On-device Few-shot Personalization for Real-time Gaze Estimation

Junfeng He* Google Inc. junfenghe@google.com
Khoi Pham* University of Maryland at College Park khoi@cs.umd.edu
Nachiappan Valliappan Google Inc. nac@google.com
Pingmei Xu Google Inc. pingmeix@google.com
Chase Roberts Google Inc. chaseriley@google.com
Dmitry Lagun Google Inc. dlagun@google.com
Vidhya Navalpakkam Google Inc. vidhyan@google.com

Abstract

Building fast and accurate gaze estimation models without additional specialized hardware is a hard problem. In this paper, we present on-device few-shot personalization methods for 2D gaze estimation. The proposed supervised method achieves better accuracy using as few as 2-5 calibration points per user compared to prior methods that require more than 13 calibration points. In addition, we propose an unsupervised personalization method which uses only unlabeled facial images to improve gaze estimation accuracy. Our best personalized model achieves 24-26% better accuracy (measured by mean error) on phones compared to the state-of-the-art using <=5 calibration points per user. It is also computationally efficient, requiring 20x fewer FLOPS when compared to prior methods. This unlocks a variety of important real world applications such as using gaze for accessibility, gaming and human-computer interaction while running entirely on-device in real-time.

1. Introduction

Eye tracking or automated gaze estimation to infer user's visual attention and behavior is a fundamental component in numerous applications, including human-computer interaction [16, 23, 24, 25], behavior monitoring [24, 3], vision-systems [27, 32], Augmented Reality/Virtual Reality, medical diagnoses [12], and gaming [7]. With the recent success of deep models in computer vision tasks, convolutional neural network (CNN) based approaches have become popular [1, 47, 19, 37, 46, 20, 2, 8, 44, 45, 40, 42] in gaze estimation research.

High accuracy is a requirement for many gaze applications listed above. One popular approach to improving gaze estimation accuracy is personalization [19, 46], where the model is customized for each user using personal labeled data (known as calibration points in eye tracking research). The calibration data is collected by asking users to look at specific locations on the screen, and capturing the corresponding front-facing camera frame as the input, and the screen location as the gaze label.

*equal contribution

One approach to personalizing deep models is to fine-tune the model weights of the last few layers with additional calibration data from the user. There are two main limitations with this approach: (a) effectively fine-tuning the model requires a large amount of calibration data, and (b) it is computationally difficult to update model parameters on-device. In [19], the authors demonstrate improved 2D gaze estimation using a personalized support vector regression (SVR) on top of a deep learning model. This approach works well with large number of calibration points, however, the accuracy is poor if the number of calibration points is $<$ 5-9.

Building an accurate, personalized gaze-estimation model that is fast, runs on-device, and uses few calibration points is still an open problem in the field. On-device personalization is preferred over server-based solutions, as it presents several benefits including low latency, fewer data privacy/security concerns, and support for poor (or no) network connectivity scenarios, thereby enabling a large number of applications in accessibility, gaming and human-computer interaction.

In this paper, we propose an end-to-end CNN model for on-device personalized gaze estimation that achieves significantly better accuracy than prior art while requiring only few calibration points. The key contributions are:

- A supervised personalization method using very few labeled calibration points ($<=$ 5), with embedding based few-shot learning.
- An *unsupervised* few-shot personalization method to improve gaze estimation accuracy with a few *unlabeled* images from the user, based on heterogeneous

teacher-student network.

- An improved CNN model architecture for calibration-free 2D gaze estimation, *SAGE* (fa**S**t **A**ccurate **G**aze track**E**r) which is more computationally efficient compared to prior work in this area [19, 37, 13].
- Combining SAGE with our *supervised* few-shot personalization method offers a gaze-estimation model that yields better accuracy (24-26%) using very few calibration points (<=5) when compared to SOTA, runs real-time on-device and is significantly faster ($\sim$20x speedup in inference, 10ms on Pixel 2).

Our paper is organized as follows: Section 2 describes related work on gaze estimation, few-shot learning, and teacher-student network; Section 3 explains our proposed methodology, where in Section 3.1, we introduce SAGE, our calibration-free gaze estimation CNN model, and in Sections 3.2 discuss our supervised personalization methods . In Section 4, we proposed a novel unsupervised personalization method. We conclude with experiments to support the proposed methods and compare them with the best known baselines in Section 5.

2. Related Work

Gaze Estimation: The problem of estimating gaze from an image has become an active research area. Broadly speaking, the gaze estimation problem can be divided into two categories: 2D gaze and 3D gaze, where the former refers to estimating the (x, y) gaze location on the device's screen [13, 19, 47] while the latter refers to estimating the 3D gaze vector representing the gaze direction [47, 37, 46, 34, 6, 28, 39]. In this paper, we will focus only on 2D gaze on screen estimation.

Gaze estimation approaches can be classified as either model-based or appearance-based. Model-based methods model the geometric structure of the eye region, e.g., iris contour, while appearance-based methods directly estimate the gaze position or direction from the input image. Early research on gaze estimation often used model-based methods. For example, the methods in [10, 26, 5] use an infrared light source or high-quality image sensor to separate the iris from the rest of the image, while other methods try to fit a geometric model to the entire face [15, 4] or occluded face [43]. In contrast, appearance-based approaches estimate the gaze position or direction using a regression procedure based on the face or eye region images. Due to the lack of training data, a common simplification is to introduce additional knowledge, such as head pose into the regression framework, or to train only shallow models [22, 33, 35, 14]. Recent work from [28] tries to map eye images to simplified representations before regressing to a 3D prediction.

Krafka et. al. [19] proposed iTracker, a CNN for 2D gaze estimation. In summary, the model takes four image inputs that are all extracted from the camera frame: (1) left eye image, (2) right eye image, (3) face image, and (4) face grid. The eye images indicate the pose of eyes relative to the face, while the face image and face grid describe the user's head pose relative to the camera and capture the user's distance to the camera. For more details, we refer the interested reader to the paper [19]. They also collected and published *Gaze-Capture*, a large public 2D gaze dataset with almost $\sim$1500 subjects and $\sim$2.5 million frames. Using the iTracker model and the large dataset, Krafka et. al. showed that a deep CNN-based network can perform reasonably well for 2D gaze estimation. Recent work also discusses using generative adversarial networks (GAN) to create synthesized training data to improve 3D gaze estimation accuracy [34, 41].

Next, we discuss prior work on personalized gaze estimation. There exist some approaches for personalized 3D gaze estimation [39, 21, 20], however, in this paper, we will mainly focus on personalized 2D gaze estimation works. In [46], a person-specific model is proposed, where person (or device) specific encoder and decoder layers are trained with data from that person (or device), and the shared feature extraction layers are trained with the full dataset. It is worth noting that the encoder and decoder require fine-tuning in order to adapt the model to a new subject. Such personalization is computationally intensive, requires a large amount of calibration data, and cannot be run on-device. Krafka et. al. [19] proposed a personalized version of iTracker, where a Support Vector Regression (SVR) model is trained with a few labeled calibration points from every subject. The input feature for the personalized-SVR is the final FC layer representation of the input, in the iTracker model. While this approach is amenable for on-device inference, as shown in [19] and our experiments, it fails to generalize when only few examples are available (< 5 labeled calibration points). In contrast, our proposed method can improve the 2D gaze estimation accuracy while having access to only a few labeled samples. o developed a personalized 3D gaze model.

Few-shot Learning: Up to now, few-shot learning based on deep neural nets is mostly proposed for classification tasks, where the classifier needs to be adapted to new classes with just a few (labeled) examples of each class. Deep few-shot classification can be grouped into two categories: meta-learning-based methods [31, 30], and embedding-based methods [17, 38, 36, 29, 18, 9]. With meta-learning, the classifier itself still needs many data to train, on top of which a meta-learner is learned with few examples, e.g., to update the parameters of the classifier [30]. As with embedding-based methods, an embedding space is trained on all examples (from both classes with many examples and new classes with few examples) usually with a neural network consisting of convolution layers and FC layers. Then examples of new classes are mapped to the embedding space, and some kind of classification system, (e.g. based on nearest neighbors or distance to class prototype like class center, etc.), is applied on the embedding space so that the classifier can learn the new classes with only a few samples. For instance, the classifier in embedding space for new classes can be computed from distance to the class centers (or its probability variation) [36], neighborhood components analysis [29], attention mechanism [38].

Our proposed supervised personalization approach is inspired by embedding-based few-shot classification methods. However, there are substantial differences between our work and previous few shot classification research: first, our work is in the context of gaze estimation, a regression problem instead of classification problem; second, our task, model personalization, is rarely discussed in previous few-shot learning area; finally and more importantly, our few-shot personalization model runs on-device in real-time, which is seldom studied.

Teacher-Student Network: Most teacher-student network architectures [11] assume input homogeneity, i.e., the teacher and student network are in the same input space. However, some recent research [49] shows that heterogeneous teacher-student network architecture, where the teacher and student networks are in different spaces, also works well. Our proposed unsupervised personalization approach is also a heterogeneous teacher-student network - the first one designed with few-shot learning, to the best of our knowledge. Moreover, instead of using conventional L2 approximation, we proposed a novel random selection method to force the student networks output to approximate the teacher network's output with the advantage of no additional hyper parameters.

3. Supervised Few-shot Personalization

3.1. SAGE Model Architecture

Before discussing about our supervised few-shot personalization method, we first introduce our unpersonalized (calibration-free) model, which is named as SAGE (fa**S**t **A**ccurate **G**aze track**E**r), a new CNN model architecture that offers better accuracy, memory usage, and speed compared to prior art of 2D gaze prediction model [19, 37]. Figure 1 illustrates the SAGE model architecture.

The inputs to the SAGE model are: left/right eye images, eye landmark features, and a unique id to represent the device. The key changes in terms of the inputs and architecture as compared to iTracker [19] are summarized below.

First, we replace a crude approximation of the head pose represented by the binary face grid with more sensitive eye corner landmark features. The landmark feature is computed by concatenation of two dimensional eye corner landmarks (x, y) that amounts to a $\mathbb{R}^8$ vector for two eyes. To make the landmark feature independent from the image resolution, we normalize x and y by dividing them by image

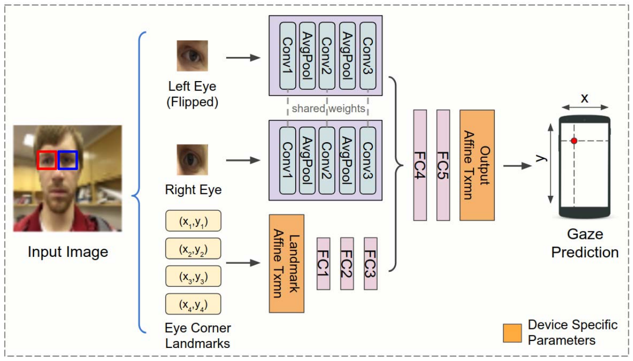

Figure 1. SAGE: Our proposed network architecture for gaze estimation.

width and height respectively.

Second, to reduce the amount of overfitting, we remove face image from the inputs, making the model rely only on the eyes and eye landmark features. Eye images exhibit much less variability compared to face images which enables better model fitting and improved generalization using lesser data. We constrain parameters of the convolutional layers to be same for both left and right eye streams similar to the iTracker model. To simplify the network's task, we flipped the left eye image left-to-right so that the weights of both eye streams can be shared more easily. Since eye images require lower resolution than the full face image, we reduce the eye image resolution to 64×64, thereby greatly improving the inference speed.

Third, to accommodate differences in the mobile phone camera's intrinsic parameters, we introduce a small set of device specific parameters: $w_{aov} \in \mathbb{R}^2$ to reflect differences in the angle of view and $w_{loc} \in \mathbb{R}^4$ to reflect differences in the camera's location with respect to the screen origin. These are applied as an affine transformation to the landmark inputs and the gaze prediction output.

Lastly, we jitter the eye bounding box and eye corner landmark locations during training. This dynamic cropping of eye regions makes the model more robust to small changes in lighting, camera movement, noise, or face/eye landmark detection errors.

We use the following sizes for the convolutional layers: $\text{CONV1}(7 \times 7 \times 32)$, $\text{CONV2}(5 \times 5 \times 32)$, $\text{CONV3}(3 \times 3 \times 32)$. The FC layers have the following sizes: $\text{FC1}(100)$, $\text{FC2}(16)$, $\text{FC3}(16)$, $\text{FC4}(16)$, $\text{FC5}(2)$.

In section 5, we thoroughly evaluate the performance of SAGE with and without the proposed personalization methods and show improvements in both accuracy and inference speed of gaze estimation over current best known methods.

3.2. Supervised Few-shot Personalization

In this section, we propose a fully-supervised on-device personalization framework that works well with only a few calibration points per user (per device orientation) and supports real-time inference. Our personalization method consists of three phases: on-server training phase to train the base model; on-device calibration phase to apply personalization for a new user; and on-device inference phase to produce gaze estimation on a new user query, as explained below.

1. **On-server training phase**: The entire training process happens offline. For each sample (query), a few labeled samples from the same person (calibration data) are used for personalization. The weights of the model is fixed after training is finished.
2. **On-device calibration phase**: Given a new user, the trained model takes a set of images with corresponding calibration point labels as input to extract user calibration features and store for later use in the inference phase. Note that the calibration step is not necessary in the inference phase and is required only once for every new user or whenever the environment factors (e.g., lighting, reflection etc.) change significantly.
3. **On-device inference phase**: The personalized model takes the query image and the cached calibration features as input to make prediction.

We denote the number of calibration points per user as K (K is a small number e.g., $K = 3$ or 5). Since the training dataset [19] contains thousands of images per user across multiple labels, we first construct a meta-dataset using randomly-generated training examples that consist of a (query image, label) pair and K corresponding (calibration point image, label) pairs belonging to the same user.

3.2.1 Model Architecture

The model architecture is illustrated in Figure 2. The query image and calibration point images are first processed to extract a feature embedding (similar to feature extraction using pre-trained models). The output feature representations are denoted by X_q, X_{c_1}, X_{c_2}, ..., X_{c_K}. Note that the feature embedding block can be swapped for any image model. In our study, we evaluate both the proposed SAGE architecture [1] (in Section 3.1) and iTracker [19] architecture after discarding the final FC layer. A 2D regression estimator is then used to compute the screen gaze location from the feature embeddings (i.e., X_q, X_{c_1}, X_{c_2}, ..., X_{c_K}) extracted in the previous step and gaze labels of the calibration points (y_{c_1}, y_{c_2}, ..., y_{c_K}).

Like most embedding based few-shot learning methods, the embedding features X_q and X_{c_j} are used to find the relationship between the query q and few-shot examples c_j, then y_q is predicted from y_{c_j} based on the learned relationship. For example, [36] employs a nearest neighbor approach, where X_{c_j}, the sample that has closest distance to

[1]When SAGE architecture is used, the layer of output affine transformation is added after the final FC layer in Figure 2.

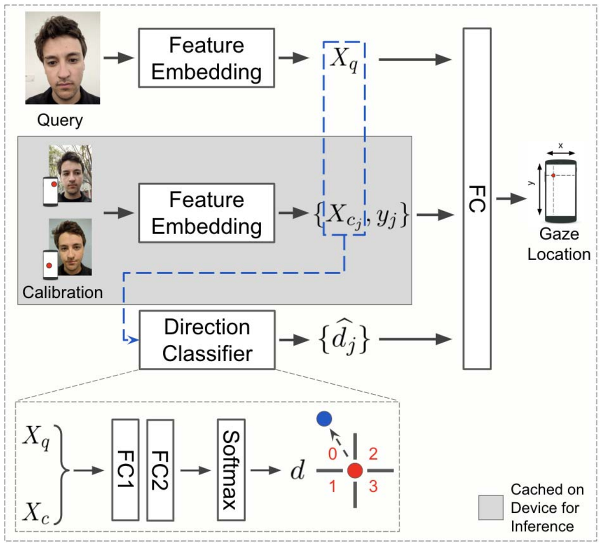

Figure 2. The proposed supervised few-shot personalization model. The figure illustrates the overall on-server training phase. X_q denotes the extracted query feature vector, X_{c_j} and y_j represent the extracted feature vector and gaze location label of the j^{th} calibration point, and $\hat{d}_j$ represents the relative direction of the query w.r.t. to the j^{th} calibration point. The gray box shows the components that can be pre-computed and cached for on-device inference. Figure best viewed in color.

X_q is first found [2], and its corresponding label y_{c_j} is then assigned to q. Unfortunately, this nearest neighbor approach is unsuitable for our regression problem, therefore, we input X_q, X_{c_j} and y_{c_j} into FC layers to learn the appropriate relationship.

Moreover, for better model robustness to noise in the labels of the calibration points, we propose employing additional information about the pairwise relative direction between the query image and calibration points. The directional relationship between calibration points and query point is robust to noise, i.e., even if there is a small noise in the 2D position of labels, the directional relationship will remain unchanged. Since labeling noise in gaze datasets is almost unavoidable, this robust relationship is useful to help train a better model.

More specifically, we first divide the screen centered at the calibration point of interest into four regions (or equivalently, four quadrants), then setup a pairwise loss for every (query, calibration) image feature pair to minimize the estimated relative gaze direction during training since we know the ground-truth query label. Hence, in addition to estimating the gaze location for the query image, we design the model to solve an additional task: given a (query, calibration) image feature pair, classify the direction of the query point with respect to the calibration point. We do this by adding a direction classifier (DC) which is composed of two FC layers with softmax, and takes pairs of (X_q, X_{c_j})

[2]For one-shot learning.

as inputs and outputs a probability vector in $\mathbb{R}^4$. To train the direction classifier, we apply a cross entropy loss on the probability output with the target being the one-hot encoded vector of the ground-truth query point direction relative to the calibration point. Feature embeddings, labels and directional outputs of all (query, calibration) pairs are concatenated and processed by the final FC layer to produce the final gaze prediction.

3.2.2 Training and Inference

We train the model end-to-end (including the feature extractor CNNs) by minimizing the following loss function:

$$\mathcal{L}(W) = \sum_{i=1}^{N} \|\phi(q^i, \{c^i_j, y^i_j\}_{j=1\dots K} \mid W) - y^i_q\|^2 \\ -\lambda \sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{d=0}^{3} 1(\hat{d}^i_j = d^i_j) \log\left(P(\hat{d}^i_j \mid W)\right),$$

where N is the number of training examples, K is the number of calibration points, W are the trainable parameters in the network, 1 is the indicator function. For the i-th training example, q^i represents the query inputs to the CNN model, c^i_j is the j^{th} calibration point in this example, y^i_q is the query ground-truth gaze location, y^i_j is the groundtruth gaze location for j^{th} calibration point, and $\phi(\cdot)$ is the output gaze location of our network. d^i_j is the ground-truth relative direction of q^i with respect to the j^{th} calibration point c^i_j, $\hat{d}^i_j$ is the corresponding estimated relative gaze direction, and $P(\cdot)$ is the output probability of direction classifier. λ is standard weighting parameters.

Note that in a CNN-based model, computational cost often resides in the convolutional layer. To speed up on-device inference, we propose to compute the feature embedding from the calibration data on device once after data collection and then cache it for future inference.

4. Unsupervised Few-shot Personalization

Getting access to user's labeled data is sometimes difficult due to users being reluctant to perform the calibration step or not following the instructions correctly. On the other hand, getting user's unlabeled data (i.e., their facial images) without asking them to perform the calibration task is much easier. Our hypothesis is that even when gaze labels are unavailable, the facial images still contain valuable information about the user that can be useful for gaze estimation. We refer to this scenario as "unsupervised personalization" since gaze labels are not provided for the facial images used for personalization. To the best of our knowledge, unsupervised personalization has not been discussed before in eye tracking research.

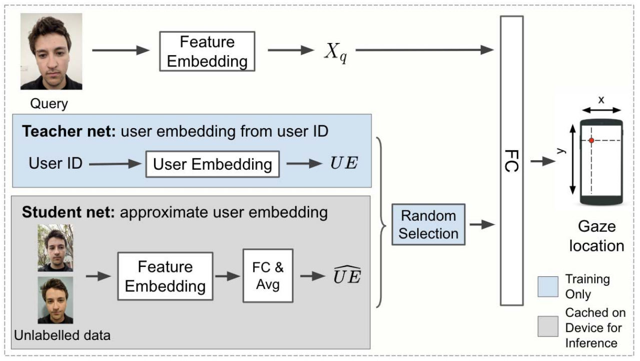

Figure 3. Our proposed unsupervised few-shot personalization method based on teacher-student network. We use the following notations: X_q represents the query feature vector, UE represent the user embedding from the teacher network, and $\hat{UE}$ is the approximated user embedding produced by the student network using a few unlabeled images. The gray box shows the components that can be pre-computed and cached for on-device inference. The blue box shows the components that are needed only during the training phase, and hence unnecessary for inference. Figure best viewed in color.

The key idea in our approach is to adopt a heterogeneous teacher-student network architecture where the student network uses few unlabeled images to learn from a teacher network that has some personal information of the user from other information sources (than unlabeled images). Figure 3 describes an overview of the unsupervised personalization approach. Although different teacher networks are applicable, in this work we focus on using a user embedding (in our experiments, the vector is $\mathbb{R}^{16}$) that represents a user's personal properties as the teacher network.

The user id embedding of the teacher network is similar to the word embedding in language models except that it is a supervised embedding learnt using the gaze labels. Like in word embeddings, by providing user id as an input, the teacher network outputs an embedding vector. We concatenate this user-specific embedding vector with the extracted features from the query image, and then train the network to predict the final gaze estimation. This allows the network to learn a meaningful embedding vector to represent the parameters of each user which helps improve the prediction accuracy. Note that on average each user has 1000 samples in the training set which is large enough to learn a meaningful embedding to represent personal gaze properties. Since the user embedding is unavailable for new users, we attempt to learn/approximate it via a student network using a few unlabeled images from the user. The resulting student network can then be applied to any new user. The underlying assumption here is that some personal gaze properties can be represented using (and hence learned from) facial images. This assumption is reasonable, considering gaze properties will be affected by glasses, eye shapes, etc.

For the student network, features are first extracted from user images and mapped to an embedding space using the penultimate FC layer of the SAGE CNN model (described in Section 3.1). As each user image produces an approximate user embedding, these vectors are combined using a simple averaging layer to make the embedding more robust. To make the student net approximate the teacher net, we propose using a random uniform selection mechanism during training by which in each training step the gradient update uses either the student net or the teacher's user embedding with equal probability. This random selection tries to force the student net's output to be close to the teacher net's output. Ideally, when both teacher net and student net converge, if the two provide exactly the same output there will be no gradient penalization when switching from teacher net to student net. We also experimented with $\mathbb{L}_2$ loss between the outputs of the teacher and student net and obtained similar results which is discussed in Section 5.6, but $\mathbb{L}_2$ loss approach will need to tune one more hyper parameter of loss weights.

The user embedding and the extracted query features are concatenated and passed through the final FC layers to predict the gaze location. The model is trained using a standard mean squared error loss function between the estimated 2D gaze location and ground-truth labels.

The proposed unsupervised approach can also benefit from the caching technique described in Section 3.2.2 for faster on-device inference. In the inference phase, the teacher network is discarded and only the cached student network output will be fed to the model along with the query image features.

5. Experiments

In this section, we test the effectiveness of the proposed personalization methods via comprehensive evaluation on a large-scale gaze dataset.

5.1. Setup

Data preparation: We use GazeCapture dataset [19] which has gaze data collected from 1474 subjects on iPhones and iPads. The subjects were asked to look at dots appearing at random locations on the screen while their face images were recorded using the front facing camera. The dataset consists of 1.5M frames with both face and two eyes being visible. We adopted the same train/test split as [19] which holds out 150 subjects for testing and the rest for training. More details regarding the dataset can be found in section 5.1 in [19].

For training our personalized model, each input image is coupled with a set of labeled images from the same person to mimic the calibration procedure. The calibration images are chosen randomly from the set of all images that belong to a person in the training partition. To reproduce the results of the personalized-SVR method in [19], calibration images are selected from a fixed gaze location.[3] We also run experiments by randomly selecting calibration images for SVR for further fair comparison.

Evaluation Metric: We report error by measuring the mean Euclidean (ME) distance between the ground-truth and the estimated gaze location and list results for phone and tablet form-factor devices separately.

Implementation details: All of the models (including baselines, SAGE and their personalized variants) are trained from scratch up to 100K iterations using Adam optimizer. We use a batch size 128 for iTracker and its personalized models, 256 for the Full-face model and a batch size 512 for SAGE and its variants. We initially set the learning rate to 0.006 and divide it by 3 every 20K iterations. The weighting parameter λ is empirically chosen as 1.0 in our experiments. We implement all CNN models using TensorFlow.

5.2. SAGE vs. Prior art

We first evaluate the calibration-free base models (i.e. models without personalization) and compare the proposed SAGE architecture with approaches iTracker [19] and Full-face model [37] as baselines.

Our proposed SAGE architecture outperforms the iTracker and Full-face model on both phones and tablets. For phones, the mean error is reduced from 1.94cm (iTracker) and 2.14cm (Full-face) to 1.78 cm. For tablets, it is reduced from 3.0cm (iTracker) and 3.51cm (Full-face) to 2.72cm.

5.3. Supervised Few-shot On-device Personalization

We evaluate the proposed Supervised Few-shot On-device (SFO) method by applying it to two different models. Since iTracker performed better than the Full-face model for the GazeCapture dataset, we focus remaining analyses on comparing the effect of personalization on the SAGE and iTracker models. We also compare the results with the existing personalization method based on SVR (iTracker-SVR) [19].

Table 2 shows that applying SFO to the SAGE model reduces error by 13% for phones and 10% for tablets by using as few as two calibration points. Using 3-9 calibration points further improves the model performance by 17-23% for phones. Similarly, applying SFO to the iTracker model reduces the mean error by 8% for phones using only 2 calibration points, 12% using 3 points, and 21% for 9 points.

We also compare SFO with SVR [19] [4]. As shown in

[3]For example, 4-point calibration means sampling images when the subject were looking at 4 corners on the screen; 5-point calibration means 4 corners plus the center; etc.

[4]Note that the accuracy of our implementation on personalized SVR is different from what was reported in [19] because we removed test augmentation which increases the inference time by $\sim$20 times and hence is not applicable for on-device applications.

Model	# of pts	Phone (ME in cm)	Tablet (ME in cm)
iTracker	0	1.94	3.02
iTracker-SFO	2	1.78 (-8.25%)	2.88 (-4.64%)
	3	1.70 (-12.37%)	2.69 (-10.93%)
	5	1.58 (-18.56%)	2.61 (-13.58%)
	9	**1.53 (-21.13%)**	**2.48 (-17.88%)**

Table 1. Results of applying the proposed supervised few-shot on-device personalization (SFO) to the iTracker model. Error is shown in cm, and the % reduction in error compared to the base model (0 calibration points) is reported within parenthesis. "# of pts" refers to the number of calibration points.

Model	# of pts	Phone (ME in cm)	Tablet (ME in cm)
SAGE	0	1.78	2.72
SAGE-SFO	2	1.55 (-12.92%)	2.44 (-10.29%)
	3	1.47 (-17.42%)	2.31 (-15.07%)
	5	1.43 (-19.66%)	2.25 (-17.28%)
	9	**1.37 (-23.03%)**	**2.10 (-22.79%)**

Table 2. Similar to Table 1. Shows results of applying the proposed supervised few-shot on-device personalization (SFO) to the SAGE model.

Model	# of imgs	Phone (ME in cm)	Tablet (ME in cm)
iTracker	0	1.94	3.02
iTracker-UFO	9	1.84 (-5.15%)	2.79 (-7.62%)
SAGE	0	1.78	2.72
SAGE-UFO	9	1.72 (-3.37%)	2.57 (-5.51%)

Table 3. Results of applying the proposed unsupervised few-shot on-device method (UFO) to the iTracker and SAGE models. Error is shown in cm (and % reduction in error within parenthesis). "# of imgs" represents number of facial images per user.

Figure 4, iTracker-SVR requires a large number of calibration points to show a substantial improvement in accuracy over iTracker (13 points for ∼8% error reduction for phones). For <5 points, SVR actually performs worse than the base model. This is due to the limited generalization ability of SVR. In comparison, our proposed supervised few-shot personalization approach can easily benefit from only a few calibration points.

5.4. Unsupervised Few-shot On-device Personalization

In this section, we evaluate the proposed Unsupervised Few-shot On-device (UFO) method by applying it to the SAGE (SAGE-UFO) and iTracker (iTracker-UFO) models (note that iTracker-SVR is not applicable in an unsupervised setting). As shown in Table 3, using 9 unlabeled facial images helped reduce the mean error by 5-8% for the iTracker model (and 3-6% for the SAGE model) albeit to a lesser extent than the supervised method. To the best of our knowledge, this is the first attempt to use unsupervised personalization in eye tracking research, hence there are no prior baselines to compare against.

5.5. MPIIFaceGaze dataset evaluation

To further demonstrate the effectiveness of our proposed personalization methods, we evaluate one of the models (iTracker with and without personalization) on MPIIFaceGaze [37, 48], another benchmark dataset for unconstrained appearance-based gaze estimation in recent years. MPIIFaceGaze dataset contains about 36000 images collected using laptops from 15 participants. In our experiments, we used the data from 10 (random) participants for training, and the rest for evaluation. We compare the iTracker-SFO model with the iTracker-SVR and calibration-free iTracker models and the results are shown in Figure 5. We find similar improvements in the mean error on laptops as with the phone/tablet error reported on the GazeCapture dataset (see Figure 4).

5.6. Ablation study

To further justify the design choices in our proposed networks, we conducted ablation studies for several network components, including direction classifier, teacher net, and random selection approach.

Ablating the direction classifier (for the iTracker-SFO model) increases the mean error on phones from 1.82 to 1.88cm, 1.70 to 1.75cm, and 1.53 to 1.61cm with 1, 3, 9 calibration points respectively.

Ablating the teacher net increases model error from 1.84cm to 1.90cm for phones (from 2.79cm to 3.00cm for tablets), using 9 calibration points with the iTracker-UFO

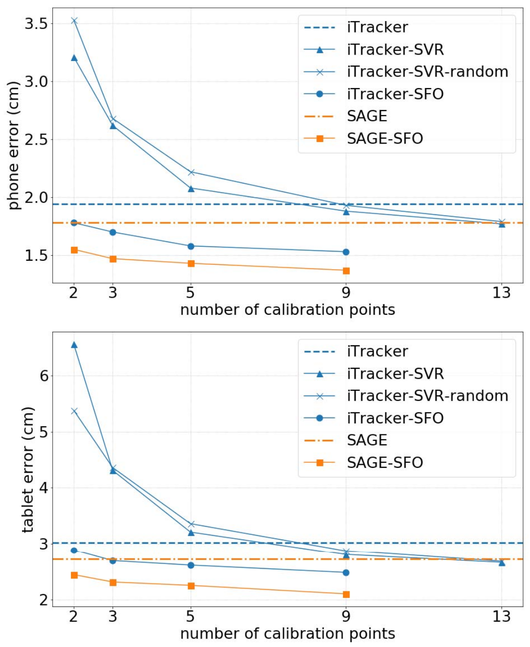

Figure 4. Performance comparison between SVR [19] and SFO. We implement and evaluate SVR in 2 settings: calibration points are collected (1) from 13 fixed locations (*iTracker-SVR*), the same as in [19], or (2) at random locations (*iTracker-SVR-random*), which is similar to our proposed approach. For any number of calibration points, our method significantly outperforms personalized SVR.

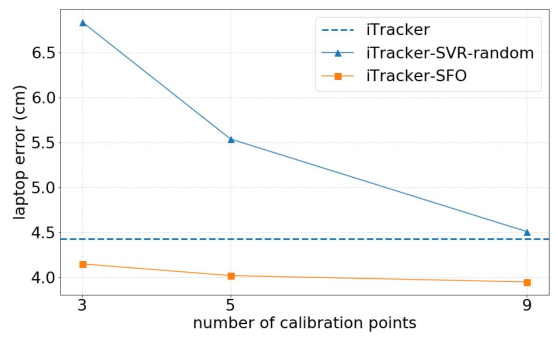

Figure 5. Similar to Figure 4. Shows results of applying the baseline SVR and our proposed SFO personalization methods on top of iTracker[19] to the MPIIFaceGaze dataset [37, 48]. Both methods choose calibration points randomly.

Model	# of pts		
	1	3	9
iTracker	933	933	933
iTracker-SVR	933	933	933
iTracker-SFO (no cache)	2,798	6,529	17,722
iTracker-SFO (cache)	933	933	932
SAGE	42	42	42

Table 4. Inference speed represented by the number of millions of FLOPs for iTracker-SFO and SAGE-SFO models, compared to baseline iTracker and iTracker-SVR. "# of pts" represents the number of calibration points used.

model.

Additional experiments comparing the random selection method for teacher-student network training to minimizing the $\mathbb{L}_2$ loss between the teacher and student net outputs showed similar results. For instance, using the SAGE-UFO method and 9 unlabeled images, random selection achieves a phone ME of 1.71cm (2.60cm for tablet) while the $\mathbb{L}_2$ loss minimization strategy achieves 1.72cm (2.57cm for tablet). However, note that the random selection method does not need tune any hyper parameter, while $\mathbb{L}_2$ loss needs to tune the loss weight hyper parameter.

5.7. Run-time Performance

We run the different models on a mobile device and compare the computational cost during inference in millions of floating point operations (FLOPs). Our SAGE-SFO method runs significantly faster compared to iTracker-SVR [19] – inference takes around 41.8×10^6 FLOPs ($\sim$20x faster than iTracker), and takes 10ms on a Pixel 2 phone. Thus, the proposed model is well suited for applications that require real-time gaze. It is worth noting that pre-computing and caching the calibration image embeddings ahead of time significantly reduces the number of FLOPs during inference (e.g., assuming 9 calibration points, SAGE-SFO without caching requires $\sim$10x more FLOPs than with caching).

Model	# of pts	Phone (ME in cm)	Tablet (ME in cm)
iTracker	0	1.94	3.02
iTracker-SVR	13	1.77 (-8.76%)	2.66 (-11.92%)
iTracker-SFO	3	1.70 (-12.37%)	2.69 (-10.93%)
iTracker-SFO	9	1.53 (-21.13%)	2.48 (-17.88%)
iTracker-UFO	9	1.84 (-5.15%)	2.79 (-7.62%)
SAGE	0	1.78 (-8.25%)	2.72 (-9.93%)
SAGE-SFO	3	1.47 (-24.22%)	2.31 (-23.51%)
SAGE-SFO	9	**1.37 (-29.38%)**	**2.10 (-30.5%)**
SAGE-UFO	9	1.72 (-11.34%)	2.57 (-14.9%)

Table 5. This table summarizes the performance of the different personalized models, along with the number of calibration points required.

5.8. Discussion

The proposed personalization methods, both supervised and unsupervised, can be applied to any base model architecture as demonstrated here with the SAGE and iTracker models. Table 5 summarizes the performance of the different personalized models and their improvement compared to the best calibration-free model (iTracker) as baseline.

Applying the proposed supervised few-shot personalization method (SFO) to the iTracker model achieves better accuracy with fewer calibration points as compared to using Support Vector Regression method (SVR) (12-21% error reduction using 3-9 calibration points for iTracker-SFO, compared to lower error reduction of 8.7% using 13 points for iTracker-SVR on phones). Even in the absence of gaze labels, applying the unsupervised few-shot personalization method (UFO), which only uses unlabeled facial images from the user, to the iTracker model as base, improves the accuracy compared to iTracker (5.15% error reduction for phones, and 7.62% for tablets).

The SAGE model is an improved CNN architecture that outperforms iTracker by 8.2% in terms of accuracy. When the supervised SFO method is applied to the SAGE model, it achieves large improvements in accuracy of 24.2% compared to iTracker, using only 3 calibration points. Overall, the best performance is achieved by the SAGE-SFO model using 9 calibration points (29.4% lower error than iTracker, and 22.6% lower error than iTracker-SVR).

6. Conclusion

In this paper, we proposed a novel few-shot personalized gaze estimation model that is accurate, fast and can run entirely on-device. In particular, the proposed CNN-based model architecture, SAGE, combined with our supervised few-shot personalization method yields bigger improvements in accuracy of $\sim$24% using only 3 calibration points per user and runs $\sim$20x faster compared to the previous best models. We believe that the improved accuracy and speed will enable real-time gaze applications for accessibility, gaming and human-computer-interaction.

References

[1] Shumeet Baluja and Dean Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In *Advances in Neural Information Processing Systems*, pages 753–760, 1994.

[2] Ernesto Brau, Jinyan Guan, Tanya Jeffries, and Kobus Barnard. Multiple-gaze geometry: Inferring novel 3d locations from gazes observed in monocular video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 612–630, 2018.

[3] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Troster. Eye movement analysis for activity recognition using electrooculography. *IEEE transactions on pattern analysis and machine intelligence*, 33(4):741–753, 2011.

[4] Jixu Chen and Qiang Ji. 3d gaze estimation with a single camera without ir illumination. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[5] Jixu Chen and Qiang Ji. Probabilistic gaze estimation without active personal calibration. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 609–616. IEEE, 2011.

[6] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 100–115, 2018.

[7] Peter M Corcoran, Florin Nanu, Stefan Petrescu, and Petronel Bigioi. Real-time eye gaze tracking for gaming design and consumer electronics systems. *IEEE Transactions on Consumer Electronics*, 58(2), 2012.

[8] Tobias Fischer, Hyung Jin Chang, and Yiannis 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.

[9] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

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

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[12] Philip S Holzman, Leonard R Proctor, Deborah L Levy, Nicholas J Yasillo, Herbert Y Meltzer, and Stephen W Hurt. Eye-tracking dysfunctions in schizophrenic patients and their relatives. *Archives of general psychiatry*, 31(2):143–151, 1974.

[13] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: A dataset and baseline algorithms for unconstrained appearance-based gaze estimation in mobile tablets. *CoRR*, abs/1508.01244, 2015.

[14] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28(5-6):445–461, 2017.

[15] Takahiro Ishikawa, Simon Baker, Iain Matthews, and Takeo Kanade. Passive driver gaze tracking with active appearance models. Technical Report CMU-RI-TR-04-08, Pittsburgh, PA, February 2004.

[16] RJ Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4, 2003.

[17] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[18] Jedrzej Kozerawski and Matthew Turk. Clear: Cumulative learning for one-shot one-class image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2018.

[19] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2176–2184, 2016.

[20] Erik Lindén, Jonas Sjöstrand, and Alexandre Proutiere. Appearance-based 3d gaze estimation with personal calibration. *arXiv preprint arXiv:1807.00664*, 2018.

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

[22] Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32(3):169–179, 2014.

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

[24] Carlos H Morimoto and Marcio RM Mimica. Eye gaze tracking techniques for interactive applications. *Computer vision and image understanding*, 98(1):4–24, 2005.

[25] Bilge Mutlu, Toshiyuki Shiwa, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 61–68. ACM, 2009.

[26] Atsushi Nakazawa and Christian Nitschke. Point of gaze estimation through corneal surface reflection in an active illumination environment. *Computer Vision–ECCV 2012*, pages 159–172, 2012.

[27] Dim P Papadopoulos, Alasdair DF Clarke, Frank Keller, and Vittorio Ferrari. Training object class detectors from eye tracking data. In *European Conference on Computer Vision*, pages 361–376. Springer, 2014.

[28] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 721–738, 2018.

[29] Hang Qi, Matthew Brown, and David G Lowe. Learning with imprinted weights. *arXiv preprint arXiv:1712.07136*, 2017.

[30] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[31] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[32] Hosnieh Sattar, Sabine Muller, Mario Fritz, and Andreas Bulling. Prediction of search targets from fixations in open-world settings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 981–990, 2015.

[33] Timo Schneider, Boris Schauerte, and Rainer Stiefelhagen. Manifold alignment for person independent appearance-based gaze estimation. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1167–1172. IEEE, 2014.

[34] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[35] Brian A. Smith, Qi Yin, Steven K. Feiner, and Shree K. Nayar. Gaze locking: Passive eye contact detection for human-object interaction. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, 2013.

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

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

[38] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[39] Kang Wang and Qiang Ji. Real time eye gaze tracking with 3d deformable eye-face model. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[41] Kang Wang, Rui Zhao, and Qiang Ji. A hierarchical generative model for eye image synthesis and eye gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 440–448, 2018.

[42] Kang Wang, Rui Zhao, Hui Su, and Qiang Ji. Generalizing eye tracking with bayesian adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11907–11916, 2019.

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

[44] Yunyang Xiong, Hyunwoo J Kim, and Vikas 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.

[45] Yu Yu, Gang Liu, and Jean-Marc Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. *arXiv preprint arXiv:1904.10638*, 2019.

[46] Xucong Zhang, Michael Xuelin Huang, Yusuke Sugano, and Andreas Bulling. Training person-specific gaze estimators from user interactions with multiple devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 624. ACM, 2018.

[47] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2015.

[48] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):162–175, 2017.

[49] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.