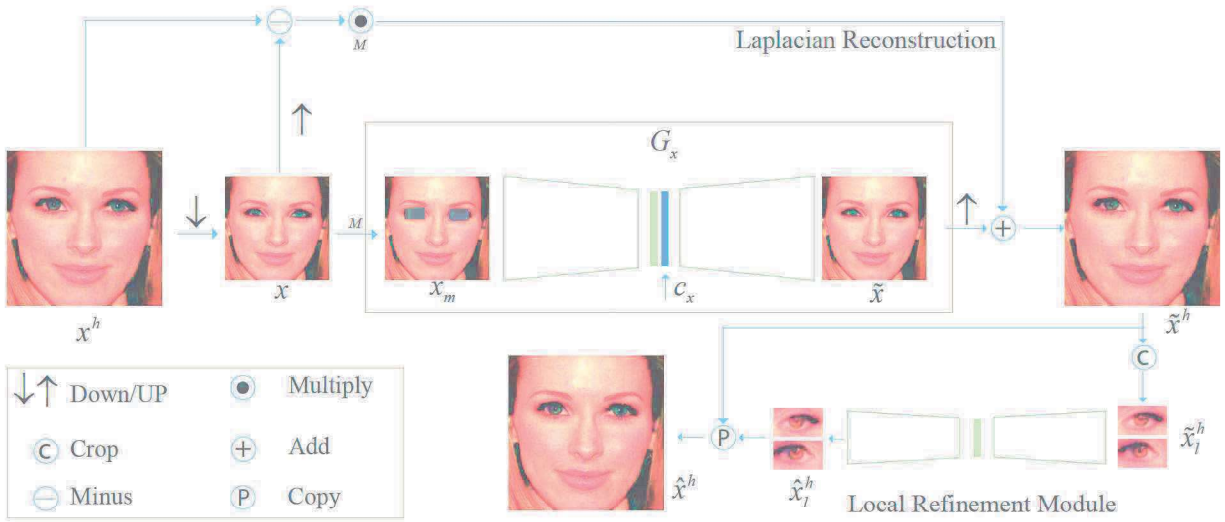

Fig. 3. Overview of the architecture for Gaze Correction Module (GCM) integrated with Coarse-to-Fine Module (CFM) which consists of one laplacian reconstruction and one local-refinement module. CFM first allows the inpainting network G_x trained using low-resolution images to attain coarse-grained inpainted results, then attains high-resolution results by the global nonparametric Laplacian reconstruction, and finally exploits a parametric local-refinement module (LRM) to compensate for high-frequency information and remove artifacts for the eye region. We use $2\times$ scales for downsampling and upsampling.

work, our method does not require the data to be labeled with additional information, such as semantic labels, sketches, or reference images.

Gaze Correction. Previous work for gaze correction can be split into three main classes: 1) hardware-driven, 2) rendering and synthesis, 3) learning-based.

The hardware support was indispensable in early research. Kollarits *et al.* [6] use half-silvered mirrors to place the camera on the optical path of the display. Yang *et al.* [4] address the eye contact problem with a view synthesis, and they use a pair of calibrated stereo cameras jointly with a face model to track the head pose in 3D. Generally speaking, these hardware-based methods are expensive.

The second class of approaches typically renders the eye region based on a 3D fitting model, which replaces the original eyes with synthetic eyeballs. Banf *et al.* [45] use an example-based approach for deforming the eyelids and sliding the iris across the model surface with a texture-coordinate interpolation. To fix the limitations caused by the use of a mesh, where the face and eyes are mixed, GazeDirector [12] separately deals with the face and eyeballs, synthesizing more high-quality images, especially for large redirection angles. These methods usually struggle in realistically rendering the corrected eyes. Additionally, modeling methods have strong assumptions that usually do not hold in practice.

Concerning the third class of methods, the core idea for most of the learning-based approaches is to use a large paired training dataset to train a statistical model [1], [8], [46], [47]. Some methods [8], [46] learn to generate the flow field, which is then used to relocate the eye pixels in the original image. For instance, Ganin *et al.* [9] use a CNN to learn the flow field, which warps the input image and redirects the gaze to the target angle. However, [9] fails to generate photo-realistic and natural shapes because it uses only pixel-wise differences between the input and the ground truth as the training loss. To address this problem, He *et al.* [10] use adversarial learning, jointly with a cycle-consistent loss, which can improve the visual quality and the redirection precision. However, these methods can hardly generate plausible results in the wild, i.e., in a scenario with large variations in the head pose, the gaze angle, or the illumination conditions. In contrast, we propose to use dual inpainting modules (GCM and GAM) to correct the gaze angle and achieve high-resolution and high-quality gaze redirection in the wild.

III. Method

The overview of our method is shown in Fig. 2 and our model consists of two main modules: Gaze Correction Module and Gaze Animation Module. Specifically, Fig. 3 illustrates Gaze Correction Module (GCM), integrating with Coarse-to-Fine Module (CFM), which is trained using the sample x from domain X. Fig. 4 illustrates Gaze Animation module (GAM), integrating with CFM, which are trained using the sample y from domain Y, and GAM exploits the corrected samples for training to make the eye feature correlate with the gaze angle (Synthesis-as-Training method). Additionally, Fig. 5 shows the pretrained autoencoder (PAM), which extracts the angle-invariant content feature as the additional input of GCM and GAM. We here clarify the adopted notations.

- $x \in R^{m \times n \times 3}$ is an image instance, where m and n are the image height and width.
- The training set is split into two domains: X, containing images with a gaze staring at the camera, and Y, containing images with a gaze staring somewhere else. X^h and Y^h correspond to the higher-resolution sets of X and Y, respectively.
- $M \in R^{m \times n \times 3}$ denotes a binary mask function of the eye region and $M^{'}$ defines the operation of extracting a rectangular sub-image (the eye region).
- P_x and P_y denote the data distributions in X and Y, respectively. P_m indicates the distribution of the masked data $M(x)$, where the eye region is removed from x. If $x \in X$ and $y \in Y$, both $M(x)$ and $M(y)$ have the same distribution, because the only difference between x and y is in the eye region. Thus, $M(x) \sim P_m$ and $M(y) \sim P_m$.

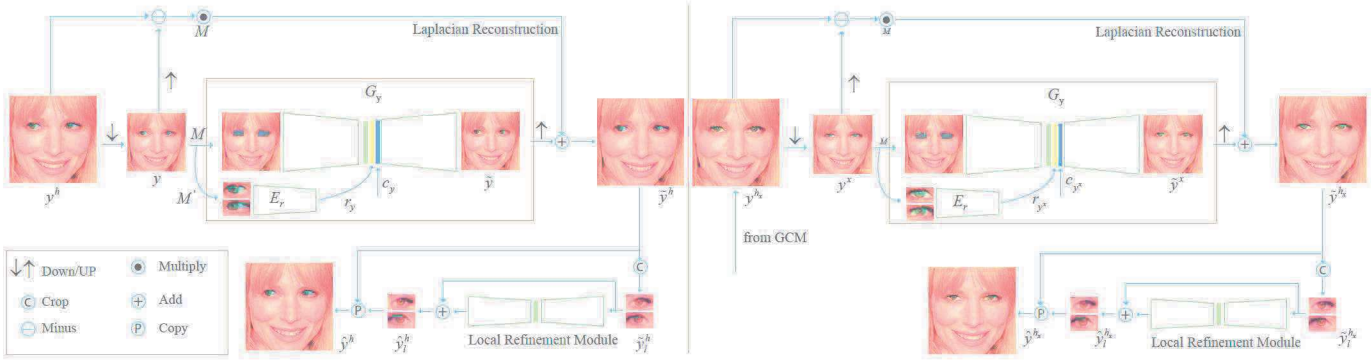

Fig. 4. An overview of the proposed Gaze Animation Module (GAM) integrated with Coarse-to-Fine Module (CFM). In the left, G_y uses the sample $y \in Y$ for training. Compared to G_x, the decoder of G_y has an extra input r which is provided by the encoder E_r. In the right, we use GCM to generate the gaze-corrected image y^{h_x}, which is then used for training G_y (Synthesis-as-Training). With the paired samples y^h and y^{h_x} for training G_y, the feature from E_r would be correlated with gaze angle, and gaze animation can be achieved by interpolating the feature.

- $r \in R^{128}$ and $c \in R^{256}$ denote the angle, the content features (being the latter angle invariant), respectively and different encoders extract them.
- F is the image horizontal flipping operation (mirroring).

A. Coarse-to-Fine Module

In order to alleviate the memory costs and reduce the number of overall training parameters while simultaneously being able to generate high-resolution facial images, we propose a CFM for GCM and GAM. This module consists of a global nonparametric Laplacian Reconstruction for the inpainting process and a local parametric Local Refinement Module (LRM) which will be introduced with details, taking GCM as an example.

1) Global Nonparametric Laplacian Reconstruction: As shown in Fig. 3, the high-resolution image x^h is downsampled by a factor of 2, obtaining x, where the latter is as input to the inpainting networks of GCM. The generated image is $\tilde{x}$. Let $u(.)$ be an upsampling operator which smooths and expands x to the original size (i.e., the resolution of x^h). The single-level Laplacian pyramid p can be obtained by:

$$p = x^h - u(x). \tag{1}$$

Then, the reconstruction process for the high-resolution image $\tilde{x}^h$ is:

$$\tilde{x}^h = u(\tilde{x}) + M(p), \tag{2}$$

where we use M to remove the eye region from p which is replaced by the zero. Then we introduce the local refinement process to improve the visual quality and remove the artifacts of the eye regions.

2) Local Parametric LRM: We use $M^{'}(\tilde{x}^h)$ to extract the local eye region $\tilde{x}_l^h$. Then, we utilize one autoencoder G_h together with residual image learning, to refine $\tilde{x}_l^h$ and get $\hat{x}_l^h$. Finally, the high-resolution complete image $\hat{x}^h$ can be obtained by replacing the local eye region $\tilde{x}_l^h$ with $\hat{x}_l^h$.

B. Gaze Correction Module

As shown in Fig. 3, we first downsample x_h to attain the low-resolution x, and then take x as the input of inpainting network G_x whose goal is to fill in the masked eye region of $x_m = M(x)$ by generating the missing eyes. This can be formulated as:

$$c_x = E_c(M^{'}(x)), \tilde{x} = G_x(M(x), c_x), \tag{3}$$

where c_x are the content (angle-invariant) features encoded using only the eye regions as input $(M^{'}(x))$ of the content encoder E_c. E_c is the encoder of G_{pre} which will be introduced in Sec. III-D.

In principle, G_x can learn the mapping from $M(x) \sim P_m$ to $x \sim P_x$ by training. Given one sample $y \sim P_y$, then, we remove the eye region to get $M(y) \sim P_m$, because x and y have different distributions only in the eye region. Thus G_x can be used to map $M(y)$ into the $G_x(M(y)) \sim P_x$ which is the intuitive basis of our correction module. This can be formulated as:

$$c_y = E_c(M^{'}(y)), y^x = G_x(M(y), c_y), \tag{4}$$

where c_y are the content (angle-invariant) features encoded using only the eye regions as input $M^{'}(y)$ of the content encoder E_c.

We train the GCM using high-resolution face images based on the CFM module. At the inference time, for the corrected result y^x, we get a high-resolution result y^{h_x} by compensating for the high-frequency details using the laplacian reconstruction and LRM. Note that the corrected result y^{h_x} is also used for training GAM. More details can be found below.

C. Gaze Animation Module

Besides correcting the gaze to stare at the camera, a more general task is gaze animation, where the gaze direction should be modified. As shown in Fig. 4, another generator G_y is used to in-paint the face image without the eye region by performing the reconstruction learning. Moreover, we extend a new eye encoder E_r for the eye region to extract the angle-specific feature, guiding the gaze redirection generation of the G_y. To achieve the disentanglement of the features, we propose a Synthesis-As-Training method, in which we use the gaze-corrected generated images as training data for training GAM. In detail, our GAM is split into two stages. In the first stage (Left of Fig. 4), we downsample y^h to get y, and train the generator G_y to fill-in the missing eye regions of an image

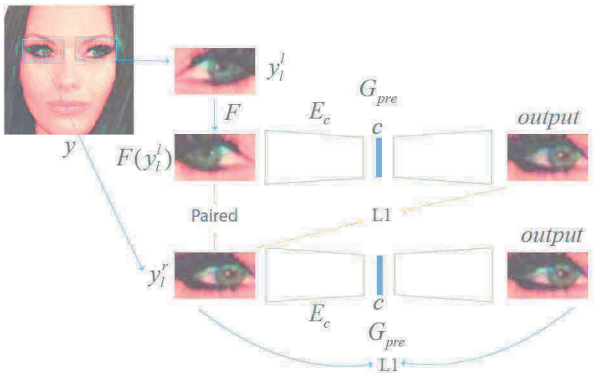

Fig. 5. The overview of pretrained autoencoder module with generator G_{pre} (PAM). PAM is trained by a self-supervised learning strategy. In detail, we crop y to both left eye y_l^l and right eye y_l^r, then, flip y_l^l by F to attain the pairs $F(y_l^l)$ and y_l^r which have similar identity, but different gaze angle. Then, the pairs would be used to train G_{pre} by reconstructing y_l^r.

($y_m = M(y)$) and produce $\tilde{y}$. G_y encodes the eye region with the latent code $r_y \in R^{128}$ by means of the encoder E_r. Moreover, r_y is used as an extra input for the decoder in G_y. In this way, we can condition G_y using the gaze-dependent feature r_y.

$$\begin{aligned} r_y = E_r(M^{'}(y)), c_y = E_c(M^{'}(y)) \\ \tilde{y} = G_y(M(y), r_y, c_y). \end{aligned} \tag{5}$$

In the second stage (Right of Fig. 4), we use GCM to correct the gaze of y^h, and it produces the synthetic sample y^{h_x}. Then, y^{h_x} is downsampled to y^x which is used for training G_y, just like y does. With the paired samples (y, y^x), which have the same masked region $M(y)$ but different eye regions, we train GAM to ensure that the encoded feature from E_r has a high correlation with the gaze angle:

$$\begin{aligned} r_{y^x} = E_r(M^{'}(y^x)), c_{y^x} = E_c(M^{'}(y^x)) \\ \tilde{y}^x = G_y(M(y^x), r_{y^x}, c_{y^x}). \end{aligned} \tag{6}$$

Then, we can attain high-resolution results $\hat{y}^h$ and $\hat{y}^{h_x}$ by compensating for the high-frequency details using the laplacian reconstruction and LRM.

D. Pretraining using Self-Supervised Learning

Preserving the consistency of the person's identity (e.g., the iris color, the eye shape) is difficult with the inpainting-based method described. To mitigate this problem, we propose to use the third generator G_{pre}, which is trained (PAM) to learn a latent representation of the content features (c), conditioning both G_x and G_y to preserve the identity information of the generated results consistent with the input.

G_{pre} is pre-trained using a self-supervised learning framework. Although our training dataset is collected from the Internet, most images have a roughly frontal pose. As shown in Fig. 5, we can easily collect paired eye-region images: The right eye y_l^r is paired with the mirrored version $F(y_l^l)$ of the left eye y_l^l. Note that y_l^r and $F(y_l^l)$ have different gaze angles, but they have a similar eye shape and iris color. Because they belong to the same person. The same holds for y_l^l and $F(y_l^r)$. Note that the only information we need to collect these pairs is the eye region position (i.e., $M(x)$). At the same time, the mirroring operation ($F(\cdot)$) is a data augmentation technique commonly used in other self-supervised learning approaches. We use these paired samples to pretrain G_{pre} using the following objective function:

$$\begin{aligned} \mathcal{L}_{pre} = & \quad \|y_l^l - G_{pre}(y_l^l)\|_1 + \|y_l^l - G_{pre}(F(y_l^r)\|_1 \\ + & \quad \|y_l^r - G_{pre}(y_l^r)\|_1 + \|y_l^r - G_{pre}(F(y_l^l))\|_1. \end{aligned} \tag{7}$$

After training, the bottleneck features c of G_{pre} are almost angle-invariant, representing only the content information (e.g., iris color, eye shape). Thus, we use the encoder network E_c of G_{pre} to provide extra input to G_x and G_y by means of its content features (see Sec. III-B and III-C). Conditioning the generation process of G_x and G_y using these content features, the inpainted results are more consistent with respect to the input identity information.

E. Loss Functions

Reconstruction Losses. We use a standard pixel-wise loss ($L1$) for training GCM. It is defined as:

$$\mathcal{L}_{re}^x = \|x - \tilde{x}\|_1 + \|x_l^h - \hat{x}_l^h\|_1, \tag{8}$$

where x_l^h and $\hat{x}_l^h$ are the eye regions of x^h and $\hat{x}^h$, respectively.

And, the reconstruction loss for GAM is defined as:

$$\mathcal{L}_{re}^y = \|y - \tilde{y}\|_1 + \|y_l^h - \hat{y}_l^h\|_1. \tag{9}$$

The reconstruction loss of GAM for the synthesis-as-training method is defined as:

$$\mathcal{L}_{re}^{y^x} = \|y^x - \tilde{y}^x\|_1 + \|y_l^{h_x} - \hat{y}_l^{h_x}\|_1. \tag{10}$$

We use $\mathcal{L}_{re}^y + \mathcal{L}_{re}^{y^x}$ as the objective function to train G_y.

Global and Local Discriminators for Adversarial Learning. Since the $L1$ loss tends to produce blurry results [20], we use three different discriminators D_x, D_y and D_h, adversarially trained together with G_x, G_y and G_h, respectively. Moreover, inspired by [39], our discriminators D_x and D_y are composed of a global part, taking the whole face as input, and a local part with taking only the local eye region as input. The global part is used to coherent the entire image as a whole, while the local part makes the local region more realistic and sharper. We concatenate the final fully-connected feature maps of both parts, which are fed to a sigmoid function to predict the probability of the image being real.

Different from D_x and D_y, D_h consists of only a local discriminator, which uses the eye regions $\hat{x}_l^h$ and $\hat{y}_l^h$ as fake inputs. In practice, we use crops slightly larger than the eye region as the input of the discriminator to alleviate the boundary mismatch problem. The objective function of D_x and G_x is defined as:

$$\begin{aligned} \min_{G_x} \max_{D_x} \mathcal{L}_{adv}^x = & \quad \mathbb{E}_x[logD_x(x, M^{'}(x))] \\ + & \quad \mathbb{E}_{\tilde{x}}[log(1 - D_x(\tilde{x}, M^{'}(\tilde{x})))] \\ + & \quad \mathbb{E}_{\tilde{y}^x}[log(1 - D_x(\tilde{y}^x, M^{'}(\tilde{y}^x)))]. \end{aligned} \tag{11}$$

The objective function of D_y and G_y is defined as:

$$\begin{aligned}\min_{G_y}\max_{D_y}\mathcal{L}_{adv}^{y} &= \mathbb{E}_{y}[logD_y(y, M^{'}(y))] \\ &+ \mathbb{E}_{\tilde{y}}[log(1-D_y(\tilde{y}, M^{'}(\tilde{y})))]. \quad (12)\end{aligned}$$

Finally, the objective function of D_h and G_h is:

$$\begin{aligned}\min_{G_h}\max_{D_h}\mathcal{L}_{adv}^{h} &= \mathbb{E}_{x^h}[logD_h(M^{'}(x^h))] \\ &+ \mathbb{E}_{\hat{x}_l^h}[log(1-D_h(\hat{x}_l^h))] \\ &+ \mathbb{E}_{y^h}[logD_h(M^{'}(y^h))] \\ &+ \mathbb{E}_{\hat{y}_l^h}[log(1-D_h(\hat{y}_l^h))]. \quad (13)\end{aligned}$$

F. Overall Objective Function

Inspired by [48], we use a latent-space reconstruction loss (l_{fp}) for the content features in the latent space to preserve further the identity information between the input image and the gaze-corrected result:

$$\mathcal{L}_{fp} = \|c_y - E_c(M^{'}(\tilde{y}))\|_1 + \|c_{y^x} - E_c(M^{'}(\tilde{y}^x))\|_1. \quad (14)$$

We use $-\ell_{adv}^{x}$, $-\ell_{adv}^{y}$ to train D_x and D_y, respectively. Concerning G_x, its overall loss is defined as:

$$\mathcal{L}_{all}^{x} = \mathcal{L}_{adv}^{x} + \mathcal{L}_{adv}^{h} + \lambda_1\mathcal{L}_{re}^{x}. \quad (15)$$

For G_y and E_r, the overall loss is defined as:

$$\begin{aligned}\mathcal{L}_{all}^{y} &= \mathcal{L}_{adv}^{y} + \mathcal{L}_{adv}^{h} + \lambda_2\mathcal{L}_{adv}^{x} \\ &+ \lambda_3\mathcal{L}_{re}^{y} + \lambda_4\mathcal{L}_{re}^{y^x} + \lambda_5\mathcal{L}_{fp}. \quad (16)\end{aligned}$$

$\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_5 are hyper-parameters controlling the contribution of each loss term.

G. Gaze Correction and Animation at Inference Time

At inference time, given an image sample y^h, we first downsample it to attain y, then obtain the gaze-corrected result y^x using G_x, and finally output high-resolution results y^{h_x} by using CFM, which can compensate for high-frequency texture details. In the case of gaze animation, as shown in Fig. 4, we modify the angle representation r by interpolating between r_y and r_{y^x}, where both features correspond to the encoded angle features of the eye region y_l and the eye region y_l^x, respectively. The interpolated angle representation can be fed to G_y to obtain an intermediate result. We can produce high-resolution gaze animation results using CFM.

IV. Experiments

This section introduces the details of our datasets, our network training, and baseline models. Then, we compare the proposed method with the state-of-the-art methods of gaze correction in the wild using both qualitative and quantitative evaluations. Next, we demonstrate the effectiveness of the proposed method on gaze animation with various outputs by interpolating and extrapolating in the latent space. Finally, we present detailed ablation studies to validate the effect of each component of our model, i.e., the Synthesis-As-Training method, the Pretrained Autoencoder (PAM) with self-supervised mirror learning, the Latent Reconstruction Loss, and the Coarse-to-Fine Module (CFM). For brevity, we refer to the method presented in [13] as **GazeGAN** and the extended version introduced in this paper as **GazeGANV2**. Note that we do not use any post-processing step for GazeGAN and GazeGANV2.

A. Datasets

Most of the existing benchmarks [50]–[53] do not contain enough image variability (e.g., a wide gaze-direction range, various head poses, and different illumination conditions) for our gaze correction task in the wild. Recently, [54] presented a large-scale gaze tracking dataset, called Gaze360, for robust 3D gaze estimation in unconstrained images. Although this dataset has been labeled with a 3D gaze direction with a wide range of angles and head poses, it still lacks high-resolution images for face and eye regions. Moreover, this dataset does not provide annotations of the eye gaze staring at the camera, which is required in our domain set X. More recently, [55] proposed a large scale (over 1 million samples) of high-resolution images for gaze estimation. However, these images are collected in laboratory conditions and are not suitable for our gaze correction task in the wild. To remedy this problem, we propose collecting new datasets consisting of lots of high-resolution portraits without labelling head poses and gaze information. In detail, five volunteers are asked to divide the row data (face) into two domains according to whether the face eyes are staring at the camera. The gaze and head estimation model can automate 'Staring at the camera' annotation. However, the existing state-of-the-art methods [54], [56] cannot achieve accurate gaze estimation for CelebHQGaze, as an overlarge domain shift exists between training data and test data.

CelebGaze. CelebGaze consists of 25,283 celebrity images, most of which have been collected from CelebA [57] and a minority from the Internet. There are 21,832 face images with the eyes staring at the camera and 3,451 face images with the eyes looking somewhere else. We crop all the images to 256×256 and compute the eye mask region using Dlib [58]. Specifically, we use Dlib to extract 68 facial landmarks, and we compute the mean of 6 points near the eye region, which is the center point of the mask. The size of the mask is fixed to 30×50. We randomly select 300 samples from domain Y and 100 samples from domain X as their corresponding test sets, and we use the remaining images for the training set. Note that this dataset is unpaired and not labeled with the specific eye angle or the head pose information.

CelebHQGaze. CelebHQGaze consists of 29,255 high-resolution celebrity images that are collected from CelebA-HQ [59]. It consists of 21,005 face images with the eyes staring at the camera and 8,250 face images with eyes looking somewhere else. Similarly to CelebGaze, we extract facial landmarks and generate the mask. All images are cropped to 512×512, and the mask size is fixed to 46×80. Similar to the CelebGaze dataset, also for the CelebHQGaze, we randomly select 300 samples from domain Y and 100 samples from domain X for the test set, and we use all the remaining images for the training set. We show two samples of the CelebHQGaze dataset in Fig. 1.

Fig. 6. Qualitative comparison for the gaze correction task on the CelebGaze dataset. The first row shows the input images, and the following rows show the gaze correction results of StarGAN [49], CycleGAN [22], PRGAN [10], GazeGAN and GazeGANV2. Magnified left eyes are shown in the last column. Zoom in for the best of view.

B. Training Details

We first train the PAM module. Then, the discriminators D_x, D_y and D_h and the generators G_x and G_y and G_h are jointly optimized. We use the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The batch size is 16 for CelebGaze and 8 for CelebHQGaze. The initial learning rate is 0.0005 for PAM, 0.0004 for G_h, and 0.0001 for the three discriminators and the two generators in the first 20,000 iterations. The learning rate is linearly decayed to 0 over the remaining iterations. The loss coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are all set to 1, while λ_5 is 0.1. To stabilize the network training in the adversarial learning, we use spectral normalization [60] for all the conv-layers of the three discriminators, but not for the generators. Our method is implemented in Tensorflow and trained with a single NVIDIA Titan X GPU.

C. Baseline Models

Gaze Correction. PRGAN [10] achieved state-of-the-art gaze redirection results on the Columbia gaze dataset [51] based on a single encoder-decoder network with adversarial learning, similarly to the StarGAN architecture [49]. The original PRGAN is trained on paired samples with labeled angles. To train PRGAN on the proposed CelebGaze and CelebHQGaze datasets, we remove the VGG perceptual loss of PRGAN, and learn the gaze redirection task between domain X and Y. We train PRGAN only with the local eye region, the same way as the original paper.

Facial Attribute Manipulation. Gaze correction and animation can be regarded as a sub-task of facial attribute manipulation. Recently, StarGAN [49] achieved very high-quality results in facial attribute manipulation. We train StarGAN on the CelebGaze dataset to learn the translation mapping between domain X and domain Y.

Moreover, gaze correction can be considered as an image translation task. Thus, we adopt CycleGAN as another baseline for our experiments. Note that we do not compare GazeGAN with AttGAN [28], STGAN [30], RelGAN [61], CAFE-GAN [62], SSCGAN [32] as they have a performance very close to StarGAN in the facial attribute manipulation task. We use the public code of StarGAN [1], CycleGAN [2] and PRGAN [3].

D. Gaze Correction

This section qualitatively and quantitatively compares the proposed method with state-of-the-arts on both CelebGaze and CelebHQGaze datasets for the gaze correction task.

Qualitative Results. As shown in the last row of Fig. 6 and Fig. 7, GazeGANV2 can correct the eyes to look at the camera while preserving the identity information such as the eye shape and the iris color, validating the effectiveness of the proposed method. The 2nd row of the figure shows the results of StarGAN [49]. We note that StarGAN could not produce precise gazes staring at the camera, and it suffers from a low-quality generation with lots of artifacts (Zoom in for the best of view). The results of CycleGAN are shown in the 3th row. Although the results of CycleGAN are very realistic and with few artifacts in the eye region, this method does not produce a precise correction of the gaze direction (e.g., see the magnified eye regions of Fig. 6 and Fig. 7). We explain that both StarGAN and CycleGAN use the cycle-consistency loss, which requires that the mapping between

[1] https://github.com/yunjey/StarGAN
[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix
[3] https://github.com/HzDmS/gaze_redirection

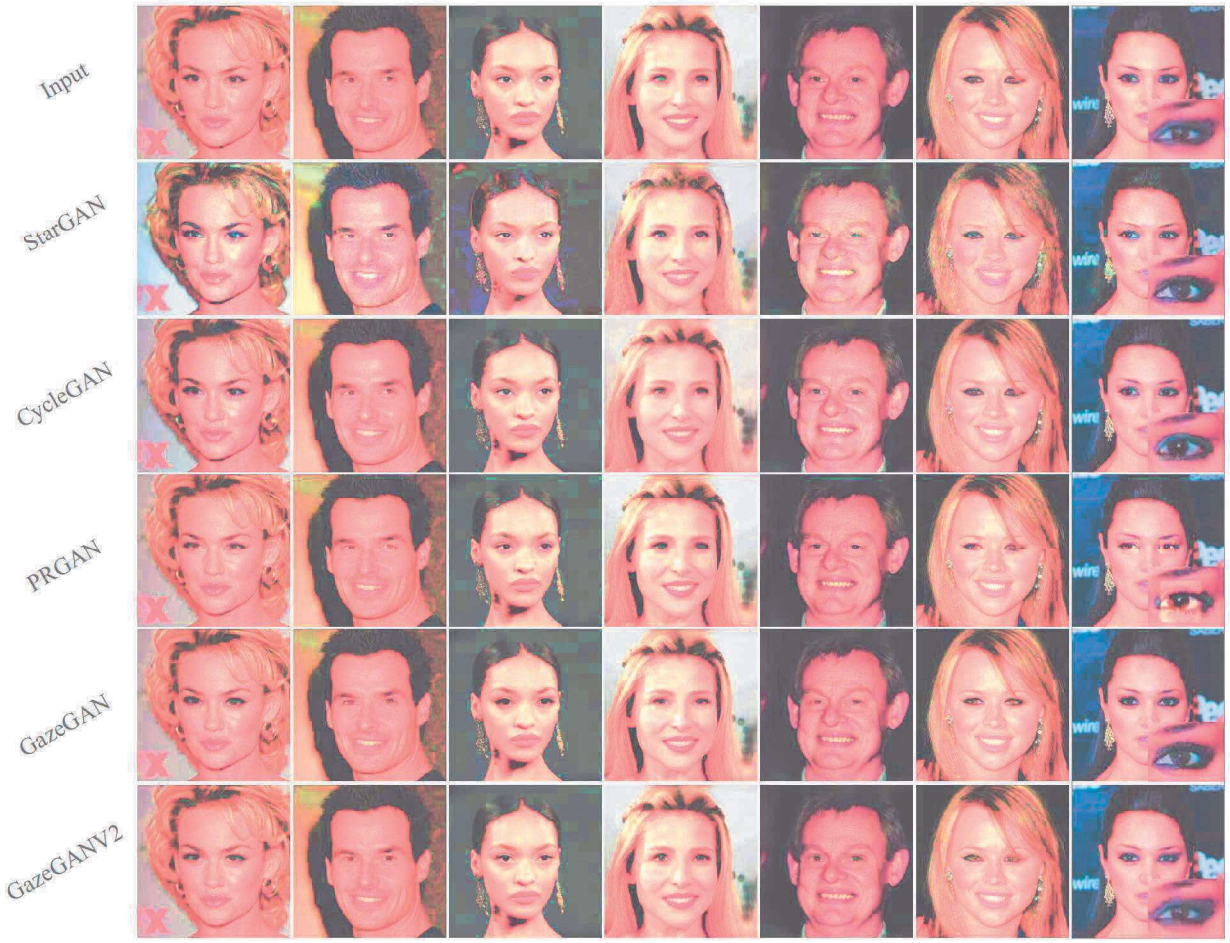

Fig. 7. Qualitative comparison for the gaze correction task on CelebHQGaze dataset. The first row shows the input images, and the following rows show the gaze correction results of StarGAN [49], CycleGAN [22], PRGAN [10], GazeGAN and GazeGANV2. Magnified left eyes are shown in the last column. Zoom in for the best of view.

TABLE I
QUANTITATIVE RESULTS ON BOTH THE CELEBGAZE AND THE CELEBHQGAZE DATASET. THE HIGHER IS BETTER FOR MSSSIM AND THE USER STUDY; THE LOWER IS BETTER FOR LPIPS AND FID. THE COLUMNS PARAMS AND FPS REPORT THE CORRESPONDING NETWORK PARAMETERS AND FRAME PER SECOND AT TEST TIME, RESPECTIVELY. US: USER STUDIES.

Method	CelebGaze						CelebHQGaze					
	MSSSIM ↑	LPIPS ↓	FID ↓	US ↑	Params ↓	FPS ↓	MSSSIM ↑	LPIPS ↓	FID ↓	US ↑	Params ↓	FPS ↓
Other	-	-	-	24.20%	-	-	-	-	-	23.20%	-	-
StarGAN [49]	0.96	0.073	82.49	3.400%	-	-	0.94	0.084	185.47	4.400%	-	-
CycleGAN [22]	0.99	0.026	70.12	15.00%	-	-	0.98	0.028	**53.690**	8.670%	-	-
PRGAN [10]	1.00	0.000	84.61	8.330%	-	-	1.00	0.000	106.79	22.40%	-	-
GazeGAN	**1.00**	**0.000**	62.12	22.40%	73.26M	30.29	**1.00**	**0.000**	60.520	25.50%	183.2M	23.20
GazeGANV2	**1.00**	**0.000**	**56.37**	**32.40%**	**47.20M**	**38.40**	**1.00**	**0.000**	63.590	**27.30%**	**84.18M**	**27.70**
GT	1.00	0.000	-	100%	-	-	1.00	0.000	-	100%	-	-

X and Y be continuous and invertible. According to the invariance of the Domain Theorem[4], the intrinsic dimensions of the two domains should be the same. However, the intrinsic dimension of Y is much larger than X, as Y has more variations for the gaze angle than X. Moreover, we compare GazeGANV2 with PRGAN [10]. PRGAN is trained using only local eye regions (same as in the original paper), which may help focus on the translation of the eye region. The results of PRGAN are shown in the 4th row of Fig. 6. Compared with GazeGANV2, PRGAN does not produce precise and realistic correction results. Additionally, PRGAN suffers from the boundary mismatch problem between the local eye region and the global face (see the last column of Fig. 7).

Finally, as shown in the last rows of both Fig. 6 and Fig. 7, comparing GazeGANV2 with GazeGAN, we observe that both models can produce realistic and faithfully results. Additionally, we show more results of portraits with a diverse head pose. Fig. 8 shows that our model can achieve acceptable gaze-correction results for portraits with different head poses.

Quantitative Evaluation Protocol. The qualitative evaluation has validated the effectiveness and the superiority of our proposed GazeGANV2 in the gaze correction task. To further

[4]https://en.wikipedia.org/wiki/Invariance_of_domain

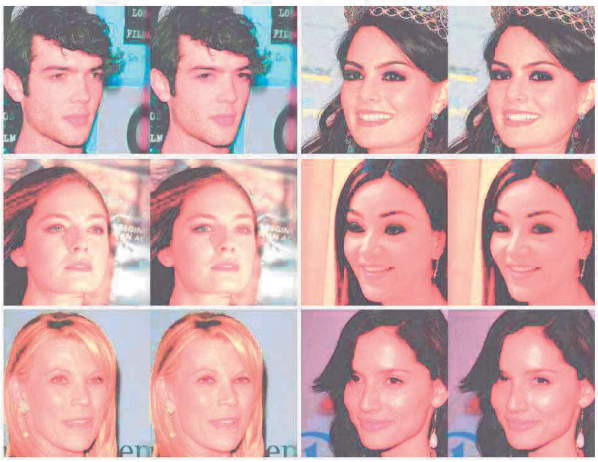

Fig. 8. More gaze correction results to show that our model can handle diverse head poses.

support the previous evaluation with quantitative results, we use the MSSSIM [63] and the LPIPS [64] metrics to measure the preservation ability of the *irrelevant regions*, i.e.,, the whole image except the eye region ($M(y^h)$). Specifically, we compute the mean MSSSIM and LPIPS scores between $M(y^h)$ and $M(\hat{y}^h)$ across all the *test* data of Y^h. Moreover, the Fréchet Inception Distance (FID) [65] has been shown to correlate well with the human judgment and has become a popular metric for GAN-based methods. We use it to evaluate the quality of the generated eye region for the gaze correction and the gaze animation tasks.

In addition to the aforementioned automatic metrics, we conduct a user study to compare the results of the gaze correction task of different models. In detail, given an input face image of the CelebGaze or CelebHQGaze test dataset (extracted from Y), we show the gaze-corrected results produced by different models to 30 respondents, who were asked to select the best image based on the perceptual realism and the precision of the gaze correction. They also can select "Other", which means that the results of all the models are not satisfactory enough. This study is based on 50 questions (i.e., 50 randomly sampled images) for each respondent.

Quantitative Results. The first two columns of the left part (CelebGaze) and the right part (CelebHQGaze) of Table I show the MSSSIM and LPIPS scores evaluating the preservation ability of the corrected images using different methods. GazeGANV2 and PRGAN obtain the best results, with 1.0 for MSSSIM and 0.0 for LPIPS. The original irrelevant regions are integrated with the generated eye region in both models using binary masks. StarGAN and CycleGAN get the worse irrelevant region preservation scores. The FID scores of the eye regions are reported in the 3rd column. In the CelebGaze dataset, GazeGANV2 and GazeGAN outperform all the other methods, reaching comparable scores on the CelebHQGaze dataset. Though CycleGAN has the best FID scores, it fails to generate precise gaze correction results. The penultimate column of both parts in Table I shows the evaluation results of the user study. For the CelebGaze dataset, the average vote for GazeGANV2 is higher than for all the other methods. The same conclusions can be drawn with the CelebHQGaze dataset. Importantly, GazeGANV2 achieves a performance very close to GazeGAN. However, it has fewer parameters and higher FPS, as shown in the last two columns of Table I.

TABLE II
A COMPARISON ON THE GAZE ANIMATION TASK BETWEEN GAZEGAN AND GAZEGANV2 WITH RESPECT TO THE GENERATION QUALITY.

Method	CelebGaze		CelebHQGaze	
	GazeGAN	GazeGANV2	GazeGAN	GazeGANV2
FID ↓	80.31	**53.32**	**70.56**	71.37

Overall, the qualitative and quantitative evaluations demonstrate the effectiveness and superiority of our approach.

E. Gaze Animation

The bottom of Fig. 9 and 10 show gaze animation results using input images with various gaze directions. The latent-space interpolation results are smooth and plausible in each row. Each column has a different gaze direction angle, but the identity information is overall preserved (e.g., the eye shape, the iris color, etc.).

The top rows of Fig. 9 and 10 show a gaze animation comparison between GazeGAN and GazeGANV2. GazeGANV2 can produce more realistic images with fewer artifacts than GazeGAN on the CelebGaze dataset, while they have comparable performance on the CelebHQGaze dataset. The quantitative result confirms it in Table II.

Finally, we show gaze animation results obtained by *extrapolating* the features r, in addition to using interpolation methods. With "extrapolation," we mean that we use values of r which lie in the line connecting r_y with r_{y^*}, but they are outside these two points. As shown in Fig. 12, our method not only achieves high-quality interpolation results but can also produce plausible gaze animations for gaze angles outside the range between the input and the gaze-corrected output.

F. Ablation Study

In this section, we conduct extensive ablation studies to investigate the contribution of each of four critical components of our proposed GazeGANV2, i.e., the Pretrained Autoencoder for content feature extraction, the Synthesis-As-Training method, the Latent Reconstruction Loss $\mathcal{L}_{fp}$ and the Coarse-to-Fine Module. We refer to these components as A, B, C, and D, respectively.

Pretrained Autoencoder (PAM). Sec. III-D shows how self-supervised learning is used to pretrain a content encoder. This encoder produces identity-specific features which condition the generation process of G_x and G_y.

Fig. 11 shows that our full-model (GazeGANV2) can better preserve identity information with respect to GazeGANV2 W/O A. To quantify this, we use G_x to reconstruct test image samples x ($x \in X$), and we measure the difference between the input image and the gaze-corrected result in local eye regions employing both MSSSIM and LPIPS metrics. Table III shows that GazeGANV2 gets better scores than GazeGANV2 W/O A, confirming our design motivation.

Fig. 9. Gaze animation results using the interpolation of the latent features r on the CelebGaze dataset. The top two rows show the images generated by GazeGAN and GazeGANV2, respectively, jointly with the eye regions. The other rows show the gaze animation results of GazeGANV2. The first and the last columns show the input images and the gaze-corrected results, respectively. The middle columns show the interpolated images.

TABLE III
COMPARISON BETWEEN GAZEGANV2 AND GAZEGANV2 W/O A, WHERE THE LATTER DENOTES REMOVING THE CONTENT REPRESENTATION EXTRACTED FROM E_c. THE SCORES ARE MEASURED BETWEEN THE INPUT IMAGE x AND INPAINTED RESULT $\tilde{x}$ ACROSS THE TEST DATA FROM X. NOTE THAT EVALUATION SAMPLES ARE FROM CELEBHQGAZE.

Metrics	GazeGANV2	GazeGANV2 W/O A
MSSSIM ↑	**0.6080**	0.5230
LPIPS ↓	**0.1680**	0.2646

TABLE IV
COMPARISON WITH GAZEGAN W/O C, WHICH DENOTES REMOVING THE LATENT RECONSTRUCTION LOSS $\mathcal{L}_{fp}$. THE SCORES ARE MEASURED BETWEEN THE INPUT IMAGE y AND THE RECONSTRUCTION RESULT $\tilde{y}$ ACROSS ALL THE TEST DATA OF DOMAIN Y. THE EVALUATION IS BASED ON THE CELEBHQGAZE DATASET.

Metrics	GazeGANV2	GazeGANV2 W/O C
MSSSIM ↑	**0.6290**	0.6100
LPIPS ↓	**0.2328**	0.2372

A shown in Fig. 13, we visualize the outputs of autoencoder with taking y_l^l, y_l^r, $F(y_l^l)$ and y_l^r as inputs after training. We can observe that the model can attain the similar reconstruction results for y_l^l and $F(y_l^r)$ as inputs, and can also attain the similar reconstruction results for y_l^r and $F(y_l^l)$ as inputs which validates the effectiveness of the objective loss.

Synthesis-As-Training Method. The gaze animation results in Fig. 9 and 10 show the effectiveness of our method in disentangling the angle representation. Fig. 16 (top) shows a t-SNE visualization of points interpolated in the latent space. In more detail, following the procedure explained in Sec. III-G, we uniformly interpolate the line connecting r_y with r_{y^x} in the angle latent space using 5 interpolation points $(I_1, ...I_5)$ for each sample y. Fig. 16 (top) shows that for each specific sample y, these five interpolation points are different from each other but strongly clustered together, which illustrates the disentanglement of the angle latent space.

TABLE V
COMPARISON BETWEEN GAZEGANV2 AND GAZEGANV2 W/O D WITH RESPECT TO THE GENERATION QUALITY IN GAZE ANIMATION.

Method	CelebGaze		CelebHQGaze	
	GazeGANV2	W/O D	GazeGANV2	W/O D
FID ↓	**53.32**	78.32	**71.37**	74.04

TABLE VI
QUANTITATIVE COMPARISON BETWEEN CFM OF GAZEGANV2 WITH BILINEAR AND SUPER-RESOLUTION MODEL ESRGAN [19].

Metrics	Bilinear	ESRGAN	CFM
MSSSIM ↑	0.9563	0.9595	**0.9827**
LPIPS ↓	0.2393	0.1476	**0.1039**
FPS ↓	30.600	4.3000	**27.700**
Params ↓	**48.88M**	80.88M	49.23M

Latent Reconstruction Loss $\mathcal{L}_{fp}$. We use G_y to fill in the eye region of test images y ($y \in Y$), and we measure the difference between the input images and the generated results employing MSSSIM and LPIPS. In Table IV, GazeGANV2 obtains better scores than GazeGANV2 W/O C, which shows

Fig. 10. Gaze animation results using the interpolation of the latent features r on the CelebHQGaze dataset. The top two rows show the images generated by GazeGAN and GazeGANV2, respectively, jointly with the eye regions. The other rows show the gaze animation results of GazeGANV2. The first and the last columns show the input images and gaze-corrected results, respectively. The middle columns show the interpolated images.

Fig. 11. A qualitative comparison between GazeGANV2 (4th row), GazeGANV2 W/O A (3rd row), and GazeGANV2 W/O D (2nd row).

that $\mathcal{L}_{fp}$ further improves the ability to preserve identity information. Moreover, we visualize the content features c_y and $c_{\tilde{y}}$ extracted from real samples y and reconstructed samples $\tilde{y}$ across all the Y test data. As shown in Fig. 16 (bottom), we observe that using our full model GazeGANV2, c_y and $c_{\tilde{y}}$ usually lie closer to each other to what happens when using GazeGANV2 W/O C, and it shows that this loss helps to represent content information consistently.

Coarse-to-Fine Module (CFM). The previous experiments validate the effectiveness of CFM. As shown in the 2nd and 4th row of Fig. 11, the gaze correction results of GazeGANV2 are more realistic than those produced by GazeGANV2 W/O D. In Table V, the quantitative comparison between GazeGAN and GazeGAN W/O D confirms the effectiveness of CFM. Fig. 15 shows the differences in the gaze correction results obtained with GazeGAN and GazeGAN W/O D. The heatmap of the difference between the two generated images shows that CFM can compensate for the high-frequency information loss of the coarse output. Finally, we compare our CFM with some upsampling methods, such as Bilinear and super-resolution method, ESRGAN [19]. By taking all samples y^h from domain Y^h, we attain all low-resolution reconstructed results $\tilde{y}$. Then, 3 different methods are used for upsampling them to attain high-resolution results. Fig. 14 shows our method achieves better reconstruction with fewer artifacts, such as eye regions. Quantitative experiments of Table VI show our CFM achieves better MSSSIM and LPIPS scores and has higher FPS and fewer parameters than ESRGAN.

V. Conclusion

We introduce a new high-resolution gaze dataset in the wild, CelebHQGaze, which is characterized by a large diversity in head poses and gaze angles. Moreover, we propose a novel unsupervised method, GazeGANV2, for gaze-direction correction and animation. GazeGANV2 formulates the gaze correction problem as an inpainting task and uses a coarse-to-fine learning strategy to generate high-resolution images. Moreover, self-supervised learning and Synthesis-As-Training methods are used to disentangle the content and angle-specific features, which can condition the generation process. The

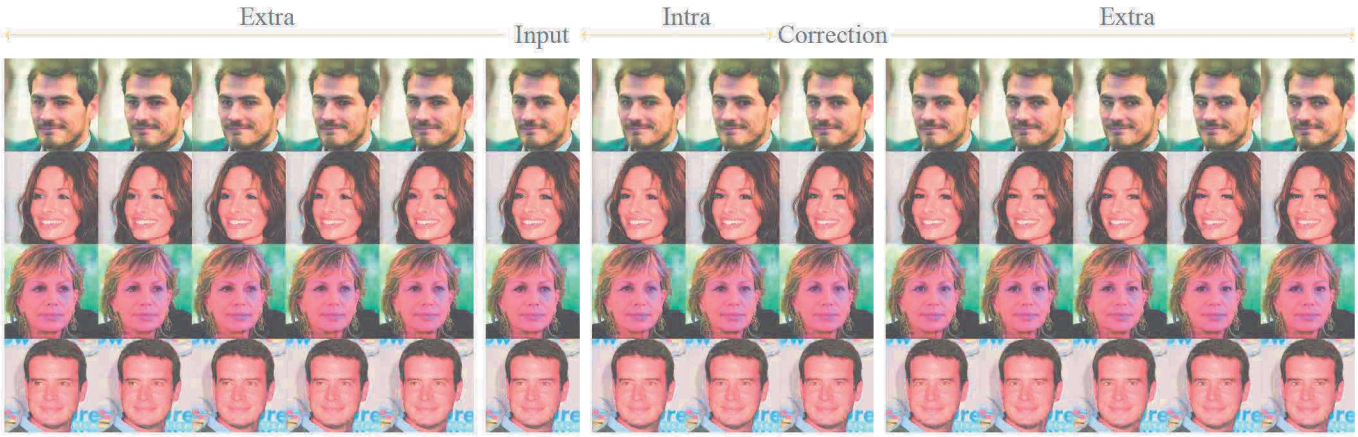

Fig. 12. Gaze animation examples are obtained by both interpolation and extrapolation of the latent features r. Extra: extrapolation; Intra: interpolation.

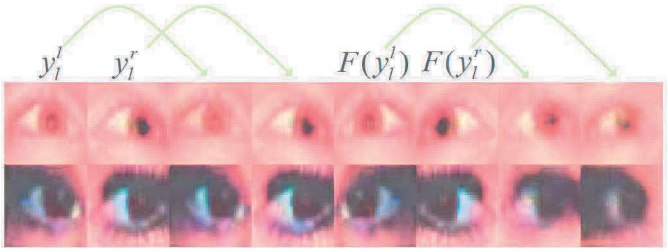

Fig. 13. The visualization for the results of PAM with taking y_l^l, y_l^r, $F(y_l^l)$ and $F(y_l^r)$ as inputs, respectively. The arrows point to the generated sample.

Fig. 14. Qualitative comparison between GazeGANV2 with Bilinear and super-resolution model ESRGAN [19]. Note that the input image would be downsampled $2\times$ as input of the inpainting model.

qualitative and quantitative results demonstrate the method's effectiveness and its superiority to the state of the arts.

Fig. 15. A qualitative comparison between the gaze-correction results produced by GazeGANV2 (3rd column) and GazeGANV2 W/O D (2nd column). The 1st column shows the input image, and the final column is a heatmap of the difference between the 3rd column and 2nd column. This residual image clearly shows semantic and texture information.

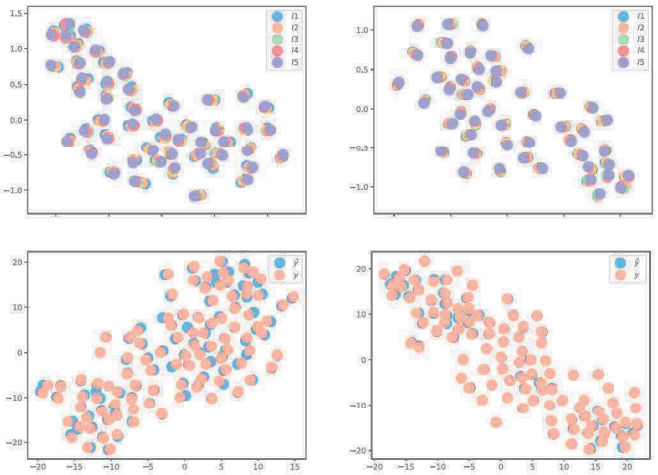

Fig. 16. Top: a t-SNE based visualization of the latent space r which represents the gaze angle (Sec. III-C).We show the corresponding latent spaces of GazeGAN (top-left) and GazeGANV2 (top-right). We plot 5 interpolated points ($I1 - I5$) for each image and we use 50 images. Bottom: t-SNE visualization of the content features c_y (orange) and $c_{\tilde{y}}$ (blue) extracted from y and $\tilde{y}$, respectively. Bottom-Left: GazeGANV2 W/O C; Bottom-Right: GazeGANV2.

REFERENCES

[1] Y. Yu, G. Liu, and J.-M. Odobez, "Improving few-shot user-specific gaze adaptation via gaze redirection synthesis," in *CVPR*, 2019.
[2] Y. Yu and J.-M. Odobez, "Unsupervised representation learning for gaze estimation," in *CVPR*, 2020.
[3] A. Criminisi, J. Shotton, A. Blake, and P. H. Torr, "Gaze manipulation for one-to-one teleconferencing." in *ICCV*, 2003.
[4] R. Yang and Z. Zhang, "Eye gaze correction with stereovision for video-teleconferencing," in *ECCV*, 2002.
[5] C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross, "Gaze correction for home video conferencing," *TOG*, 2012.
[6] R. Kollarits, C. Woodworth, J. Ribera, and R. Gitlin, "34.4: An eye contact camera/display system for videophone applications using a conventional direct-view lcd," in *Society for Information Display, International Symposium*, 1996.
[7] K.-I. Okada, F. Maeda, Y. Ichikawaa, and Y. Matsushita, "Multiparty videoconferencing at virtual social distance: Majic design," in *ACM conference on Computer supported cooperative work*, 1994.
[8] D. Kononenko and V. Lempitsky, "Learning to look up: Realtime monocular gaze correction using machine learning," in *CVPR*, 2015.
[9] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky, "Deepwarp: Photorealistic image resynthesis for gaze manipulation," in *ECCV*. Springer, 2016.
[10] Z. He, A. Spurr, X. Zhang, and O. Hilliges, "Photo-realistic monocular gaze redirection using generative adversarial networks," in *ICCV*, 2019.
[11] M. C. Buehler, S. Park, S. D. Mello, X. Zhang, and O. Hilliges, "Content-consistent generation of realistic eyes with style," in *ICCVW*, 2019.
[12] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Gazedirector: Fully articulated eye gaze redirection in video," in *CGF*, 2018.

[13] J. Zhang, J. Chen, H. Tang, W. Wang, Y. Yan, E. Sangineto, and N. Sebe, "Dual in-painting model for unsupervised gaze correction and animation in the wild," in *ACM MM*, 2020.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.
[15] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.
[16] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and Locally Consistent Image Completion," *TOG*, 2017.
[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.
[18] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *ECCV*, 2018.
[19] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *ECCVW*, 2018.
[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.
[21] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, "Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation," in *CVPR*, 2020.
[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.
[23] S. Liu, Y. Sun, D. Zhu, R. Bao, W. Wang, X. Shu, and S. Yan, "Face aging with contextual generative adversarial nets," in *ACM MM*, 2017.
[24] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *CVPR*, 2019.
[25] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," *arXiv preprint arXiv:2007.15651*, 2020.
[26] A. Mallya, T.-C. Wang, K. Sapra, and M.-Y. Liu, "World-consistent video-to-video synthesis," *arXiv preprint arXiv:2007.08509*, 2020.
[27] J. Zhang, Y. Shu, S. Xu, G. Cao, F. Zhong, M. Liu, and X. Qin, "Sparsely grouped multi-task generative adversarial networks for facial attribute manipulation," in *ACM MM*, 2018.
[28] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," *TIP*, 2019.
[29] Z. He, M. Kan, J. Zhang, and S. Shan, "Pa-gan: Progressive attention generative adversarial network for facial attribute editing," *arXiv preprint arXiv:2007.05892*, 2020.
[30] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen, "Stgan: A unified selective transfer network for arbitrary image attribute editing," in *CVPR*, 2019.
[31] X. Chen, B. Ni, N. Liu, Z. Liu, Y. Jiang, L. Truong, and Q. Tian, "Coogan: A memory-efficient framework for high-resolution facial attribute editing," 2020.
[32] W. Chu, Y. Tai, C. Wang, J. Li, F. Huang, and R. Ji, "Sscgan: Facial attribute editing via style skip connections," 2020.
[33] H. Tang, S. Bai, L. Zhang, P. H. Torr, and N. Sebe, "Xinggan for person image generation," in *ECCV*, 2020.
[34] H. Tang, D. Xu, G. Liu, W. Wang, N. Sebe, and Y. Yan, "Cycle in cycle generative adversarial networks for keypoint-guided image generation," in *ACM MM*, 2019.
[35] A. Siarohin, S. Lathuilière, E. Sangineto, and N. Sebe, "Appearance and Pose-Conditioned Human Image Generation using Deformable GANs," *TPAMI*, 2020.
[36] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. Efros, and R. Zhang, "Swapping autoencoder for deep image manipulation," *NeurIPS*, 2020.
[37] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ToG*, 2009.
[38] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.
[39] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *TOG*, 2017.
[40] H. Liu, B. Jiang, Y. Xiao, and C. Yang, "Coherent semantic attention for image inpainting," in *CVPR*, 2019.
[41] H. Zhang, Z. Hu, C. Luo, W. Zuo, and M. Wang, "Semantic image inpainting with progressive generative networks," in *ACM MM*, 2018.
[42] Y. Jo and J. Park, "Sc-fegan: Face editing generative adversarial network with user's sketch and color," in *ICCV*, 2019.
[43] B. Dolhansky and C. Canton Ferrer, "Eye in-painting with exemplar generative adversarial networks," in *CVPR*, 2018.
[44] K. Olszewski, D. Ceylan, J. Xing, J. Echevarria, Z. Chen, W. Chen, and H. Li, "Intuitive, interactive beard and hair synthesis with generative models," *CVPR*, 2020.
[45] M. Banf and V. Blanz, "Example-based rendering of eye movements," in *CGF*, 2009.
[46] D. Kononenko, Y. Ganin, D. Sungatullina, and V. Lempitsky, "Photo-realistic monocular gaze redirection using machine learning," *TPAMI*, 2017.
[47] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, "Few-shot adaptive gaze estimation," in *CVPR*, 2019.
[48] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.
[49] 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 *CVPR*, 2018.
[50] 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*, 2014.
[51] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: passive eye contact detection for human-object interaction," in *ACM symposium on User interface software and technology*, 2013.
[52] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *TPAMI*, 2017.
[53] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," *ECCV*, 2020.
[54] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *CVPR*, 2019.
[55] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in *ECCV*, 2020.
[56] X. Cai, B. Chen, J. Zeng, J. Zhang, Y. Sun, X. Wang, Z. Ji, X. Liu, X. Chen, and S. Shan, "Gaze estimation with an ensemble of four architectures," *arXiv preprint arXiv:2107.01980*, 2021.
[57] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *CVPR*, 2015.
[58] D. E. King, "Dlib-ml: A machine learning toolkit," *JMLR*, 2009.
[59] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *CVPR*, 2020.
[60] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *ICLR*, 2018.
[61] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao, "Relgan: Multi-domain image-to-image translation via relative attributes," in *CVPR*, 2019.
[62] J. gi Kwak, D. K. Han, and H. Ko, "Cafe-gan: Arbitrary face attribute editing with complementary attention feature," 2020.
[63] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2003.
[64] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.
[65] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," in *NeurIPS*, 2017.

Jichao Zhang received the M.S. degree from School of Computer Science and Technology, Shandong University in 2019. He is currently pursing the Phd. degree at the University of Trento. His research interests include deep generative model, neural rendering, 2D/3D image generation and editing.

Jingjing Chen received the B.S. degree from Shandong University, Jinan, China in 2020. She is currently pursuing a master degree at Zhejiang University. Her research interests include computer vision and machine learning.

Hao Tang is currently a Postdoctoral with Computer Vision Lab, ETH Zurich, Switzerland. He received the master's degree from the School of Electronics and Computer Engineering, Peking University, China and the Ph.D. degree from the Multimedia and Human Understanding Group, University of Trento, Italy. He was a visiting scholar in the Department of Engineering Science at the University of Oxford. His research interests are deep learning, machine learning, and their applications to computer vision.

Enver Sangineto is Assistant Professor with the University of Trento, Italy. He received his PhD in Computer Engineering from the University of Rome "La Sapienza". After that he has been a post-doctoral researcher at the Universities of Rome "Roma Tre" and "La Sapienza" and at the Italian Institute of Technology (IIT) in Genova. His research interests include both discriminative and generative methods and learning with minimal human supervision.

Peng Wu entered the school with software of Shandong University in 2018 and the artificial intelligence experimental class in 2019. Now he is a junior in the artificial intelligence experimental class of Shandong University. He is full of interest in the field of computer vision, and has been in this area of learning and exploration.

Yan Yan received the Ph.D. degree in computer science from the University of Trento. He is currently Gladwin Development Chair Assistant Professor with the Department of Computer Science, Illinois Institute of Technology. He was an Assistant Professor at Texas State University and a Research Fellow at the University of Michigan and the University of Trento. His research interests include computer vision, machine learning, and multimedia.

Nicu Sebe is Professor with the University of Trento, Italy, leading the research in the areas of multimedia information retrieval and human behavior understanding. He was the General Co- Chair of ACM Multimedia 2013, and the Program Chair of ACM Multimedia 2007 and 2011, ECCV 2016, ICCV 2017 and ICPR 2020. He is a Co-Chair of ACM Multimedia 2022 and a Program Chair of ECCV 2024. He is a fellow of the International Association for Pattern Recognition.

Wei Wang is currently an Assistant Professor in the Department of Information Engineering and Computer Science at University of Trento, Italy in which he received his Ph.D Degree. He was a Postdoctoral Research Fellow in CVLab at EPFL. His research interests include face analysis, human action understanding, augmented reality (AR), segmentation, optimization problems, etc.