GazeFlow: Gaze Redirection with Normalizing Flows

Yong Wu§, Hanbang Liang§, Xianxu Hou, Linlin Shen*
Computer Vision Institute, College of Computer Science and Software Engineering.
Shenzhen University
{wuyong19, lianghanbang2019}@email.szu.edu.cn, hxianxu@gmail.com, llshen@szu.edu.cn

***Abstract*—Gaze estimation often requires a large scale datasets with well annotated gaze information to train the estimator. However, such a dataset requires costive annotation and is usually very difficult to collect. Therefore, a number of gaze redirection approaches have been proposed to address such a problem. However, existing methods lack the ability to precisely synthesize images with target gaze and head pose in complex lighting scenes. As a powerful technique to model the distribution of given data, normalizing flows have the ability to generate photo-realistic images and provide flexible latent space manipulation. In this work, we present a novel flow-based generative model, GazeFlow [1], for gaze redirection. The visual results of gaze redirection show that the quality of eye images synthesized by GazeFlow is significantly higher than that of other approaches like DeepWarp and PRGAN. Our approach has also been applied to augment the training data to improve the accuracy of gaze estimators and significant improvement has been achieved for both within dataset and cross dataset experiments.**

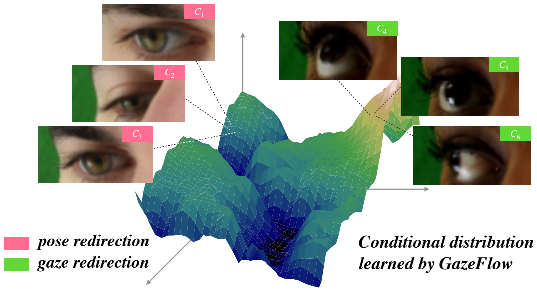

Fig. 1. An example of conditional distribution of eye images. c_i denotes different head poses and gazes. Our model GazeFlow learns the conditional distribution over training dataset, in which we can sample images with specific conditions. Neighboring data points in the distribution share similar characteristics.

I. Introduction

Gaze estimation is a task to predict the direction or the point where a person is looking at, which is an effective attention tool to express the interactions between humans and objects. Eye detection and gaze estimation have become useful tool for Human-Computer Interaction [1], [2] and AR/VR [3], [4].

In recent years, appearance-based gaze estimation has become one of the most popular methods, which use face or eye images to predict gaze directions. Deep learning methods have become a powerful tool for gaze estimation [5]–[8]. However, their performance often relies on a large amount of annotated data. Although it is easy to obtain face images nowadays, it is still very difficult to obtain face or eye images with labeled gaze directions. One possible solution is to use gaze redirection technique to synthesize images to augment existing datasets.

Training models on synthetic images has become a popular way for gaze estimation recently. Traditional methods for synthesizing eye images usually re-render the eye region with a 3D model and modify their gaze directions [9], [10]. 3D cameras are needed to scan human face and eye region. With the development of deep learning, learning-based gaze redirection models [11]–[13] have also been developed. For instance, DeepWarp [11] uses a neural network to predict the warping field of the input images. The network is trained on pairs of images collected from the same person with fixed head pose and lighting conditions. When these conditions change, they would fail to generate photo-realistic eye images. He *et al.* [12] introduce Generative Adversarial Network (GAN) [14] to gaze redirection. They extend the work [5] and use paired data to train a network, PRGAN, to synthesize photo-realistic eye images conditioned on a target gaze direction. Recently, Xia *et al.* present interpGaze [13] for controllable gaze redirection and interpolation. All the above methods require a stable laboratory environment to obtain images and annotations. Usually, they collect datasets by themselves under certain circumstances. Zheng *et al.* [15] propose a novel self-transforming encoder-decoder architecture to generate face images with high-fidelity gazes and head orientations. Shrivastava *et al.* [16] use unlabeled real images to refine synthetic images through GAN, and make a large improvement for gaze estimation. Wang *et al.* [17] propose Hierarchical Generative Model (HGM), to synthesizes eye images with the given gaze in a top-down inference and estimate gaze from an eye image through bottom-up inference. However, GAN-based models are often unstable, due to multiple loss functions and adversarial training.

Existing methods are mostly based on GAN and Autoencoder based architecture, which normally requires paired data to calculate pixel-wise reconstruction loss. To relax the requirement of paired data, we propose a novel flow-

* Corresponding Author.
§ Equal contribution.
[1] The code will be made available at https://github.com/CVI-SZU/GazeFlow

TABLE I

Comparing our work against similar methods. (1) PR represents photo-realistic images. (2) w/o EI indicate whether external information, such as external devices and external landmarks, is required. (3) w/o Paired means whether paired data is required during training or inference.

Main Idea	Methods	PR	w/o EI	w/o Paired
Graphics	UT Multiview [9]	✓	✗	✓
	UnityEyes [10]	✗	✓	✓
CNN	DeepWarp [11]	✓	✗	✗
GAN	Shrivastava *et al.* [16]	✓	✓	✓
	HGM [17]	✓	✗	✓
	PRGAN [12]	✓	✓	✗
	interpGaze [13]	✓	✓	✗
	STED-Gaze [15]	✓	✓	✗
Normlizing Flows	Ours	✓	✓	✓

based model, called GazeFlow, for gaze redirection. Existing methods can not well disentangle the head pose from eye gaze image. By contrast, our method can separately edit gaze and head pose of given eye images, as shown in Figure 1. Without the requirement of paired data, our approach can synthesize high quality eye directions of targeting angle in complex scenes. Table I lists the key differences between our approaches and other methods, including graphics based approaches [9], [10] and GAN based methods [12], [13], [16], [17]. As shown in the table, our approach have no requirement of the controlled environment, external devices and paired data. Based on the high quality of redirection, our approach can be used to generate a large number of eye images with known gazes and improve the performance of existing gaze estimators.

To summarize, our contribution in this paper is three-fold:

- We present a flow-based eye image generation model, GazeFlow, for gaze redirection and estimation. As the first approach using normalizing flows for this task, GazeFlow does not require paired data collected from controlled environments for training and inference.
- Our method can disentangle the gaze and head pose, and can separately modify yaw and pitch of either gaze or head pose.
- Our method is able to generate images with continuous annotations of gazes and head poses, which can be directly used to augment data to improve the performance of the existing gaze estimation models.

II. Related works

A. Gaze Redirection

Gaze redirection is a task to redirect eye images to the target gaze specified by angle labels. Warping-based methods [9], [11] take an eye image as input, and warp the input image with a desired output appearance. As Generative Adversarial Network (GAN) has been successfully used in many image generation tasks [19], [20], He *et al.* [12], Xia *et al.* [13] and Zheng *et al.* [15] use GAN to generate photo-realistic eye images. However, the multiple loss functions make GAN's training unstable. Generally, GAN based methods require the eyes images captured with fixed head poses for training, and their performances are not robust against environment and lighting variances. For example, to redirect the gaze of an input eye image, PRGAN [12] not only require the target gaze label, but also the paired eye image from the same subject, which present the target gaze with the same head pose captured under similar lighting condition. Wang *et al.* [17] propose a GAN-based two-way model to produce eye images or obtain gaze direction. Yu *et al* [21] propose a self-supervised method to learn a low dimensional gaze representation without gaze annotation.

B. Eye Image Synthesis for Gaze Estimation

Due to the lack of large datasets with precise gaze annotations, training a gaze estimation model from synthetic eye images has become popular in recent years. Earlier works tried to render a scene containing the face or eye region of a certain subject [22]. These methods often require a depth map of the face and use 3D transformations to synthesize image with new gaze directions. Sugano *et al.* [9] propose UT Multiview dataset with different views. They use eye images collected from real world to perform a 3D reconstruction in order to synthesize eye images. Wood *et al.* [10], [23], [24] use 3D morphable models to fit texture and shape of the eye, and re-render the synthesized eyeballs to the source image. Based on the gaze data generated in simulated environmental settings, they propose UnityEyes synthesis framework to render eye images and use these images to train a CNN [23] or kNN [10] for better results. However, there exists a large domain gap between synthesized images and real ones, Shrivastava *et al.* [16] refine the output of the simulator with a refining neural network to reduce such a domain gap.

C. Normalizing Flows

As a powerful density estimation tool, normalizing flows [25] have been widely used in various computer vision tasks such as image generation [18], [26], [27] and super-resolution [28]. Since normalizing flows possess distinctive characteristics such as invertibility and exact likelihood estimation, they can be used to model the distribution over the given datasets in order to generate more samples from the distribution. Compared to other generative models, such as Variational Auto-encoder VAE [29] and GAN [14], normalizing flows can invert real image back to the distribution, retrieve unique latent code according to Equation (1) and allow exact log-likelihood based training. In terms of generative modeling, normalizing flows have been proven to be strong enough to generate photo-realistic images [27], [30], [31]. Additionally, there exist different types of normalizing flows [26], [31]–[34], which achieve good image generation performances. To enhance training dataset for gaze estimation and gaze redirection, instead of only generating eye images, accurate gaze and head pose labels are also needed. We turn the gaze redirection task into eye image synthesis conditioned on given directions, which is exactly what conditional normalizing

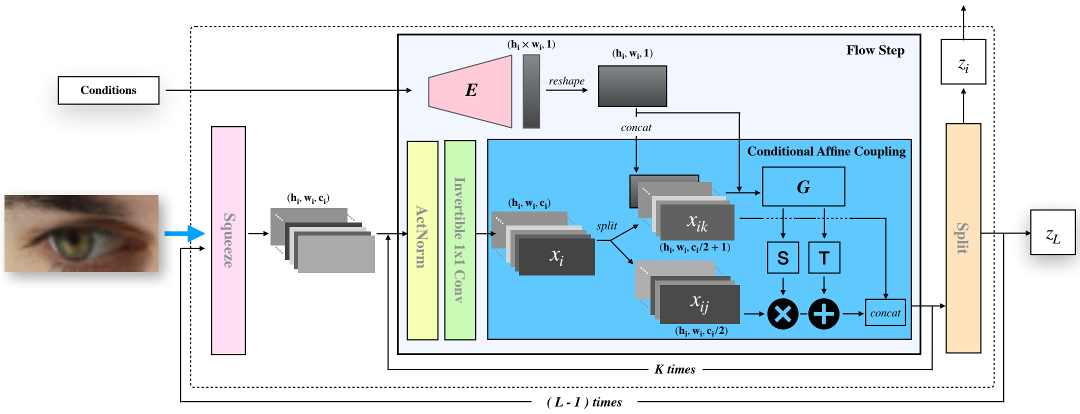

Fig. 2. Overview of our GazeFlow in forward mode. We mainly modify the affine coupling layers in Glow with our proposed conditional affine coupling. The architecture of Glow consists of K Flow Steps and L levels of multi-scale layers proposed in RealNVP [18]. The arrow in blue represents the input image fed into the model. Condition encoder $\boldsymbol{E}$ processes the condition and reshape it into shape (h_i, w_i) in current flow step. Encoded condition is then fed into a neural network $\boldsymbol{G}$ with x_{ik} (one half of x_i). Scale S and shift T then transform x_{ij} (the other half of x_i).

flows are designed for. By training a conditional distribution $p_{\mathbf{x}|\mathbf{c}}(\mathbf{x} \mid \mathbf{c}, \boldsymbol{\theta})$, we can sample images with given labels $\mathbf{c}$, which we will call conditions in the rest of the paper.

III. Methodology

A. Overview

To construct a latent space containing rich semantic information, we replace normal Autoencoder with invertible neural network and embed the images with conditions into latent codes. Since the model is invertible, we do not need any pixel-wise reconstruction objectives. Instead, our model combine invertible neural network and maximum log-likelihood training. We pick Glow [27] as our architecture's backbone. Instead of using affine coupling layer in Glow, we introduce conditional affine coupling with an auxiliary condition encoder. To take advantage of the invertibility of normalizing flows, forward mode is used to represent the mapping from image space to latent space, while inverse mode is used to represent the mapping from latent space to image space. Note that no matter forward mode or inverse mode, the same model is used. The overview of our GazeFlow in forward mode is shown in Figure 2.

B. Conditional Normalizing Flows

Normalizing flows require a bijective mapping $f : R^d \rightarrow R^d$ such that there exists an inverted mapping h where $h := f^{-1}$. By applying change-of-variables, normalizing flows map an arbitrary data from training set to another data $\mathbf{z}$ from a simple base distribution $p_{\mathbf{z}}(\mathbf{z})$ (e.g. standard normal distribution). The target distribution $p_{\mathbf{x}}(\mathbf{x})$ can be explicitly computed as,

$$p_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\theta}) = p_{\mathbf{z}}\left(f_{\boldsymbol{\theta}}(\mathbf{x})\right) \left| \det \frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}} \right| \tag{1}$$

where θ represents the parameters of bijective mapping f. In order to condition every input data point $\mathbf{x}$ on its labels, we require the bijective mapping to take an extra input as conditions $\mathbf{c}$: $f(\mathbf{x}, \mathbf{c}) = \mathbf{z}$ such that $h(\mathbf{z}, \mathbf{c}) = \mathbf{x}$. Rewriting Equation (1) into a conditional normalizing flow form, we have

$$p_{\mathbf{x}|\mathbf{c}}(\mathbf{x} \mid \mathbf{c}; \boldsymbol{\theta}) = p_{\mathbf{z}}\left(f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{c})\right) \left| \det \frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{c})}{\partial \mathbf{x}} \right| \tag{2}$$

Note that f can be a chain of bijective mapping such as $f_\theta = f_i(f_{i-1}(...f_1(\mathbf{x}, \mathbf{c})))$. Equation (2) allows us to train the model $p_{\mathbf{x}|\mathbf{c}}(\mathbf{x}, \mathbf{c})$ by minimizing the negative log-likelihood (NLL) over training data $\mathbf{x}$. By applying multiple bijective mappings into the model, we have our objective as:

$$\begin{aligned} \mathcal{L} &= -\log p_{\mathbf{x}|\mathbf{c}}(\mathbf{x} \mid \mathbf{c}; \boldsymbol{\theta}) \\ &= \log p_z(f_\theta(\mathbf{x}; \mathbf{c})) + \sum_{i=1}^{N} \log \left| \det \frac{\partial f_i}{\partial f_{i-1}} \right| \end{aligned} \tag{3}$$

where N represents the number of bijective mappings in the model. Note that each flow layer f_i is required to calculate a Jacobian determinant term with respect to the input. Therefore, invertibility and low computational cost for Jacobian determinant are the two keys to design a flow layer.

C. GazeFlow

To compose invertible and well conditioned normalizing flows with tractable Jacobian determinant, we build our model GazeFlow upon unconditional Glow architecture with modified affine coupling. The overview of GazeFlow is depicted in Figure 2 and the details are introduced briefly as following, along with our proposed condition affine coupling and condition encoder in this section.

1) Squeeze Layer: Introduced in [18], the layer provides an inervitble mapping to adjust spatial sizes and number of channels. The layer transformed a $H \times W \times C$ tensor into a $\frac{H}{2} \times \frac{W}{2} \times 4C$ tensor.

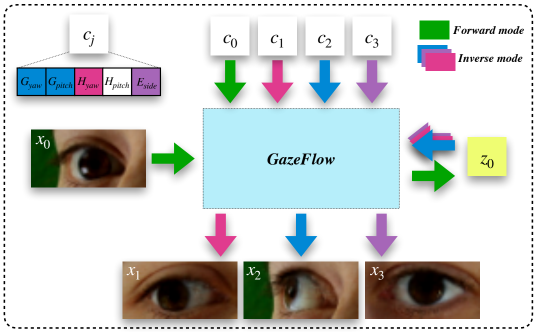

Fig. 3. An illustration of gaze redirection by editing conditions described by Equation (5). Arrow in green represents image encoding with its own condition in forward mode. Arrows with red, blue and purple represent the decoding process for conditional changes in H_{yaw}, (G_{yaw}, G_{pitch}) and E_{side}.

2) Flow Steps: As the core part of our architecture to transform the base distribution into the target distribution, our flow steps include an Activation Normalization (ActNorm) layer [27], an invertible 1×1 convolution layer [27] and a conditional affine coupling layer. ActNorm is originally proposed to apply channel-wise normalization through a learnable scale and bias. The goal of the invertible 1×1 convolution is to learn a permutation for channels shuffling, since only half of channels are transformed. Unlike the one purposed by Lu [35], our Invertible 1×1 convolution layers and ActNorm layers are set to unconditional for stable training.

3) Conditional Affine Coupling: Our proposed layer is a type of conditional normalizing flows. Affine Coupling [18] is a simple but effective bijective transformation for constructing normalizing flow layer with low computational cost and invertibility. It splits input into two parts, $\mathbf{x_1}$ and $\mathbf{x_2}$, with the same size along channels. $\mathbf{x_1}$ and encoded conditions are then passed into a neural network, which will output a scale and shift to transform $\mathbf{x_2}$ into $\mathbf{z_2}$. $\mathbf{x_1}$ remains unchanged and is concatenated with $\mathbf{z_2}$ as output $\mathbf{z}$ of this layer. By adding conditions into affine coupling, we extend affine coupling to conditional affine coupling, shown in Figure 2. The extension does not introduce extra computational cost to Jacobian determinant calculation since Jacobian matrix in Equation (4) remains triangular:

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{x}_1 \\ \mathbf{z}_2 &= \exp\left(\boldsymbol{G}_{\boldsymbol{\theta},\mathbf{S}}(\mathbf{x}_1, \mathbf{g}(\mathbf{c}))\right) \cdot \mathbf{x}_2 + \boldsymbol{G}_{\boldsymbol{\theta},\mathbf{T}}\left(\mathbf{x}_1, \mathbf{g}(\mathbf{c})\right) \\ \mathbf{z} &= concat(\mathbf{z}_1, \mathbf{z}_2) \end{aligned} \tag{4}$$

where $\boldsymbol{G}_{\boldsymbol{\theta}}$ is an arbitrary neural network, $\mathbf{g}$ is a condition encoder to turn condition into a factor of transformation for $\mathbf{x_2}$. Since $\mathbf{x_1}$ and $\mathbf{c}$ remain unchanged, $\mathbf{x_2}$ can be easily inverted. The Jacobian term is exponent of $G_{\boldsymbol{\theta},\mathbf{S}}(\mathbf{x}_1, \mathbf{g}(\mathbf{c}))$.

4) Condition Encoder: It is a part of our proposed conditional afffine coupling. The conditions, gaze and head pose, are discrete with respect to each image since limited images in training set can not cover all gaze and head pose direction. Therefore, in order to perform continuous gaze redirection, we need to map gaze and head pose of provided images into continuous space with our proposed condition encoder. With the condition encoder, conditions are mapped into a continuous condition feature distribution to extract semantic information for disentanglement. The condition encoder consists of fully connected layers and a reshape layer which reshapes the output of fully connected layers, such that it can be concatenated with x_1 described in Equation (4). Having one condition encoder for each conditional affine coupling layer improves the disentanglement of gaze and head pose of given eye images.

Algorithm 1: Condition Sampling Based Eye Image Synthesis

```
Input      : A set of training images with gaze
             annotation
             $X = \{(x_1, c_1), (x_2, c_2), \cdots, (x_N, c_N)\}$;
             GazeFlow inverse mode $F^{-1}$ ;
             Latent code $z_i$ sampled from $p_z(z)$ or
             encoded from a given eye image $x_i$
Output     : A set of synthesized images
             $Y = \{(y_1, c_{k_1}), (y_2, c_{k_2}), \cdots, (y_M, c_{k_M})\}$
Initialize: Randomly sample $M$ conditions
             $\{c_{k_1}, c_{k_2}, \cdots, c_{k_m}, \cdots, c_{k_M}\}$ from
             $\{c_1, c_2, \cdots, c_m, \cdots, c_N\}$;
             Set $Y = \{\}$
for $m = 1$ to $M$ do
    Sample random noise $\varepsilon_m$ from $\mathcal{N}(\mathbf{0}, \mathbf{1})$
    $c_{k_m} = c_{k_m} + \varepsilon_m$
    Generate $y_m = F^{-1}(z_i, c_{k_m})$
    Add $(y_m, c_{k_m})$ into Y
end
Return: $Y$
```

With layers described above, GazeFlow can transform the simple base distribution into the complex eye image distribution. Base distribution is also considered to be the latent space of GazeFlow where images are encoded into latent codes.

We perform gaze redirection with GazeFlow by condition manipulation. As shown in Figure 3, the condition in GazeFlow is a 1×5 vector, $[G_{yaw}, G_{pitch}, H_{yaw}, H_{pitch}, E_{side}]$, which represent the gaze in yaw and pitch direction, the head pose in yaw and pitch direction and the eye side (left/right). We firstly introduce how to redirect the gaze of a given eye image using condition manipulation. A condition sampling based approach is then proposed to synthesize a set of eye images for data augmentation.

D. Condition Manipulation

An eye image x_0 can be retrieved from latent code z_0 by feeding it along with its original condition c_0 back to model in inverse mode according to Equation (2). Similarly, we encode eye image to latent code with its own condition in forward mode F. To generate image x_j with new gaze, we pass the latent code with new condition back to the model in inverse

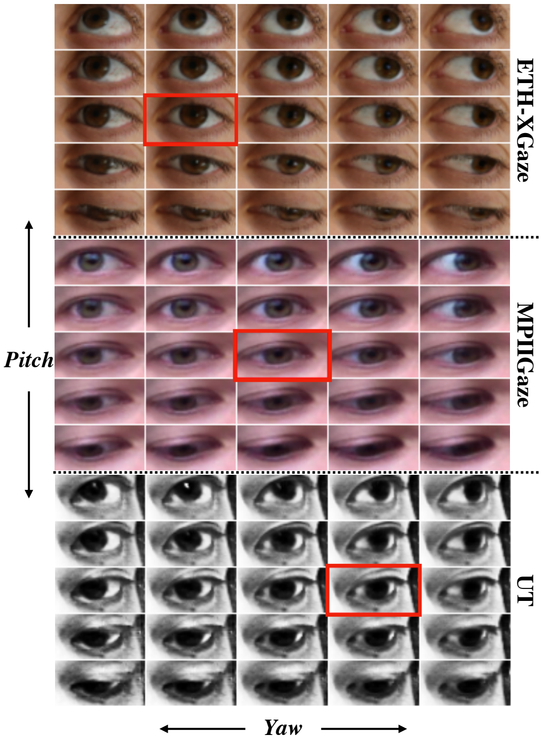

Fig. 4. Gaze redirection results of the GazeFlow on ETH-XGaze, MPIIGaze and UT. The original images are circled in red. Pitch and yaw are changed horizontally and vertically.

mode F^{-1}:

$$\begin{aligned} c_0 &= \left[G^0_{yaw} \, , \, G^0_{pitch} \, , \, H^0_{yaw} \, , \, H^0_{pitch} \, , \, E^0_{side} \right] \\ z_0 &= F(x_0, c_0) \\ c_j &= \left[G^j_{yaw} \, , \, G^j_{pitch} \, , \, H^j_{yaw} \, , \, H^j_{pitch} \, , \, E^j_{side} \right] \\ x_j &= F^{-1}(z_0, c_j) \end{aligned} \tag{5}$$

An eye image x_0 with its original gaze c_0 can thus be changed to x_j with new gaze c_j. Besides, image x_j preserves its original characteristic and details since latent code z_0 is fixed. Figure 3 illustrates this process. By using new conditions in inverse mode, one can perform not only gaze redirection, but also head pose redirection and eye flipping. For example, the blue arrows in figure denotes the process of gaze changes in both pitch and yaw, by feeding the condition c_2 for decoding.

E. Condition Sampling Based Eye Image Synthesis.

While eye images with new gazes can be synthesized for a given eye image by passing a new condition to our GazeFlow model, we could apply it to augment data to improve the performance of gaze estimator as well. As the labelling cost of eye gazes is generally expensive, such an augmentation could substantially increase the number of eye images with known gazes and thus increase the accuracy of existing gaze estimators.

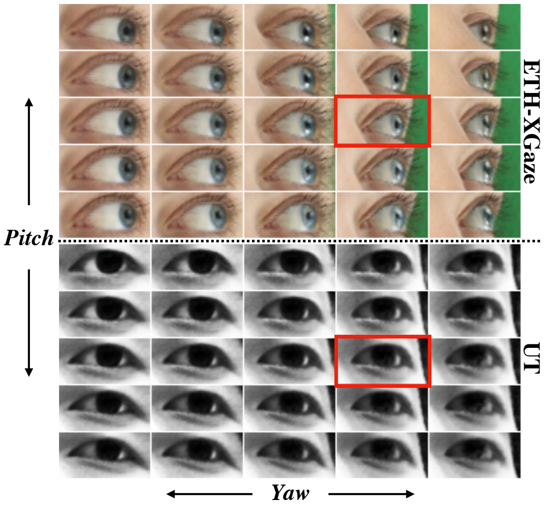

Fig. 5. Head pose redirection on ETH-XGaze and UT. The original images are circled in red. Head poses (pitch and yaw) are changed in the same step.

Given an eye image dataset with limited number of N labeled training images $\{(x_1, c_1), (x_2, c_2), \cdots, (x_i, c_i), \cdots, (x_N, c_N)\}$, where c_i record the pitches and yaws of eye gaze for image x_i, we propose a condition sampling based algorithm to synthesize a set of M eye images $Y = \{(y_1, c_{y_1}), (y_2, c_{y_2}), \cdots, (y_m, c_{y_m}), \cdots, (y_M, c_{y_M})\}$ based on image x_i. While any reasonable condition c_j could be applied to generate a redirected gaze for eye image x_i, such a c_j might not follow the distribution of c_i for available training images and produce outliers, which might not benefit the training of gaze estimators. To overcome such an issue, we sample M conditions $\{c_{k_1}, c_{k_2}, \cdots, c_{k_m}, \cdots, c_{k_M}\}$ from the set of $\{c_1, \cdots, c_m, \cdots, c_N\}$ and use them as the new conditions for eye image synthesis. To increase the diversity of conditions, a small noise ε sampled from a standard normal distribution could be added to introduce some randomness to the sampled conditions. Experiments show that adding a reasonable small noise can benefit the performance of gaze estimators. Note that we only modify the yaw and pitch of gaze and head pose (G_{yaw}, G_{pitch}, H_{yaw}, H_{pitch}) here. While the latent code z_i of existing image x_i could be used to preserve the identity and characteristics of the corresponding person, the base distribution $p_z(z)$ could also be used to generate the latent code. The details of our algorithms are listed in Algorithm 1.

IV. Experiment

We first evaluate the performance of our gaze redirection approach by visually evaluating the quality of eye images synthesized for different gazes, then present the results of existing gaze estimators when our approach is used to generate the eye images with known labels for data augmentation.

A. Datasets

a) ETH-XGaze: ETH-XGaze [36] is a recently proposed gaze dataset, which consists of over one million high-resolution face images with varying gazes captured under

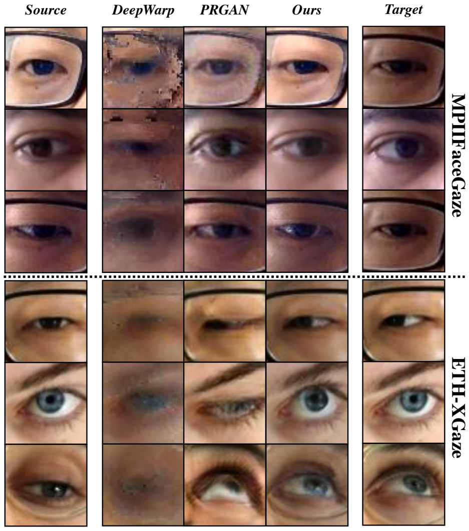

Fig. 6. Gaze redirection of different methods. The results on MPIIFaceGaze and ETH-XGaze are shown above and below the dotted line respectively.

different head poses. The images are captured from 110 participants using 18 cameras in different positions. Over two million eye images with annotated gazes could be cropped out from the available face images.

b) MPIIGaze & MPIIFaceGaze: MPIIGaze [37] is a large gaze dataset collected from laptop cameras in everyday use, with complex illumination variations. The dataset contains more than 400k single eye images captured from 15 subjects. While the whole set of images can be used for testing, a smaller test set of 45k images (3k/person) is also proposed in [37] for leave-one-person-out experiments.

MPIIFaceGaze contains the full face images for part of the eye images in MPIIGaze. While about 35k face images are available, 70k eye images with annotated gazes could be cropped out from the dataset.

c) UT Multiview: UT Multiview [9] contains three different sets, i.e., raw, test and synthetic sets. 128k preprocessed eye images captured from different subjects with various head poses and gazes are provided in the test set. The 2,304k eye images in synthetic set are synthesized using the 3D model provided in the raw set. For simplicity, UT Multiview is referred as UT in the following sections.

B. Implementation Details

For images with size $32 \times 64 \times 3$, we set the number of flow steps $K = 18$ and multi-scale level $L = 3$. On the other hand, for images with size $64 \times 64 \times 3$, the number of flow steps and multi-scale level are set to $K = 36$ and $L = 4$, respectively. We use two fully-connected layers with SoftPlus activation function [38] as our condition encoder. We simply use negative log likelihood shown in Equation (3) as the training objective.

C. Condition Manipulation Based Gaze Redirection

We now visually show the examples of gaze redirection performed by our GazeFlow model on ETH-XGaze, MPIIGaze and UT datasets. All of the images are resized to 32×64 and in this experiment, the number of images used to train our GazeFlow model for ETH-XGaze, MPIIGaze and UT are 500k, 400k and 120k, respectively. Figure 4 shows the gaze redirection results of our approach for three example eye images (circled in red) from the three datasets. Starting from the yaw and pitch of the example eye image, we interpolate five yaw and pitch angles with a fixed step size and use them as the new condition to generate the eye images with redirected gazes using Equation (5). As shown in the figure, our GazeFlow can generate vivid eye images with diverse gaze yaws and pitches.

TABLE II
THE SIMILARITY BETWEEN EYE IMAGES REDIRECTED BY DIFFERENT METHODS AND TARGET IMAGES IN MPIIFACEGAZE.

Method	**Deepwarp**	**PRGAN**	**GazeFlow**
Similarity	0.511	0.719	**0.927**

Since ETH-XGaze and UT contains relatively large angles of head pose, we perform head pose redirection on these two datasets. Figure 5 shows head pose redirection results of GazeFlow for an example eye image (circled in red) from each dataset. Similarly, the interpolated head pose yaws and pitches are passed to GazeFlow to generate the eye images with redirected head poses. As the images in ETH-XGaze dataset are collected in a laboratory environment with green screen background, green background also appears in the synthesized eye images for head poses with big yaw angles. More gaze and head redirection results are available in the supplemental materials. [2]

We now compare the gaze redirection performance of our approach with that of DeepWarp [11] and PRGAN [12], using MPIIFazeGaze and ETH-XGaze datasets. As both DeepWarp and PRGAN require paired eye images for training, we randomly selected 70k and 500k pairs of eye images from MPIIFazeGaze and ETH-XGaze datasets, respectively, and use these images to train three models. A pair of eye images is captured from the eye of the same subject with the similar head pose and different gazes.

Figure 6 shows the redirection results of the three models for six example eye images (not available in the training set) from the two datasets. While reference eye image with target gazes (shown in the final column) are required by DeepWarp and PRGAN, our approach actually only requires the pitch and yaw of target gazes and head poses. As shown in the 2^{nd} column, the eye images generated by DeepWarp are not clear and contain lots of artifacts. While the eye images generated by PRGAN for MPIIFaceGaze do not well preserve the identity, those images generated for ETH-XGaze do not present gazes similar with the target images. Moreover, DeepWarp and PRGAN both have trouble generating realistic images since they are not robust against extreme lighting and head poses. The eye images generated by GazeFlow can well

[2] A demo in Colab is available: https://bit.ly/3r6W4yl

TABLE III
PERFORMANCE OF OUR AUGMENTATION ON MPIIGAZE AND UT DATASETS.

Protocol	Test Data	Training Data	Amount	mAE
Within Dataset	MPIIGaze test set (one subject) leave-one-person-out	MPIIGaze (14 subjects)	42k	5.8
		Synthesized by GazeFlow (14 subjects)	42k	6.4
			126k	6.1
			378k	**5.6**
	UT (test set)	UT Synthetic Set	64k	6.8
			128k	6.2
			256k	5.9
		Synthesized by GazeFlow	64k	6.4
			128k	5.6
			256k	**5.3**
Cross Dataset	MPIIGaze test set (all 15 subjects)	UT Synthetic Set	64k	16.1
			128k	17.1
			256k	16.9
		Synthesized by GazeFlow	64k	16.2
			128k	15.4
			256k	**14.9**

preserve the identity and present precise gaze redirection at the same time.

Now we randomly select 100 pairs of eye images to quantitatively evaluate the redirect performance of DeepWarp, PRGAN and our GazeFlow. While both target eye image and target gaze are required as the input of DeepWarp and PRGAN, only target gaze label is needed by GazeFlow. As the pair of target and input image is available, we use Preact-ResNet-8 [39] pretrained using MPIIGaze training set to predict gaze from both target and redirected eye images, and measure their similarities using cosine distance. As shown in Table II, the eye images generated by GazeFlow (0.927) is much more similar with the target images than that of PRGAN (0.719) and DeepWarp (0.511).

D. Eye Image Synthesis Based Data Augmentation for Gaze Estimation

We now use the *Condition Sampling Based Eye Image Synthesis* algorithm described in Algorithm (1) to augment training data for gaze estimation. The Preact-ResNet-8 [39] is used here as the gaze estimator. MPIIGaze [37] and UT [9] datasets are adopted here for evaluation. mAE (mean Angular Error), designed to compute angular difference between the estimated gaze with ground truth in 3D space, is used as the accuracy indicator.

We first evaluate the performance of our augmentation using the leave-one-person-out test set of MPIIGaze, which consists of 45k images captured from 15 subjects. Our GazeFlow is trained using $\sim$360k eye images (45k images in the test set are excluded) in MPIIGaze. We synthesize redirected eye images and labels using GazeFlow, which are resized to 36×60 to train the gaze estimation model. For each of the 14 subjects in training folders, we sample his conditions (gazes and head poses) and generate 3k/ 9k/ 27k eye images, to train the estimator. As shown in Table III, the performance of estimator generally improve with the number of eye images synthesized by GazeFlow, and achieve 5.6 mAE when the number is 378k, which is about 0.2 lower than that of estimator trained using the 42k images in the original training folders.

TABLE IV
GAZE ESTIMATION PERFORMANCE ON MPIIGAZE FOR DIFFERENT DATA AUGMENTATION APPROACHES. R/S REPRESENTS THAT WHETHER THESE METHODS USE REAL(R) OR SYNTHETIC(S) DATA.

Methods	Amount	R/S	mAE
Zhang *et al.* [37]	128k	R	13.9
Zhang *et al.* [7]	128k	R	11.7
Wood *et al.* [23]	12k	S	13.6
Wood *et al.* [23]	140k	S+R	11.1
Wood *et al.* [10]	-	S	10.0
Shrivastava *et al.* [16]	-	S	11.2
Shrivastava *et al.* [16]	-	S+R	7.8
HGM [17]	-	S+R	7.7
Ours	45k	S	5.4
Ours	405k	S	**5.0**

Now we evaluate the performance of our augmentation using UT test set, which consists of 128k images with gaze annotations. The eye images of synthetic set in UT dataset are synthesized using a reconstructed 3D eye model with patch-based multi-view stereo algorithm [40] and Poisson reconstruction method [41]. The synthesis process requires UT test set and its 3D eye model. Our GazeFlow model is directly trained using only the images in UT test set. A set of new synthesized images are then generated using our GazeFlow model. We choose UT test set (128k images) to evaluate the performance of gaze estimation. The Preact-ResNet-8 based gaze estimator is trained using the two synthetic sets and evaluated on the test set for mAE assessment.

As shown in Table III, again the mAE of the estimator decreases with the number of training images for both synthesis approaches. For all of the different numbers of synthesized images, the mAE of GazeFlow is consistently lower than that of UT 3D model based method. The lowest mAE of 5.3 is achieved when 256k eye images synthesized by our GazeFlow are used to train the gaze estimator. These results demonstrate that images synthesized from GazeFlow are consistent with the distribution of the augmented dataset.

To test the generalization capability of the eye images synthesized by our GazeFlow model, we now perform a cross-dataset evaluation by training the estimator using images synthesized for UT and testing its performance on the full test set of MPIIGaze. As shown in Table III, the performance of estimator actually drops with the increase in number of images from UT synthetic set used for training. It seems that there is a dominant gap between the two image domains. While the performance of estimator consistently improves with the increase of number of images synthesized by our GazeFlow, the lowest mAE (14.9) is achieved when 256k synthesized images are used to train the gaze estimator. However, the mAE is significantly higher than that (5.6) trained using the images synthesized for MPIIGaze dataset. Such an image domain gap needs to be further investigated in the future.

Table IV shows the performance of our approach against state of the art methods available in literature, when MPIIGaze dataset is used for testing. To follow the same testing protocol, the whole test set of 45k images are evaluated and the overall accuracy is reported. In addition to UT, SynthesEyes [23] and UnityEyes [10] are widely used datasets for data augmentation [10], [16], [17], [23]. However, the synthesized eye images in SynthesEyes and UnityEyes are not photo-realistic, and there is a large gap between the synthesized images and real images. As shown in the table, our GazeFlow achieves the lowest mAE when 405k images synthesized by our GazeFlow are used to train the gaze estimator. Our mAE (5.0) is significantly lower (-2.7) than that (7.7) of runner up, i.e., HGM [17], which use both UT and UnityEyes for training.

V. Conclusion

To summarize, we proposed a novel normalizing flows based method, GazeFlow, to synthesize eye images with given gazes and head poses. The visual results of gaze redirection show that the quality of eye images synthesized by GazeFlow is significantly higher than that of other approaches like DeepWarp and PRGAN. Our approach has also been applied to augment the training data to improve the accuracy of gaze estimators and significant improvement has been achieved for both within dataset and cross dataset experiments.

Acknowledgement

The work is supported by National Natural Science Foundation of China (91959108 and U1713214).

References

[1] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction," in *Advances in physiological computing*. Springer, 2014, pp. 39–65.

[2] L. Fridman, B. Reimer, B. Mehler, and W. T. Freeman, "Cognitive load estimation in the wild," in *CHI*, 2018, pp. 1–9.

[3] B. I. Outram, Y. S. Pai, T. Person, K. Minamizawa, and K. Kunze, "Anyorbit: Orbital navigation in virtual environments with eye-tracking," in *ETRA*, 2018, pp. 1–5.

[4] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards foveated rendering for gaze-tracked virtual reality," *TOG*, vol. 35, no. 6, p. 179, 2016.

[5] W. Zhu and H. Deng, "Monocular free-head 3d gaze tracking with deep learning and geometry constraints," in *ICCV*, Oct 2017.

[6] Y. Cheng, F. Lu, and X. Zhang, "Appearance-based gaze estimation via evaluation-guided asymmetric regression," in *ECCV*, September 2018.

[7] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE TPAMI*, vol. 41, no. 1, pp. 162–175, 2017.

[8] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, and J. Kautz, "Few-shot adaptive gaze estimation," in *ICCV*, 2019, pp. 9368–9377.

[9] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3d gaze estimation," in *CVPR*, 2014, pp. 1821–1828.

[10] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," in *ETRA*, 2016, pp. 131–138.

[11] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky, "Deepwarp: Photorealistic image resynthesis for gaze manipulation," in *ECCV*. Springer, 2016, pp. 311–326.

[12] Z. He, A. Spurr, X. Zhang, and O. Hilliges, "Photo-realistic monocular gaze redirection using generative adversarial networks," in *ICCV*, 2019, pp. 6932–6941.

[13] W. Xia, Y. Yang, J.-H. Xue, and W. Feng, "Controllable continuous gaze redirection," in *ACM MM*, 2020, pp. 1782–1790.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014, pp. 2672–2680.

[15] Y. Zheng, S. Park, X. Zhang, S. De Mello, and O. Hilliges, "Self-learning transformations for improving gaze and head redirection," *NIPS*, vol. 33, 2020.

[16] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *CVPR*, July 2017.

[17] K. Wang, R. Zhao, and Q. Ji, "A hierarchical generative model for eye image synthesis and eye gaze estimation," in *CVPR*, 2018, pp. 440–448.

[18] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *ICLR*. OpenReview.net, 2017.

[19] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *CVPR*, 2020.

[21] Y. Yu and J.-M. Odobez, "Unsupervised representation learning for gaze estimation," in *CVPR*, 2020, pp. 7314–7324.

[22] C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross, "Gaze correction for home video conferencing," *TOG*, vol. 31, no. 6, pp. 1–6, 2012.

[23] E. Wood, T. Baltrušaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling, "Rendering of eyes for eye-shape registration and gaze estimation," in *ICCV*, December 2015.

[24] 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, no. 2. Wiley Online Library, 2018, pp. 217–225.

[25] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *ICML*, 2015, pp. 1530–1538.

[26] L. Dinh, D. Krueger, and Y. Bengio, "NICE: non-linear independent components estimation," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[27] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NIPS*, 2018, pp. 10 215–10 224.

[28] A. Lugmayr, M. Danelljan, L. V. Gool, and R. Timofte, "Srflow: Learning the super-resolution space with normalizing flow," in *ECCV*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12350. Springer, 2020, pp. 715–732.

[29] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *ICLR*, 2014.

[30] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *ICML*, 2019, pp. 2722–2730.

[31] X. Ma, X. Kong, S. Zhang, and E. H. Hovy, "Macow: Masked convolutional generative flow," in *NIPS*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 5891–5900.

[32] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "FFJORD: free-form continuous dynamics for scalable reversible generative models," in *ICLR*. OpenReview.net, 2019.

[33] T. Q. Chen, J. Behrmann, D. Duvenaud, and J. Jacobsen, "Residual flows for invertible generative modeling," in *NIPS*, H. M. W. andresflow Hugo Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 9913–9923.

[34] C. Huang, D. Krueger, A. Lacoste, and A. C. Courville, "Neural autoregressive flows," in *ICML*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 2083–2092.

[35] Y. Lu and B. Huang, "Structured output learning with conditional generative flows," in *AAAI*, vol. 34, no. 04, 2020, pp. 5005–5012.

[36] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Ethxgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in *ECCV*, 2020.

[37] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *CVPR*, 2015, pp. 4511–4520.

[38] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, J. Fürnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*. Springer, 2016, pp. 630–645.

[40] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE TPAMI*, vol. 32, no. 8, pp. 1362–1376, 2009.

[41] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *SGP*, vol. 7, 2006.