

# SalGaze: Personalizing Gaze Estimation using Visual Saliency

Zhuoqing Chang, Matias Di Martino, Qiang Qiu, Steven Espinosa, and Guillermo Sapiro  
Duke University, Durham, NC 27708, USA

{zhuoqing.chang, juan.di.martino, qiang.qiu, steven.espinosa, guillermo.sapiro}@duke.edu

## Abstract

*Traditional gaze estimation methods typically require explicit user calibration to achieve high accuracy. This process is cumbersome and recalibration is often required when there are changes in factors such as illumination and pose. To address this challenge, we introduce SalGaze, a framework that utilizes saliency information in the visual content to transparently adapt the gaze estimation algorithm to the user without explicit user calibration. We design an algorithm to transform a saliency map into a differentiable loss map that can be used for the optimization of CNN-based models. SalGaze is also able to greatly augment standard point calibration data with implicit video saliency calibration data using a unified framework. We show accuracy improvements over 24% using our technique on existing methods.*

## 1. Introduction

Gaze estimation is the problem of estimating a person's line of sight. It is important as eye gaze reflects a person's underlying cognitive process [11] which can be used in a wide array of applications including digital content marketing [50], diagnosing psychiatric conditions such as autism [23], and automated driving [2].

Due to differences in the structure and appearance of the eye, calibration is typically required to learn the parameters that are intrinsic to the user in order for gaze estimation algorithms to achieve high precision. This calibration process is typically in the form of having the user look at certain points on the target screen. One main issue with this process is that a one-time calibration typically only works in the same environment setting. Changes in factors such as illumination, head position, and facial appearance can drastically affect the estimation accuracy and recalibrating for each scenario is not feasible. Active gaze calibration can also be too restrictive in some scenarios. For instance, gaze is an important biomarker for autism risk assessment in toddlers [38], however, it is very challenging to require a toddler to perform active calibration as well as get feedback

on the calibration performance. The same challenges also appear for elderly populations with neurodegenerative disorders [3]. Due to these reasons, having a way to passively calibrate for the user is critical for making gaze estimation a more pervasive technology, in particular when using off-the-shelf devices, such as cameras embedded on mobile devices, and deploying in general in-the-wild scenarios.

With recent advances in machine learning, the combination of appearance-based gaze estimation and deep learning has become a popular method for remote estimation of gaze [53, 7, 54, 48, 13, 55]. Appearance-based algorithms have the advantage of only using an image of the person's face or eyes as input, therefore eliminating the need of any specialized hardware other than a regular camera. Deep learning algorithms have been shown to be powerful tools for gaze estimation due to its ability to incorporate factors including illumination, head pose, appearance, etc., using end-to-end learning. The biggest challenge with this approach is the need for a huge amount of data with ground truth gaze labels to train the network. Recent works, [1, 44, 14, 18, 56, 24, 51], have made great efforts on creating such large datasets, showing promising results. However, the process of collecting labeled gaze data is still a tedious task, and without powerful domain transfer techniques, which have yet to be demonstrated for gaze estimation, such data is limited to the devices, and scenarios for which it was collected. The lack of data as well as an efficient process for collecting data is one of the major bottlenecks of deep learning methods for gaze estimation. Addressing this challenge is the goal of this paper.

Cognitive science has shown that the human visual system has a strong tendency to focus on highly salient regions in the visual field. There has been extensive work in the area of computational saliency to emulate this behavior. While conventional research in this field leverages ground truth gaze data for visual saliency estimation, in this work we propose to invert the process and utilize visual saliency information for gaze estimation. We argue and demonstrate that saliency information within the scene can be used for calibration purposes without active participation from the user.

In this paper, we present SalGaze, a novel framework that leverages visual saliency information, properly processed as here introduced, for personalized gaze estimation using deep learning models. Calibration is transparently performed while the user watches a few short video clips. By using a here proposed differentiable loss map, SalGaze is also able to combine standard point-based calibration data with free-viewing video data under a unified framework, further improving performance if more accuracy is desired. We experiment with both empirical saliency data collected from eye trackers, and saliency data generated from two state-of-the-art saliency algorithms. We show accuracy improvements of more than 24% using our technique.

## 2. Related work

**Gaze Estimation.** One of the pioneering works in appearance-based gaze estimation was done by Tan *et al.* [45]. They used over 200 calibration samples to achieve very high accuracy under fixed head pose and illumination setting. Lu *et al.* [30] improved on this by drastically lowering the amount of calibration samples required while maintaining high accuracy. However, their method still required the person’s head to be completely still. Sugano *et al.* [44] and Chang *et al.* [5] used a multi-camera setup and a depth camera respectively to acquire 3D information and synthesize images of the eye under various head poses. An extensive survey of gaze estimation methods can be found in [15].

Many deep learning based algorithms for gaze estimation have been proposed over the last couple of years. Zhang *et al.* [56, 57] trained a convolutional neural network (CNN) on over 200,000 images collected from 15 participants over a 3 months period. Wood *et al.* [52, 51] used advanced computer graphics to synthesize large amounts of eye images to train CNN models. Shrivastava *et al.* [39] used a generative adversarial network (GAN) to produce more realistic synthesized eye images. Krafka *et al.* [24] used Amazon Turk to collect over 2 million images of people using iPhones and iPads. Park *et al.* [35, 34] trained a deep network to regress to intermediate eye landmarks and pictorial representations of the eyeball before estimating the gaze. These contributions have made significant breakthroughs in the area of calibration-free person-independent gaze estimation which shows the importance of having large amounts of data. Our work aims to extend these methods by allowing the use of personalized data that can be collected at scale.

**Gaze Personalization.** Appearance-based algorithms have mostly been focused on person-independent gaze estimation. To the best of our knowledge, only a couple of works have tried to address the challenge of personalizing the algorithm to a specific user. Krafka *et al.* [24] used the

CNN features of a few calibration samples to train a Support Vector Regression model. Several methods [28, 7, 53] have tried to explicitly incorporate person-dependent parameters in the model. These parameters are estimated with a few samples during testing. Others [55, 43] use synthesis techniques to augment the number of samples for a specific user to train a person-specific model or fine-tune a generic model. The above methods all require explicit calibration to collect limited samples of the user. Due to the transparent nature of our data collection process, we are not limited to collecting only a few samples from a new user, therefore allowing for better personalization.

**Saliency Prediction.** Seminal work in this area was done by Itti *et al.* [19]. They presented a computational model that extracted low-level features such as color and orientation to predict a global saliency map. More recently, a plethora of deep learning based methods have been proposed for static saliency prediction. Kümmerer *et al.* [26, 27] proposed two deep saliency prediction networks, DeepGaze I and DeepGaze II, that was built on the AlexNet [25] and VGG-19 [40] models respectively. Pan *et al.* [33] used a GAN to generate saliency maps. Cornia *et al.* [9] and Liu and Han [29] combines Long Short-Term Memory networks (LSTM) [17] with ResNet [16] to infer saliency by incorporating global and scene contexts. The problem of saliency prediction is not the focus of this work, we use saliency as a tool for gaze estimation. Further progress on computational saliency estimation can potentially further improve the results we obtain in this paper.

**Gaze and Saliency.** While the problem of gaze estimation and saliency prediction have received a lot of attention in each of their own respective areas, there are only a handful of works, to date, that make a connection between them. Sugano *et al.* [42] were the first to utilize saliency as a probability map for gaze estimation. They use Gaussian process regression to establish a mapping between eye images and gaze positions on a monitor under a fixed head pose setting. Chen *et al.* [6] used a model-based approach where the parameters of the eyeball are estimated in a probabilistic manner using saliency information of the stimuli. Their method is based on the pupil center corneal reflection (PCCR) which requires the use of an infrared camera in order to locate the pupil position. Many other works [36, 37, 12, 41, 8] uses the saliency information of an image to determine if a person in the same image is looking at a salient object. Our work differs from them as we use the saliency information of an out-of-frame target for precise gaze estimation. Contrary to these works, our method is designed for deep learning algorithms that utilize a loss function, and as such we design a new saliency-informed differentiable cost function which is also capable of combining

saliency information with point-wise calibration. The ability of transparently collecting large amounts of gaze data using saliency information further enhances the potential of our method.

### 3. Personalized Gaze Estimation from Saliency

In Section 3.1, we mathematically formulate the problem of 2D gaze estimation for standard point-wise calibration data. Then in Section 3.2, we extend the formulation to using saliency information and derive a solution by designing a differentiable loss map. We show that traditional point-based calibration is a special case of our solution and therefore can be combined with it. Implementation details of our CNN model are provided in Section 3.3.

#### 3.1. Point Loss

Let  $\Omega \subset \mathbb{R}^2$  denote an open set where a person’s gaze is tracked, e.g. a computer monitor or a phone screen,  $I$  the input, typically the face or eye image of the person or a combination of them, and  $f$  a model capable of estimating the person’s gaze  $\hat{p}$  from this image, i.e.,  $\hat{p} = f(I)$  ( $\hat{p} \in \Omega$ ).

The standard calibration procedure consists of collecting images  $I_1, \dots, I_n$  of users looking at pre-specified locations  $p_1, \dots, p_n$  ( $p_i \in \Omega$ ) on the screen. The parameters of the model  $f$  are denoted as  $\theta = (\theta_1, \dots, \theta_m)$ . We can optimize these parameters such that the empirical error on the collected data is minimized,

$$\theta_{opt} = \operatorname{argmin}_{\theta} \sum_{i=1}^n d(f_{\theta}(I_i), p_i)^2, \quad (1)$$

where  $d$  represents some distance between the predicted gaze  $\hat{p} = f(I)$  and the ground truth gaze  $p$ , e.g., the one induced by the L2 norm  $d(u, v) = \|u - v\|_2 = \sqrt{\sum_k (u_k - v_k)^2}$ . Equation (1) can be solved, for example, using stochastic gradient descent, as the loss  $\mathcal{L}(\theta) = \sum_{i=1}^n d(f_{\theta}(I_i), p_i)^2$  is differentiable,

$$\frac{\partial \mathcal{L}_i}{\partial \theta_j} = 2(f_{\theta}(I_i) - p_i)^T \frac{\partial f_{\theta}(I_i)}{\partial \theta_j}. \quad (2)$$

Since  $f$  is implemented using differentiable CNN models in this work,  $\frac{\partial f_{\theta}}{\partial \theta}$  is well defined and can be computed numerically. Gradient descent leads to the updating rule,

$$\theta^{t+1} = \theta^t - \delta \underbrace{(f_{\theta}(I) - p)}_{\hat{p}}^T \frac{\partial f_{\theta}(I)}{\partial \theta}, \quad (3)$$

where  $\delta$  represents the gradient descent step size. Essentially, the parameters of the model are slowly modified such that  $\hat{p} \rightarrow p$ . As shown in Eq. (3), the term proportional to  $-(\hat{p} - p)$  pushes the prediction of the model in the opposite direction of the error vector, therefore improving the accuracy. This simple optimization technique has been shown to be very robust in the context of gaze estimation [24].

#### 3.2. Probability Map Loss

We extend the ideas in the previous section to the scenario where, instead of precise point calibration data, we have, for each input image  $I_i$ , saliency information  $s_i : \Omega \rightarrow [0, 1]$  of the content the user is watching.  $s(x, y)$  can be interpreted as a measure of the likelihood that the user is looking at the point  $(x, y) \in \Omega$ .

A naive way to adapt Eq. (1) to exploit the new calibration data  $\{s_i\}$  would be,

$$\theta_{opt} = \operatorname{argmin}_{\theta} \sum_{i=1}^n g(s_i(f_{\theta}(I_i))), \quad (4)$$

where  $g$  is defined as some smooth monotonic decreasing function, e.g.,  $g(u) = -u$  or  $g(u) = 1/u$ . This formulation makes sense in regions where  $s(x, y) \neq 0$  since if the model predicts a position  $f_{\theta}(I)$  of high probability,  $s(f_{\theta}(I))$  would be large, and the loss  $g(s(f_{\theta}(I)))$  would be small. However, for regions where  $s(x, y) = 0$ , which could often occur in saliency maps,  $\|\nabla s\|$  is zero and a gradient descent-like optimization technique will fail.

In order to obtain a well-posed and robust optimization scheme, we propose to compute a loss map  $l(s)$  with the following properties:

- $l(s)$  should be continuous, differentiable and have properties described in 3.1;
- it should encourage predictions to occur at regions with large saliency values;
- for point data,  $l(s)$  should represent the distance to this point as in Eq. (1).

To that end, we adapt ideas from [4] and implement a *Reinitialization-like* equation as we detail next.

First we set a threshold  $\lambda$  and compute a binary map  $l_0 : \Omega \rightarrow \{0, 1\}$ ,  $l_0(x, y) = 1$  if  $s(x, y) < \lambda$  and 0 otherwise. Then we use this binary image as the initial condition of the Partial Differential Equation (PDE)

$$\begin{cases} \frac{\partial u((x, y), t)}{\partial t} + l_0(x, y)(\|\nabla u((x, y), t)\| - 1) = 0, \\ u((x, y), 0) = l_0(x, y). \end{cases} \quad (5)$$

Figure 1 illustrates the evolution of  $u((x, y), t)$ . Algorithm 1 describes the numerical implementation of the reinitialization scheme described in Eq. (5). A robust implementation of the gradient computation step in Algorithm 1 is given in Algorithm 2.

**Definition 3.1.** Let  $\Gamma$  be a close set in  $\Omega$  and  $\operatorname{dist}(p, \Gamma)$  the distance of a point  $p$  to the set  $\Gamma$  defined as

$$\operatorname{dist}(p, \Gamma) = \inf_{q \in \Gamma} \|q - p\|_2. \quad (6)$$

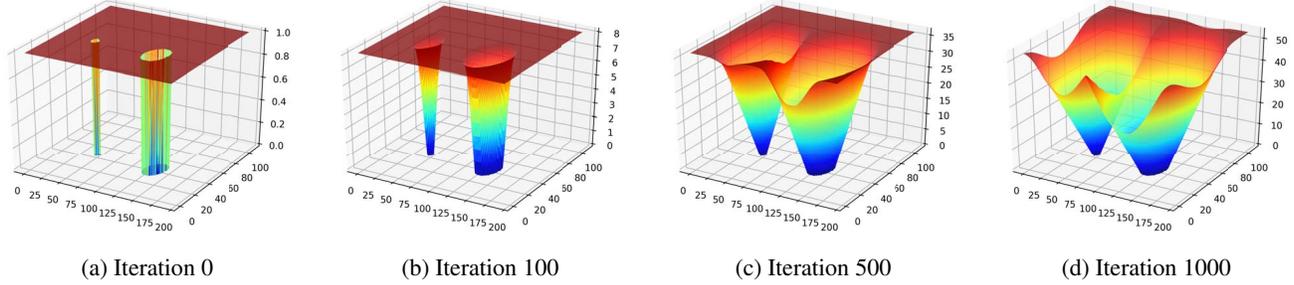


Figure 1: Evolution from binary map to loss map. (a), (b), (c), and (d) shows the loss map at different iterations of the reinitialization algorithm.

**Definition 3.2.** The skeleton of  $\Gamma$ , denoted by  $S_\Gamma$ , is the set of points  $x \in \mathbb{R}^2$  such that there exist at least two distinct points  $y$  and  $z$  in  $\Gamma$  satisfying

$$|x - y| = |x - z| = \text{dist}(x, \Gamma).$$

**Proposition 3.1.** If  $\Gamma$  is a closed subset of  $\Omega$ ,  $u(p) = \text{dist}(p, \Gamma)$ , and  $\bar{S}_\Gamma$  denotes the closure of the skeleton of  $\Gamma$ ,

1.  $u$  is 1-Lipschitz for all  $p \in \Omega$ ,
2.  $\|\nabla u\| = 1$  for all  $p \in \Omega \setminus \bar{S}_\Gamma$ .

**Proposition 3.2.** Let  $\Gamma = \{q \in \Omega / l_0(q) = 0\}$ . The function  $u : \Omega \times \mathbb{R}_+ \rightarrow \mathbb{R}$  defined by,

$$u(p, t) = \begin{cases} \inf_{|q| \leq t} (l_0(p + q) + t) & \text{if } t \leq t_p, \\ \text{dist}(p, \Gamma) & \text{if } t > t_p, \end{cases} \quad (7)$$

where

$$t_p = \inf \{t \in \mathbb{R}_+ / \inf_{|q| \leq t} (l_0(p + q)) = 0\}$$

is the unique solution of (5) uniformly continuous on  $\Omega \times [0, T]$ ,  $\forall T > 0$  and vanishing on  $\Gamma \forall t \in [0, T]$ .

Proposition 3.1 is similar to the Proposition 4.3.1 presented in [4] in that we define a positive distance function instead of a signed distance function, and  $\Gamma$  as a closed subset of  $\Omega$  instead of a closed curve in  $\mathbb{R}^2$ . In addition, Proposition 3.2 can be proved with a straightforward adaptation of the proof of Theorem 4.3.4 presented in [4].

We define  $l(x, y)$  as the solution of Eq. (5) with initial condition  $l_0(x, y)$ , i.e.,  $l(x, y) = u((x, y), t^*)$  for any  $t^*$  larger than the longest straight line contained in  $\Omega$ , therefore,  $l(x, y)$  is the distance of  $(x, y)$  to the set defined by  $\{q \in \Omega / l_0(p) = 0\}$ . Finally, given the data  $\{I_i, s_i\}$ , we compute the corresponding loss maps  $\{l_i\}$ , and adapt the gaze model by,

$$\theta_{opt} = \underset{\theta}{\text{argmin}} \sum_{i=1}^n l_i(f_\theta(I_i))^w. \quad (8)$$

We can chose  $w = 1$  or  $w = 2$  to minimize the distance or the squared distance respectively. We refer to the former as the *map loss* and the latter as the *squared map loss*. When the prediction  $\hat{p}$  is outside the valid set  $l_0 = 0$ , the gradient of our defined loss will update the model to predict towards the closest point in the set. The norm of the gradient is proportional to the distance to the set when  $w = 2$ , or a unitary vector when  $w = 1$ . As we will show in Section 4.4,  $w = 2$  leads to a faster convergence rate, while  $w = 1$  is more robust to outliers.

It is important to mention that point calibration data can be seen as a special case where for a given calibration point  $p$ , its equivalent saliency map is  $s(x, y) = 0, \forall (x, y) \in \Omega \setminus \{p\}$  and  $s(p) = 1$ . Therefore our formulation allows the use of both point and saliency data sources for gaze estimation.

---

**Algorithm 1** Reinitialization Equation. Implementation of the solution of the PDE described in Eq. (5).

---

**Input:** binary loss map  $l_0 : [1...H] \times [1...W] \rightarrow \{0, 1\}$

- 1: Set iteration parameters:
  - 2:  $\text{tol} = 0.01$  stop criteria
  - 3:  $\delta = .1$  step size
  - 4: Initialize variables.
  - 5:  $\text{diff} = \text{tol} + 1$
  - 6:  $u_0 = l_0$
  - 7: **while**  $\text{diff} > \text{tol}$  **do**
  - 8:  $u_{0x}, u_{0y} = \text{computeGradient}(u_0)$  (see Alg. 2)
  - 9:  $u_1 = u_0 - \delta l_0 \left( \sqrt{u_{0x}^2 + u_{0y}^2} - 1 \right)$
  - 10:  $\text{diff} = \|u_1 - u_0\|_2$
  - 11:  $u_0 = u_1$
  - 12: **end while**
  - 13: **return**  $u_1$
- 

### 3.3. Implementation Details

Our model architecture is based on iTracker [24], a CNN that predicts the user's gaze position on an iPhone or iPad.

**Algorithm 2** computeGradient (robust implementation).

**Input:**  $u : [1\dots H] \times [1\dots W] \rightarrow [0, 1]$ .

- 1: Initialize
- 2:  $\delta_x^+ = \text{zeros}(H, W)$
- 3:  $\delta_y^+ = \text{zeros}(H, W)$
- 4:  $\delta_x^- = \text{zeros}(H, W)$
- 5:  $\delta_y^- = \text{zeros}(H, W)$
- 6: Compute the nonoscillatory upwind scheme (see [4] appendix A.3 for details).
- 7:  $\delta_x^+[:, 2 : -2] = \max(u[:, 2 : -2] - u[:, 1 : -3], 0)$
- 8:  $\delta_y^+[2 : -2, :] = \max(u[2 : -2, :] - u[1 : -3, :], 0)$
- 9:  $\delta_x^-[:, 2 : -2] = -\min(u[:, 3 : -1] - u[:, 2 : -2], 0)$
- 10:  $\delta_y^-[2 : -2, :] = -\min(u[3 : -1, :] - u[2 : -2, :], 0)$
- 11:
- 12:  $u_x = \max(\delta_x^+, \delta_x^-)$
- 13:  $u_y = \max(\delta_y^+, \delta_y^-)$
- 14: **return**  $u_x, u_y$

The network uses images of the user’s face collected from the front facing camera of the device as input, and outputs the position on the screen of where the user is looking at. Specifically, the face and two eye regions of the image as well as the face grid, a binary matrix representing the spacial position of the face inside the image, are used as 4 inputs to the network. The output is the 2D gaze position relative to the camera. This technique allows for pooling data from different device models and device orientations to train a relatively robust model, although its performance still degrades when extrapolating to new conditions (e.g., different devices or acquisition scenarios).

We modify the iTracker architecture in the following way: (1) The size of the input face and eye images is reduced from  $224 \times 224$  to  $64 \times 64$  to reduce training time. (2) An extra fully connected layer is added at the end of the network for fine-tuning purposes. (3) The local response normalization layer after each convolutional block is replaced with the superior batch normalization layer. (4) The mean squared loss is replaced with our custom map loss defined in Section 3.2. The final model architecture is shown in Figure 2. Adapting a known architecture to the new personalized formulation helps to illustrate the plug-and-play style of our proposed framework.

Our model is implemented in Python using Tensorflow. The optimization is performed using the Adam [22].

## 4. Experiments

In this section, we perform experiments to evaluate how different hyperparameters, data quantity, and different types of saliency computations affect the performance of our method.

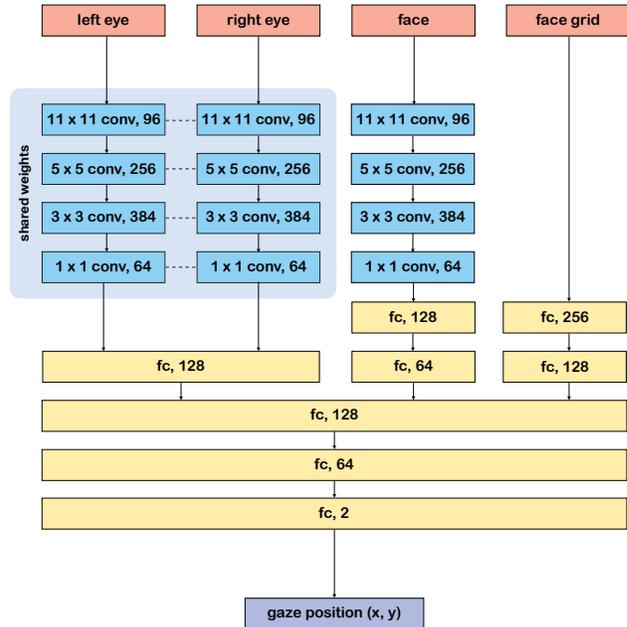


Figure 2: Network architecture

### 4.1. Data Collection

We asked 9 participants (7 male, 2 female) to each record 10 sessions of data using an iPad Air 2 in landscape orientation. Each session is composed of 2 phases: a point phase and a video phase. The point phase is similar to a traditional calibration procedure where 20 points sequentially appear at random locations on the screen for 2 seconds each. In the video phase, several short 15 to 20 second video clips are played with rest intervals between them. The 10 sessions of data for each participant are collected in different locations across several days to encourage variability in pose, illumination, and general acquisition environments.

The videos we show to participants come from two public saliency databases: SAVAM [31] and Coutrot Database 1 [10]. SAVAM contains 41 fragments of  $1920 \times 1080$  motion video from various feature movies, commercial clips, and stereo video databases. Coutrot Database 1 contains 60 videos of people or faces, moving objects, and landscapes at  $720 \times 576$  resolution. We use the set of videos with original soundtracks for Coutrot Database 1. Ground truth saliency data collected with commercial eye trackers are provided for each video. From these videos, we manually selected 33 that we found relatively more engaging. The videos we used are  $\{2, 5, 6, 11, 12, 22, 24, 29, 30, 34, 38\}$  from SAVAM and  $\{6, 8, 15, 17, 18, 21, 23, 24, 25, 26, 28, 46, 47, 48, 50, 51, 52, 53, 54, 56, 57, 59\}$  from Coutrot Database 1. Participants view each video only once across the 10 sessions.

## 4.2. Data Preprocessing

We compute the loss map for each video frame by first mapping the saliency maps from the pixel coordinate space to the camera coordinate space defined in [24] based on the device specifications and device orientation. We use a  $101 \times 101$  grid to represent the square area between -25cm and 25cm in the camera coordinate space. The loss map is then computed using the method described Section 3.2. We set  $\lambda$  equal to the 95th percentile of the saliency map for generating the binary map. This process is shown in Figure 3.

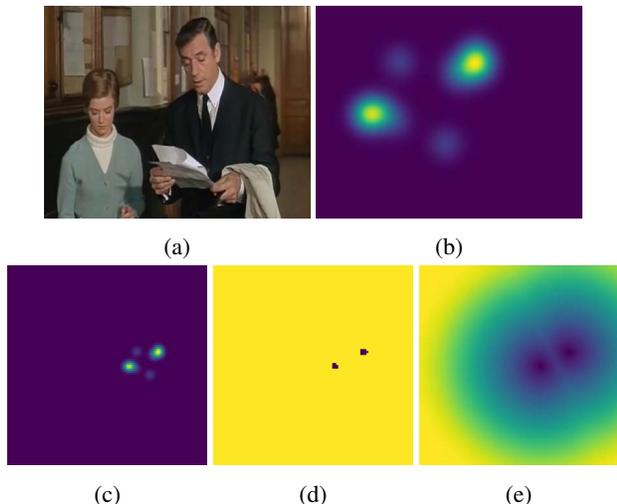


Figure 3: Loss map generation process. The saliency image (b) of a particular video frame (a) is mapped to a  $101 \times 101$  grid representing the camera coordinate space (the prediction space) from -25cm to 25cm. The mapped saliency image (c) is then converted into binary image (d) after which the reinitialization algorithm is used to generate the differentiable loss map (e).

The recorded videos are synced to the presented videos such that each recorded frame in the point phase is associated with the presented point location and each recorded frame in the video phase is associated with the loss map of the corresponding presented video frame. Due to high similarity between consecutive frames, we sample the recorded video and loss maps at 1 frame per second. We also discard any frame within the first 0.5 seconds of a point appearing to allow for participants to focus their gaze. Dlib [21] is then used to detect the face and facial landmarks in each frame. We use the face region given by Dlib to crop out the face image and compute the face grid defined in [24]. We define the eye region as the square area centered at the midpoint of the eye corner landmarks with a side length equal to 1.8 times the horizontal distance between the eye corner landmarks. We found this results in similar eye images to [24] which uses a native iOS algorithm to generate the croppings.

## 4.3. Baseline Model

Our baseline model is trained using the same training and validation data from GazeCapture [24]. We resize the input images to  $64 \times 64$  and train for 75,000 iterations with a batch size of 256. A comparison of our baseline model to iTracker [24] on 3 subsets of the GazeCapture test data is shown in Table 1. Due to decreasing the resolution of the input images, which we did for computational purposes, our baseline model performs slightly worse than iTracker. However, our goal here is to not to outperform iTracker but rather to set a baseline for indirect comparison with our personalized models in the following sections. While they do not report the performance on the set of test data collected on the iPad Air 2 in landscape orientation, if we extrapolate based on the trend of our baseline model, the error is approximately 3.6cm. We show, in the next section, that our low-resolution personalized model can achieve an error of 3.3cm.

Model	Test Data Partition		
	All Phones	All Tablets	iPad Air 2 (Landscape)
iTracker [24]	2.04	3.32	N/A
Baseline	2.26	3.76	4.07

Table 1: Mean error (cm) of our baseline model and iTracker on different test sets of GazeCapture.

## 4.4. Calibration with Saliency

We validate the usefulness of saliency data for personalized gaze estimation by fine-tuning the last two fully connected layers of our baseline model described in Section 4.3. For each participant, we use data from the video phase and their corresponding loss maps for fine-tuning and test on data from the point phase. We compare the effect of using the map loss with the squared map loss described in Section 3.2. As the user is not constrained or guided to look at specific locations during the video phase, we expect the collected data to present outliers where the input image does not coincide with the saliency map. Therefore, we also explore the effect of an iterative outlier removal (IOR) technique where we remove the top 5% of data with the largest loss from training every 2 epochs. Appropriate filtering of the input videos and saliency data is important for the proposed framework, and this simple approach has been found sufficient; we discuss more on this in the Section 5. For each participant, the baseline model is fine-tuned for 10 epochs per configuration.

Figure 4 shows the average gaze error across all participants for each epoch. It can be seen that the error stabilizes after 4 epochs. Therefore, we decide to fix the fine-tuning procedure to 4 epochs for this and subsequent experiments.

Quantitative results are shown in Table 2. It can be seen that all 4 configurations improve over the baseline and IOR helps with reducing the error. As expected and commented in Section 3.2, the squared loss leads to a faster convergence but is more susceptible to outliers with which IOR helps to mitigate. For the experiments in sections 4.5 and 4.6, we use the map loss with IOR configuration.

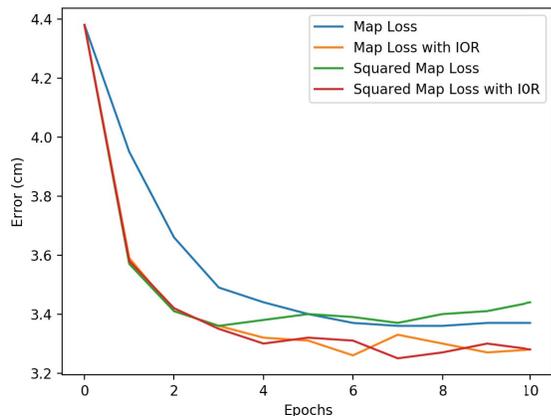


Figure 4: Average gaze error using different training configurations.

Configuration	Error (cm)	Improvement (%)
Baseline	$4.38 \pm 1.18$	0
Map Loss	$3.44 \pm 0.74$	21.46
Map Loss w/ IOR	$3.32 \pm 0.77$	24.20
Squared Map Loss	$3.38 \pm 0.73$	22.83
Squared Map Loss w/ IOR	$3.30 \pm 0.78$	24.66

Table 2: Average gaze error using different training configurations.

#### 4.5. Scalability with Data Quantity

Contrary to standard point-wise calibration, using visual saliency for calibration is transparent and therefore provide a means for collecting large amounts of data in a non-intrusive way. To examine how the performance of our approach scales with the amount of data, we use data from 1, 2, 4, and 8 of the 10 sessions to fine-tune the baseline model for each participant. Similar to Section 4.3, fine-tuning is performed on data from the video phase of the selected sessions, and testing is performed on data from the point phase of all 10 sessions. Quantitative results are shown in Table 3. It can be seen that the estimation error decreases with the increase of training data.

Sessions of data used for training	Error (cm)	Improvement (%)
Baseline	$4.38 \pm 1.18$	0
1	$3.73 \pm 0.77$	14.84
2	$3.62 \pm 0.77$	17.35
4	$3.46 \pm 0.78$	21.00
8	$3.38 \pm 0.71$	22.83

Table 3: Gaze estimation error using different amounts of training data.

#### 4.6. Generated vs. Empirical Saliency

In the above experiments, the saliency data we use for fine-tuning were collected by [10, 31] using commercial eye trackers. While this gives us reliable data, it also limits us to using videos of which ground truth saliency is available. Having the capability to use arbitrary videos would greatly broaden the applicability of our method. In this experiment, we explore the potential of using saliency prediction algorithms to generate saliency maps for gaze estimation. Specifically, we examine two state-of-the-art saliency prediction algorithms: SAM [9], an image saliency prediction algorithm; and ACL [49], a video saliency prediction algorithm. We use the ResNet version of SAM that is trained on the 2015 SALICON [20] dataset. ACL is trained on Hollywood-2 [32], UCF sports [32], and DHF1K [49] datasets. Both SAM and ACL models are publicly available. We independently run SAM and ACL on the video clips to generate their saliency data. We refer to the saliency data collected from eye trackers as empirical saliency data. Examples of some video frames and their respective saliency maps are shown in Figure 5.

For each type of saliency data, we personalize the baseline model in the same way as sections 4.4 and 4.5 where, for each participant, the data from the video phase is used for fine-tuning and data from the point phase is used for testing. Essentially, the network is seeing the same inputs but with different loss map labels. Quantitative results are shown in Table 4. We can see that in terms of performance, Empirical > ACL > SAM. This is consistent with the assumption that empirical saliency from eye trackers is more accurate than their generated counterpart. Furthermore, since we use video content for data collection, it makes sense that using the video-based algorithm ACL performs better than the image-based algorithm SAM. Nevertheless, using any of the three types of saliency data show clear improvement over the baseline.

### 5. Discussion

In this section we discuss two main challenges of using saliency data for gaze calibration.

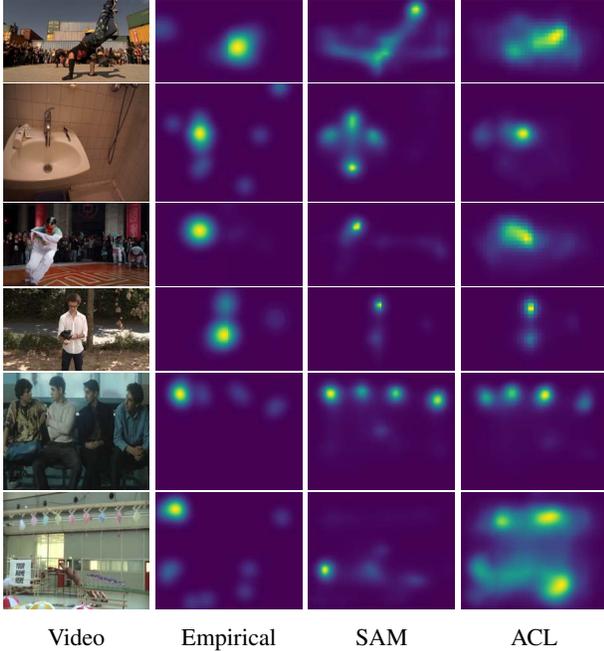


Figure 5: Examples of videos frames and their empirical and generated saliency maps.

Saliency Type	Error (cm)	Improvement (%)
Baseline	$4.38 \pm 1.18$	0
Empirical	$3.32 \pm 0.77$	24.20
SAM	$3.99 \pm 0.58$	8.90
ACL	$3.70 \pm 0.54$	15.53

Table 4: Gaze estimation error using different types of saliency computations.

### 5.1. Imprecise Labels

One of the biggest differences between our approach and traditional gaze estimation methods is that we do not have precise ground truth gaze labels. Since there is no constraint on where the user is looking, there is a portion of data where the input image is not consistent with the associated saliency map. This portion of data can be interpreted as mislabelled data which will hinder the network from learning the correct parameters. In this paper, we tried to address this issue using an iterative outlier removal technique. However, we believe that using a more sophisticated method to filter the training data such that there is higher correlation between the input images and saliency labels as well as selecting genres of videos with clearer saliency regions (e.g., sports) will further improve the performance.

### 5.2. Unbalanced Data

In the study of visual saliency, the center bias is a well known phenomenon where objects of interest appear more often near the center of the scene [47, 46]. Figure 6 shows the average distribution of the empirical and generated saliency maps across all videos for SAVAM and Coutrot Database 1. This non-uniformity skews the prediction to favor the center region which results in larger errors at the boundaries. We think one area of future work could be directed at correcting or compensating for this bias, e.g., sample the training data to have a more uniform distribution, or use a spatially weighted loss.

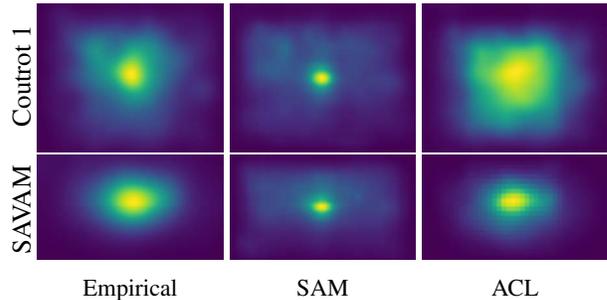


Figure 6: Average saliency distributions of all videos from SAVAM and Coutrot Database 1.

## 6. Conclusion

In this paper, we proposed SalGaze, a novel framework for gaze estimation using visual saliency information. We designed an algorithm to transform a saliency map into a differentiable loss map that is well suited for the optimization of CNN-based models. SalGaze is able to combine implicit video calibration data with explicit point calibration data using a unified framework. Our technique does not require explicit attention from the user and can run in the background while the user uses the device. This lack of constraints may lead to outliers in the training data. We show that we are able to partially overcome this issue by means of an iterative outlier removal procedure. Our method also enables the collection of large amounts of gaze data which is critical for deep learning based algorithms. We show accuracy improvements over 24% after adapting a state-of-the-art gaze estimation algorithm with saliency information using SalGaze.

## Acknowledgements

Work partially supported by NSF, ARO, ONR, NGA, NIH, Simons Foundation, Amazon AWS, and Microsoft.

## References

- [1] Gaze Locking: Passive Eye Contact Detection for Human?Object Interaction. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 271–280, Oct 2013. [1](#)
- [2] S. Alletto, A. Palazzi, F. Solera, S. Calderara, and R. Cucchiara. Dr(eye)ve: A dataset for attention-based tasks with applications to autonomous and assisted driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016. [1](#)
- [3] T. J. Anderson and M. R. MacAskill. Eye movements in patients with neurodegenerative disorders. *Nature Reviews Neurology*, 9(2):74, 2013. [1](#)
- [4] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer Publishing Company, Incorporated, 2nd edition, 2010. [3](#), [4](#), [5](#)
- [5] Z. Chang, Q. Qiu, and G. Sapiro. Synthesis-based low-cost gaze analysis. In *International Conference on Human-Computer Interaction*, pages 95–100. Springer, 2016. [2](#)
- [6] J. Chen and Q. Ji. Probabilistic gaze estimation without active personal calibration. In *CVPR 2011*, pages 609–616. IEEE, 2011. [2](#)
- [7] Z. Chen and B. E. Shi. Appearance-based gaze estimation via gaze decomposition and single gaze point calibration. *arXiv preprint arXiv:1905.04451*, 2019. [1](#), [2](#)
- [8] E. Chong, N. Ruiz, Y. Wang, Y. Zhang, A. Rozga, and J. M. Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 383–398, 2018. [2](#)
- [9] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018. [2](#), [7](#)
- [10] A. Coutrot and N. Guyader. Toward the introduction of auditory information in dynamic visual attention models. In *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4. IEEE, 2013. [5](#), [7](#)
- [11] M. K. Eckstein, B. Guerra-Carrillo, A. T. M. Singley, and S. A. Bunge. Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Developmental Cognitive Neuroscience*, 25:69 – 91, 2017. Sensitive periods across development. [1](#)
- [12] L. Fan, Y. Chen, P. Wei, W. Wang, and S.-C. Zhu. Inferring shared attention in social scene videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6460–6468, 2018. [2](#)
- [13] T. Fischer, H. Jin Chang, and Y. Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 334–352, 2018. [1](#)
- [14] 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*, pages 255–258. ACM, 2014. [1](#)
- [15] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2009. [2](#)
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [2](#)
- [17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [2](#)
- [18] Q. Huang, A. Veeraraghavan, and A. Sabharwal. Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28(5-6):445–461, 2017. [1](#)
- [19] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998. [2](#)
- [20] M. Jiang, S. Huang, J. Duan, and Q. Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015. [7](#)
- [21] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. [6](#)
- [22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#)
- [23] A. Klin, W. Jones, R. Schultz, F. Volkmar, and D. Cohen. Visual Fixation Patterns During Viewing of Naturalistic Social Situations as Predictors of Social Competence in Individuals With Autism. *JAMA Psychiatry*, 59(9):809–816, 09 2002. [1](#)
- [24] K. Krafska, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016. [1](#), [2](#), [3](#), [4](#), [6](#)
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [2](#)
- [26] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*, 2014. [2](#)
- [27] M. Kümmerer, T. S. Wallis, and M. Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*, 2016. [2](#)
- [28] E. Lindén, J. Sjöstrand, and A. Proutiere. Appearance-based 3d gaze estimation with personal calibration. *arXiv preprint arXiv:1807.00664*, 2018. [2](#)
- [29] N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Transactions on Image Processing*, 27(7):3264–3274, 2018. [2](#)
- [30] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *2011 International Conference on Computer Vision*, pages 153–160. IEEE, 2011. [2](#)

- [31] V. Lyudvichenko, M. Erofeev, Y. Gitman, and D. Vatolin. A semiautomatic saliency model and its application to video compression. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 403–410. IEEE, 2017. 5, 7
- [32] S. Mathe and C. Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1408–1424, 2014. 7
- [33] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O’Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017. 2
- [34] S. Park, A. Spurr, and O. Hilliges. Deep pictorial gaze estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2
- [35] S. Park, X. Zhang, A. Bulling, and O. Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, page 21. ACM, 2018. 2
- [36] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In *Advances in Neural Information Processing Systems*, pages 199–207, 2015. 2
- [37] A. Recasens, C. Vondrick, A. Khosla, and A. Torralba. Following gaze in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1435–1443, 2017. 2
- [38] N. J. Sasson and J. T. Elison. Eye tracking young children with autism. *JoVE (Journal of Visualized Experiments)*, (61):e3675, 2012. 1
- [39] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017. 2
- [40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2
- [41] H. Soo Park and J. Shi. Social saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4777–4785, 2015. 2
- [42] Y. Sugano, Y. Matsushita, and Y. Sato. Calibration-free gaze sensing using saliency maps. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2667–2674. IEEE, 2010. 2
- [43] Y. Sugano, Y. Matsushita, and Y. Sato. Appearance-based gaze estimation using visual saliency. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):329–341, 2012. 2
- [44] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, 2014. 1, 2
- [45] K.-H. Tan, D. J. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pages 191–195. IEEE, 2002. 2
- [46] B. W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of vision*, 7(14):4–4, 2007. 8
- [47] P.-H. Tseng, R. Carmi, I. G. Cameron, D. P. Munoz, and L. Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of vision*, 9(7):4–4, 2009. 8
- [48] K. Wang, R. Zhao, H. Su, and Q. Ji. Generalizing eye tracking with bayesian adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11907–11916, 2019. 1
- [49] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4903, 2018. 7
- [50] M. Wedel and R. Pieters. Eye tracking for visual marketing. *Foundations and Trends in Marketing*, 1(4):231–320, 2008. 1
- [51] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016. 1, 2
- [52] E. Wood, T. Baltrušaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV 2015)*, 2015. 2
- [53] Y. Xiong, H. J. Kim, and V. Singh. Mixed effects neural networks (menets) with applications to gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7743–7752, 2019. 1, 2
- [54] Y. Yu, G. Liu, and J.-M. Odobez. Deep multitask gaze estimation with a constrained landmark-gaze model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 1
- [55] Y. Yu, G. Liu, and J.-M. Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 2
- [56] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015. 1, 2
- [57] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):162–175, 2017. 2