# Eye Know You: Metric Learning for End-to-end Biometric Authentication Using Eye Movements from a Longitudinal Dataset

Dillon Lohr, Henry Griffith, and Oleg V Komogortsev

***Abstract*—The permanence of eye movements as a biometric modality remains largely unexplored in the literature. The present study addresses this limitation by evaluating a novel exponentially-dilated convolutional neural network for eye movement authentication using a recently proposed longitudinal dataset known as GazeBase. The network is trained using multi-similarity loss, which directly enables the enrollment and authentication of out-of-sample users. In addition, this study includes an exhaustive analysis of the effects of evaluating on various tasks and downsampling from 1000 Hz to several lower sampling rates. Our results reveal that reasonable authentication accuracy may be achieved even during both a low-cognitive-load task and at low sampling rates. Moreover, we find that eye movements are quite resilient against template aging after as long as 3 years.**

***Index Terms*—Eye movements, biometric authentication, metric learning, template aging, dilated convolution**

## I. Introduction

EYE movement biometrics have received considerable attention in the literature over the past two decades [1]. This focus is motivated by the specificity and permanence of human eye movements [2]. Eye movement biometric systems offer notable advantages over alternative modalities, including the ability to support liveness detection [3], [4] and spoof-resistant continuous authentication [5]. Eye movements are also well suited for integration within multimodal biometric systems [6].

Despite the considerable literature within this domain, several improvements are necessary to advance the large-scale commercial viability of this technology. For example, most existing literature formulates eye movement biometrics as a closed-set classification problem [7]–[10]. This approach is problematic for real-world scenarios in which new users must be continuously enrolled and authenticated.

Moreover, the existing knowledge base is further limited by the validation of the proposed models on a variety of diverse datasets, many of which are characterized by short-term test-retest intervals [8], [9], [11], [12]. This lack of a suitable gold-standard validation set with sufficient temporal duration limits both comparability between results and the assessment of template aging effects. Finally, although eye tracking sensors within emerging commercial devices often are characterized by limited temporal precision and deployment in low resource environments, few studies have explored performance variability versus signal sampling rate, nor the capacity to reduce the number of model parameters to support deployment in embedded environments.

The research described herein attempts to address many of the aforementioned limitations. We train an exponentially-dilated convolutional neural network (CNN) that learns meaningful embeddings via multi-similarity (MS) loss [13]. Inputs consist of fixed-length subsequences of eye movements during various tasks, including reading, tracking jumping dots, watching videos, and playing an interactive game. Similarity scores are measured as the mean cosine similarity between temporally-aligned subsequence embeddings. The proposed technique is verified on several tasks from the GazeBase dataset [14], which consists of 322 participants recorded up to 18 times each over a 37-month period. We also compare against a statistical baseline and the current state-of-the-art, DeepEyedentificationLive (DEL) [4].

The main contributions of this study are:

- The development of an exponentially-dilated convolutional neural network model offering state-of-the-art performance with 440 times fewer learnable parameters.
- The initial demonstration of multi-similarity loss in a metric learning framework for eye movement biometrics.
- The most thorough assessment of eye movement permanence to date, with reasonable authentication performance demonstrated for a 37-month test-retest interval.
- The most thorough assessment of task dependence to date, with comparable authentication performance achieved for a low-cognitive-load task (i.e., jumping dot stimulus) versus traditionally recommended high-cognitive-load tasks (e.g., reading [15] or visual search [9]).

## II. Prior Work

Since the introduction of eye movements as a biometric in 2004 [8], significant research has focused on improving their viability. A collective review of related work published prior to 2015 may be found in [20]. Moreover, comparative results for studies analyzing common datasets are provided in [21], which summarizes the results of the most recent BioEye competition. As noted within these reviews, the majority of prior work uses a common processing pipeline, with the recordings initially partitioned into specific eye movement events using a classification algorithm, followed by the formation of the biometric

The authors are with the Department of Computer Science, Texas State University, San Marcos, TX, 78666 USA (e-mail: djl70@txstate.edu; hkgriffith1@gmail.com; ok11@txstate.edu).

Manuscript received April 19, 2005; revised August 26, 2015.

TABLE I
A SUMMARY OF THE METHODOLOGICAL ASPECTS OF SELECTED WORKS. N IS THE NUMBER OF SUBJECTS EMPLOYED WHEN MEASURING PERFORMANCE; THE FULL SIZE OF THE DATASET MAY HAVE BEEN LARGER. ST MEANS SHORT-TERM AND LT MEANS LONG-TERM.
*: DATASET WAS PREVIOUSLY PUBLIC BUT IS UNAVAILABLE AT THE TIME OF WRITING.
**: A MODIFIED VERSION OR SUBSET OF THE DATASET IS PUBLICLY AVAILABLE.

| Study | Year | Open-set? | Tasks | Sampling rates (Hz) | N subjects | Test-retest interval | Public dataset? |
|---|---|---|---|---|---|---|---|
| [8] | 2004 | N | Jumping dot | 250 | 9 | same day | N |
| [7] | 2015 | N | Reading; jumping dot | 250 | 76–77 (ST); 18–19 (LT) | 30 min. (ST); 1 year (LT) | N* |
| [16] | 2017 | Y | Reading | 1000 | 149 (ST); 34 (LT) | med. 19 min. (ST); med. 11.1 months (LT) | Y** [14] |
| [9] | 2018 | N | Visual search | 300, 150, 75, 30 | 58 | same day (ST); avg. 18 days (LT) | N |
| [10] | 2018 | N | Image viewing | 500 | 32 | $\leq$ 30 min. | N |
| [11] | 2019 | Y | Video viewing | 30 | 105 | same day | Y [17] |
| [18] | 2020 | Y | Reading (ST); jumping dot (LT) | 1000 | 25 (ST); 10 (LT) | same day (ST); 2–8 weeks (LT) | Y** [18] |
| [4] | 2020 | Y | Jumping dot | 1000 | 25 | $\geq$ 1–4 weeks | Y [4] |
| [19] | 2020 | Y | Jumping dot | 1000, 500, 250, 125, 62, 31 | 25 | $\geq$ 1–4 weeks | Y [4] |
| [12] | 2020 | Y | Reading | 1000 | 67–68 | avg. 20 min. | Y** [14] |
| Present | 2021 | Y | Reading; jumping dot; static dot; video viewing; interactive game | 1000 (all tasks); 500, 250, 125, 50, 31.25 (reading only) | 14–59 | 20 min. to 37 months | Y [14] |

template as a vector of discrete features from each event. One problem with such approaches is that event classification is a difficult problem [22], so it adds another layer of complexity that influences biometric performance. Only recently have studies begun utilizing end-to-end deep learning workflows [10], [11].

The winners of the BioEye 2015 competition, George & Routray [7], used a radial basis function (RBF) network for computing similarities between probe and gallery vectors. Features describing the position, velocity, and acceleration for fixations and saccades were extracted from the segmented signal. The algorithm was validated using a dataset of 153 individuals recorded twice during both a reading task (TEX) and a random saccades task (RAN) with 30 minutes between recording sessions and recorded again after one year. They achieved an equal error rate (EER) of 2.59% for RAN and 3.78% for TEX when the recording sessions were separated by 30 minutes. When the recording sessions were separated by one year, they achieved 10.96% EER for RAN and 9.36% for TEX. As the proposed method requires retraining the network upon the enrollment of each new user, it is not feasible for large-scale practical deployment.

In addition to eye movement-specific features, other representations of eye movement recordings have also been explored in the literature. For example, Li et al. [9] used a multi-channel Gabor wavelet transform (GWT) to extract texture features from eye movement trajectories during a visual search task. Support vector machine (SVM) classifiers were used for biometric identification and verification. Results were verified using a dataset consisting of 58 subjects recorded across several trials, with a minimum EER of 0.89% reported. Texture-based eye movement features were recently reinvestigated in [23], where downsampling of the filtered images was proposed for the feature extraction step in order to preserve spatial structure. In addition to the aforementioned restriction regarding new user enrollment, both of these studies utilized recordings with only a small temporal separation.

Jia et al. [10] introduced deep learning techniques for eye movement biometrics. A recurrent neural network (RNN) was built using long short-term memory (LSTM) cells. The output layer used softmax to produce class probabilities. Their approach was validated using a dataset of 32 subjects recorded across several trials of a high-cognitive-load task, with a minimum EER of 0.85% reported. This study did not explore its method's long-term efficacy, as recordings for each subject were collected during a single, 30-minute period.

Friedman et al. [16] employed a statistical approach for eye movement biometrics. A novel event classification algorithm, the modified Nyström and Holmqvist (MNH) algorithm [24], was used to classify several types of events. A set of over 1,000 features [25] was extracted from each recording. This approach was validated using a subset of the dataset considered herein, consisting of 298 subjects recorded twice each during a reading task. Using data separated by approximately 20 minutes, a best-case EER of 2.01% was reported. With data separated by approximately 11 months from a set of 68 subjects, EER increased to 10.16%.

Jäger et al. [18] utilized involuntary micro eye movements for biometric authentication and identification. Raw eye movement signals were initially transformed to isolate desired micro eye movements according to their characteristic velocities, with the resulting scaled values fed into a CNN with two separate subnets. The approach was validated using two datasets (75 subjects during a reading task recorded at 1000 Hz [26], and a newly recorded dataset consisting of 10 users). This approach was later extended into DeepEyeden-

tificationLive (DEL) [4] to include liveness detection and was evaluated on a different dataset of 150 subjects, the JuDo1000 dataset [4], which is publicly available. However, the EERs presented in the later study were based on only 25 identities, and the recordings were collected with a relatively short temporal separation. DEL was also evaluated on temporally- and spatially-degraded signals by Prasse et al. [19].

Abdelwahab & Landwehr [11] introduced metric learning to the eye movement biometrics literature using deep distributional embeddings. Namely, sequences of six-dimensional vectors (binocular gaze and pupil data) at 30 Hz were fed to a deep neural network which produced distributional embeddings using a Wasserstein distance metric. The approach was validated on the publicly-available Dynamic Images and Eye Movements (DIEM) dataset [17], which contains eye movement data of 210 subjects viewing various video clips (sports, movie trailers, etc.). The recordings in the DIEM dataset were collected with only a small temporal separation.

Lohr et al. [12] also explored the use of metric learning for eye movement biometrics. Eye movement recordings were segmented into fixations, saccades, and PSOs using the MNH algorithm [24], and discrete feature vectors were extracted from each event. Three separate multilayer perceptrons (MLPs), one for each of the 3 event types, were trained on these feature vectors with triplet loss [27] to create meaningful embeddings. Distances were computed for each event type separately and then fused with a weighted sum. The approach was validated using a dataset of 269 subjects recorded twice each during a reading task. An average EER of 6.29% was reported for recordings separated by approximately 20 minutes. Like most prior studies, the permanence of eye movements was not explored.

The technique described herein expands upon the work of Lohr et al. [12] by feeding recordings directly into the model (removing the additional complexity of event classification). Additionally, the more sophisticated MS loss [13] is used, a single exponentially-dilated CNN is trained rather than multiple event-specific MLPs, and performance is evaluated on a longitudinal dataset collected over a 37-month period. The present study also explores the authentication performance of additional tasks other than reading and of downsampled eye movement signals.

## III. Methodology

### A. Dataset

We used the GazeBase [14] dataset available on Figshare [28]. This dataset consists of 322 college-aged subjects, each recorded monocularly (left eye only) at 1000 Hz with an EyeLink 1000 eye tracker. Nine rounds of recordings (R1–9) were captured over a period of 37 months, thereby enabling the analysis of template aging. Each subsequent round comprises a subset of subjects from the preceding round (with one exception, subject 76, who was absent from R3 but returned for R4–5), with only 14 of the initial 322 subjects present across all 9 rounds. Each round consists of 2 recording sessions separated by approximately 30 minutes, totaling 18 recording sessions. Recordings contain the horizontal and vertical components of the left eye's gaze position in terms of degrees of the visual angle. In each recording session, every subject performed a series of 7 eye movement tasks: a horizontal saccades task (HSS), a video-viewing task (VD1), a fixation task (FXS), a random saccades task (RAN), a reading task (TEX), a ball-popping task (BLG), and another video-viewing task (VD2). More details for each task can be found in [14]. Since VD2 was similar to VD1, we only used VD1 in our experiments.

### B. Training and testing splits

The subjects in the dataset were split in the following manner. First, we created a held-out test set using all recordings from the 59 subjects that were present in R6. The test set contained nearly 50% of all recordings in GazeBase. The test set was only used at the very end of our experiments to get a final, unbiased measure of our models' performance.

Next, we split the remaining subjects into 4 folds (which we will label F1–4) for cross-validation. We had three goals when balancing the folds, keeping in mind that some subjects have more recordings than others: (1) each fold should have a similar number of subjects, (2) each fold should have a similar number of recordings, and (3) the method to create the folds should be deterministic to facilitate reproducibility.

We accomplished these goals by using two priority queues (heaps)—one for the folds and the other for the subjects—and iteratively assigning subjects to folds. Each fold was weighted first by the number of subjects assigned to it and second by the total number of recordings present for those subjects, and the fold with the lowest weight was given the highest priority. Each subject was weighted by the total number of recordings present for that subject, and the subject with the highest weight was given the highest priority. In case of ties, an arbitrary-but-deterministic element was given higher priority. At each iteration, we extracted the highest priority element from both heaps and assigned the chosen subject to the chosen fold. The chosen fold was then placed back onto the heap with its updated priority. This process was repeated until the subject heap was empty. In the end, the largest fold had at most 1 more subject than the smallest fold, and the number of recordings present in each fold was as balanced as possible.

We ran three sets of experiments: (1) compare our metric learning model against three baseline models, using data from TEX and tuning hyperparameters based on the average performance across the 4 folds; (2) use data from the other individual tasks to assess our model's performance on types of eye movements other than reading; and (3) downsample the TEX data to assess our model's performance on signals with lower sampling rates.

### C. Signal pre-processing

We start with a sequence of $T$ tuples $(t^{(i)}, x^{(i)}, y^{(i)})$, $i = 1, \ldots, T$, where $t^{(i)}$ is the time stamp (s) and $x^{(i)}, y^{(i)}$ are the horizontal and vertical components of the monocular (left

eye) gaze position (°). Next, we compute per-channel velocity (°/s) using the one-sample backward difference method:

$$\delta_x^{(i)} = \frac{x^{(i)} - x^{(i-1)}}{t^{(i)} - t^{(i-1)}},\ i = 2, \ldots, n \tag{1}$$

$$\delta_y^{(i)} = \frac{y^{(i)} - y^{(i-1)}}{t^{(i)} - t^{(i-1)}},\ i = 2, \ldots, n. \tag{2}$$

Then, we replace any NaN velocities with 0 and clip both $\delta_x^{(i)}$ and $\delta_y^{(i)}$ within the range $[-1000, 1000]$ to minimize the influence of outliers.

During some preliminary experiments, we found that the slow and fast velocity transformations used in the DeepEye-dentification line of work [4], [18], [19] indeed led to better results on the validation set compared to z-score transforming raw velocity values. Therefore, we extracted the following 4 values for each $(\delta_x^{(i)}, \delta_y^{(i)})$ tuple:

$$x_{\text{slow}}^{(i)} = \tanh(c\delta_x^{(i)}) \tag{3}$$

$$y_{\text{slow}}^{(i)} = \tanh(c\delta_y^{(i)}) \tag{4}$$

$$x_{\text{fast}}^{(i)} = \begin{cases} z_x(\delta_x^{(i)}), & \text{if } \sqrt{{\delta_x^{(i)}}^2 + {\delta_y^{(i)}}^2} \geq v \\ z_x(0), & \text{otherwise} \end{cases} \tag{5}$$

$$y_{\text{fast}}^{(i)} = \begin{cases} z_y(\delta_y^{(i)}), & \text{if } \sqrt{{\delta_x^{(i)}}^2 + {\delta_y^{(i)}}^2} \geq v \\ z_y(0), & \text{otherwise} \end{cases}, \tag{6}$$

where $c = 0.02$ and $v = 40$ are fixed hyperparameters taken from [4], and $z_x$ and $z_y$ are separate z-score transformations for each velocity channel. The mean and standard deviation for $z_x$ and $z_y$ were computed across all velocities in the train set ($\delta_x^{(i)}$ and $\delta_y^{(i)}$, respectively).

### D. Sampling rate degradation

The GazeBase dataset contains recordings of very high signal quality [29] recorded with an EyeLink 1000 eye tracker. Other eye trackers, such as Magic Leap One [30] or Vive Pro Eye [31], have lower signal quality (e.g., 60 Hz sampling rate for the Magic Leap One or 120 Hz for the Vive Pro Eye). It is expected that in the future, eye tracking would become ubiquitous in virtual- and augmented-reality head-mounted displays due to the many benefits it could bring, including foveated rendering, continuous authentication, and increased immersion in video games. But since eye tracking signal quality varies across devices, in this study, it was important to consider how signal quality (specifically, sampling rate) impacts authentication performance. To this end, we down-sampled the eye movement signals using SciPy's `decimate` function [32]. We targeted degraded sampling rates of 500, 250, 125, 50, and 31.25 Hz. We chose 31.25 Hz instead of 30 Hz to simplify the downsampling process.

### E. Network architecture

As input, we feed in a number of time steps equivalent to 1.024 s (rounded down, if not a whole number) and, if necessary, zero-pad the end of the sequence to length 1024. For each time step, we use the 4 channels defined in Equations 3–6. Our network performs a mapping $f : \mathbb{R}^{4\times1024} \rightarrow \mathbb{R}^{128}$.

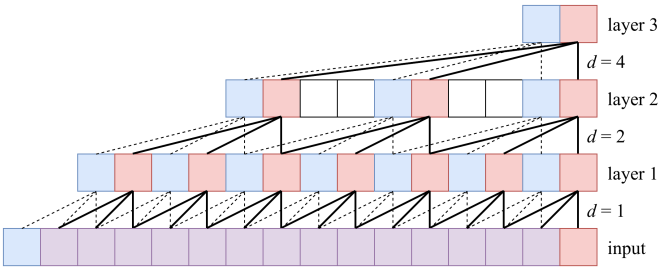



Fig. 1. Visualization of exponentially-dilated convolutions with kernel size 3, stride 1, and no padding. The convolutions in the $\ell$-th layer use a dilation of $d = 2^{\ell-1}$. With this configuration, if the input has length $2^q$ and there are $q - 1$ layers, then the final layer has a receptive field of $2^q - 1$ values from the input (shown in red and blue, with overlap in purple).

The network consists of a series of exponentially-dilated convolutional layers followed by a series of fully-connected layers. Exponentially-dilated convolutions were first proposed for semantic segmentation of images in [33] and have since seen success in other domains like audio synthesis [34] and time series classification [35], [36]. By exponentially increasing the dilation for subsequent convolutional layers, we achieve an exponential increase in the receptive field with only a linear increase in the number of learnable parameters. See Figure 1 for a visualization of exponentially-dilated convolutions and Figure 2 for a diagram of our network architecture.

### F. Multi-similarity loss

MS loss [13], like many other metric learning loss functions, is embedding-based (in contrast to classification-based losses like cross-entropy) [37] and pair-based. Minibatches are constructed from multiple samples each from a subset of subjects so that both inter- and intra-class variations can be observed. Pairs of samples are constructed within each minibatch. A pair is *positive* if samples in the pair are from the same class or *negative* if they are from different classes. The goal is to bring positive pairs closer together in the embedding space and to push negative pairs farther apart. In other words, we want to construct a well-clustered embedding space. One challenge with pair-based losses is selecting the most informative pairs to accelerate learning. Using pairs that are too easy does not help the model learn, and using pairs that are too hard may lead to instability during training [38].

MS loss takes into account three different types of similarities: self-similarity, positive relative similarity, and negative relative similarity. This is a more sophisticated approach than most other losses that focus on either self-similarity (e.g., contrastive loss [39]) or relative similarity (e.g., triplet loss [27]) but not both. The most informative pairs are selected with an online pair mining technique and assigned similarity-based weights that decay exponentially as the pairs become less informative. A larger weight is given to positive pairs with low similarity and to negative pairs with high similarity. Together, these aspects help MS loss form a well-clustered embedding space and overcome the challenge of selecting

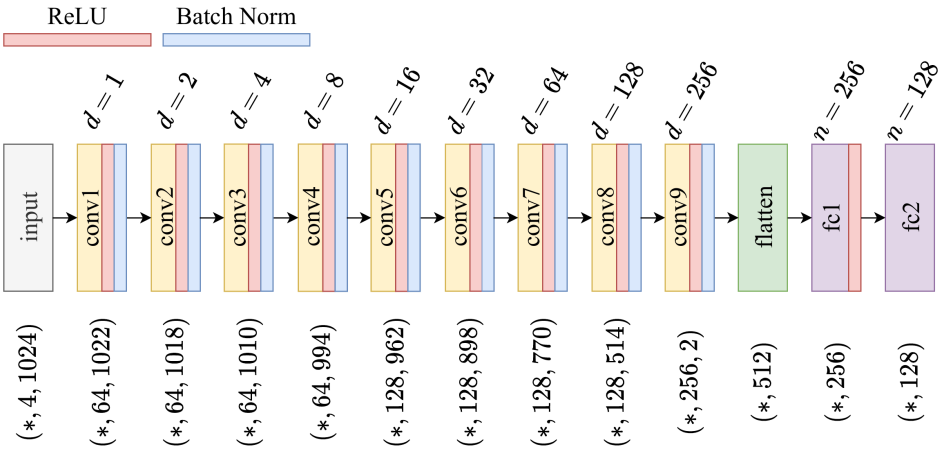



Fig. 2. Network architecture. Each convolution layer uses kernel size 3, stride 1, and no padding, and is followed by ReLU and batch normalization. The first fully-connected layer is followed by ReLU. The output of the final fully-connected layer acts as the embedding of the input. The numbers at the bottom reflect the shape of the data leaving each layer, ordered as minibatch size, channels, and time steps. The $d$ above each convolution block is the dilation used in that layer, and the $n$ above each fully-connected block is the number of output nodes in that layer. This network has a total of 475,264 learnable parameters.

informative pairs. MS loss is formulated as

$$L = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{\alpha}\log\left(1+\sum_{k\in P_i}\mathrm{e}^{-\alpha(S_{ik}-\lambda)}\right) + \frac{1}{\beta}\log\left(1+\sum_{k\in N_i}\mathrm{e}^{\beta(S_{ik}-\lambda)}\right)\right), \tag{7}$$

where $\alpha, \beta, \lambda$ are hyperparameters, $m$ is the size of the minibatch, $P_i$ and $N_i$ are the sets of indices of the mined positive and negative pairs for each anchor sample $\mathbf{x}_i$, and $S_{ik}$ is the cosine similarity between the pair of samples $\{\mathbf{x}_i, \mathbf{x}_k\}$. For more technical details and descriptive figures, please refer to the MS loss paper [13].

Although the latest metric learning loss functions (such as MS loss) may not improve upon earlier loss functions as much as the literature would suggest [37], many (including MS loss) do appear to lead to marginal improvements over cross-entropy or triplet loss after controlling for several factors including network architecture, batch size, and optimizer.

### G. Measuring similarity between two recordings

The similarity between two recordings is measured as the mean cosine similarity across the first $n$ temporally-aligned subsequence embeddings. That is, given two recordings $A, B$ where $\mathbf{a}^{(i)}$ is the embedding of the $i$-th non-overlapping subsequence from $A$ (and $\mathbf{b}^{(i)}$ from $B$), we compute the similarity between $A$ and $B$, denoted $S_{A,B}$, as

$$S_{A,B} = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbf{a}^{(i)}\cdot\mathbf{b}^{(i)}}{||\mathbf{a}^{(i)}||\,||\mathbf{b}^{(i)}||}. \tag{8}$$

### H. Training

For a given task and sampling rate, we trained 4 different models, each one using a different held-out fold as the validation set and the remaining 3 folds as the training set.

We used the AdamW [40] optimizer with learning rate and weight decay determined via hyperparameter search (see Section III-I). All other optimizer hyperparameters were left at their default values. We used MS loss [13] with an online miner as implemented in the PyTorch Metric Learning (PML) library [41]. The hyperparameters for MS loss were also determined via hyperparameter search.

When constructing a minibatch of size $m$, we observed that simply selecting $k$ samples each from $\frac{m}{k}$ classes/subjects tended to oversample from R1. To sample from all recording rounds equally and to guarantee that each minibatch contains longitudinal similarities, we pick one subject present in all of R1–5 (i.e., all rounds present in the training/validation sets), along with another random subject from each of R1–5. Then, for each of R1–5, we randomly sample $k$ subsequences each from the two subjects chosen for that round, with half of those samples taken from recording session 1 and the other half from recording session 2. Since we have 2 subjects each from 5 rounds, our minibatch size is $m = 10k$. We set $k = 8$ for a minibatch size of 80.

Rather than sampling from a fixed set of subsequences (e.g., constructed with a rolling window approach), the subsequences used during training were chosen from arbitrary positions in each recording. That is, when sampling a subsequence of length $w$ from a recording of length $T$, we start from a random time step $i \in [1, T - w + 1]$ and use the next $w$ contiguous time steps $\{i, i + 1, \ldots, i + w - 1\}$. In doing so, we force the network to learn similarities (and differences) between arbitrary subsequences of eye movements during the task and across recording sessions, hopefully improving its ability to generalize to new subjects. In our experiments, subsequence length $w$ was a number of time steps equivalent to 1.024 s (rounded down, if not a whole number).

After every 100 training iterations (i.e., 100 minibatches), the model's performance was evaluated in the following manner using data from the validation set. We first computed the embeddings of the first 10 non-overlapping sub-

sequences (10.240 s) of each recording in session 1 of R1. These embeddings acted as the enrollment set. We did the same for each recording in session 2 of R1 to construct the authentication set. We then computed $S_{A,B}$ for all $A$ in the enrollment set and all $B$ in the authentication set using Equation 8 with $n = 10$. A receiver operating characteristic (ROC) curve was built from these similarities and was used to find the equal error rate (EER)—the point where the false acceptance rate (FAR) was equal to the false rejection rate (FRR). We repeated this process with the same enrollment set but different authentication sets built from session 2 of R2–5, resulting in a total of 5 measures of EER. The mean of these 5 EERs was used as the combined measure of the model's performance.

The model trained for a maximum of 100,000 iterations, but we also employed early stopping to help reduce training time. Early stopping with a patience of 200 performance evaluations (20,000 iterations)—seeking to minimize the aforementioned performance measure—was used to determine if training should stop early. Whether training lasted the full 100,000 iterations or stopped early, we kept the version of the model with the best performance evaluation and discarded any other model checkpoints.

### I. Hyperparameter tuning

We had 6 hyperparameters to tune: 2 for AdamW (learning rate and weight decay), 3 for MS loss ($\alpha$, $\beta$, and $\lambda$), and 1 for the online mining used within MS loss ($\varepsilon$). Bayesian optimization [42] was employed to optimize these hyperparameters.

A total of 31 search iterations were performed. The first iteration used a fixed hyperparameter configuration that we empirically found to work well on the training and validation sets during preliminary experiments. The next 5 iterations (2–6) randomly explored the search space to help prevent the optimizer from getting stuck in local optima (a fixed random seed was used so that the same 5 random points were probed for all models). The final 25 iterations (7–31) intelligently balanced exploring and exploiting the search space using the upper confidence bound (UCB) utility/acquisition function.

At each search iteration, a total of 4 models were trained with the selected hyperparameters using the training procedure described in Section III-H. The final valuation for that point in the search space was mean(EER) $+ 1.96 \times$ SD(EER), using the combined measure of EER described in Section III-H and aggregated across the 4 models. Bayesian optimization sought to minimize this valuation. The set of hyperparameters with the lowest valuation after all 31 search iterations was used during our final analyses. Hyperparameters were tuned separately for each task and sampling rate. The hyperparameter search space and the hyperparameters used for each model are all provided in the supplementary material.

### J. Final evaluation on test set

There are two main scenarios for biometrics: authentication and identification. In the authentication scenario, a user attempts to access a system by claiming to be a specific enrolled user and presenting their biometric sample (e.g., a fingerprint), and a decision is made based on the sample's similarity to the biometric template of that specific enrolled user. In the identification scenario, a user attempts to access a system by claiming to be *any* enrolled user and presenting their biometric sample, which is then compared against the biometric template of *every* enrolled user to see if there is any match. As more users are enrolled, it becomes increasingly difficult to detect impostors in the identification scenario, as the impostor's presented biometric sample need only be similar to any one enrolled user's biometric template. This phenomenon has been formally studied using synthetic and real datasets [43]. Therefore, we consider only the authentication scenario, for which performance is expected to remain relatively consistent regardless of the number of enrolled users [43].

An important consideration for any biometric modality is how it compares to existing security methods. For example, the 4-digit pin is one of the most common security methods for smartphones and is often used as a backup authentication method if biometrics fail. If a biometric modality is less secure than a 4-digit pin, it would have little practical benefit on its own. There are $10^4$ possible ways to construct a 4-digit pin with the numbers 0–9, so assuming each pin is equally likely to be chosen, there is a $10^{-4}$ chance that an impostor would correctly guess a specific user's pin. Therefore, we provide measures of false rejection rate (FRR) when false acceptance rate (FAR) is fixed at $10^{-4}$, abbreviated FRR @ FAR $10^{-4}$. According to the FIDO Biometrics Requirements [44], a biometric system should achieve a FRR @ FAR $10^{-4}$ of no more than 5%.

We formed genuine and impostor pairs within the held-out test set for various test-retest intervals in a manner similar to the performance evaluation described in Section III-H. The enrollment set consists of the embeddings from the first $n$ non-overlapping subsequences of each recording in session 1 of R1. Nine separate authentication sets were constructed using the embeddings from the first $n$ non-overlapping subsequences of each recording in session 2 of R1–9. We then computed, for each authentication set separately, $S_{A,B}$ for all $A$ in the enrollment set and all $B$ in the authentication set using Equation 8.

Using the above method, we have a maximum of 59 genuine pairs and 3422 impostor pairs (e.g., when authenticating with R1) and a minimum of 14 genuine pairs and 812 impostor pairs (when authenticating with R9). This is fewer than the 10,000 impostor pairs needed to estimate FRR @ FAR $10^{-4}$.

To enable the estimation of FRR @ FAR $10^{-4}$, we perform the following resampling approach. We use the PearsonDS [45] R package to fit a Pearson family distribution to the empirical impostor similarity distribution for a given authentication set. We sample 20,000 new impostor similarities from this fitted distribution and discard the empirical impostor similarities. This fitted distribution closely matches the mean, variance, skewness, and kurtosis of our empirical distribution. We do the same for the genuine similarity scores to balance the classes. This provides us with enough data to be able to estimate FRR @ FAR $10^{-4}$. By resampling from a fitted Pearson family distribution instead of, say, bootstrapping, we are able to sample values that are close to—but not exactly the same as—our empirical distribution of similarity scores. See

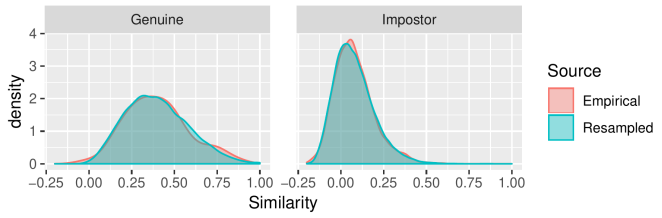



Fig. 3. A representative comparison between empirical and resampled similarity distributions. EERs based on the resampled distributions tend to be slightly pessimistic relative to EERs based on the empirical distributions.

Figure 3 for a representative comparison between empirical and resampled distributions.

We then construct a ROC curve and estimate EER for each authentication set separately, producing a total of 9 EER measures based on test-retest intervals as short as 20 minutes and as long as 37 months.

This entire process was repeated for each $n \in \{1, 5, 10\}$ to assess how authentication performance varies with the amount of data.

### K. Baseline models

*1) Statistical baseline (STAR):* One of the baselines we compare against is a statistical approach based on [16] that we will refer to as STAR. The same 4 folds and test set were used for STAR as were used for our metric learning approach. Briefly, each recording was classified into fixations, saccades, post-saccadic oscillations (PSOs), and noise using the modified Nyström and Holmqvist (MNH) classification algorithm [24]. A set of over 1,000 features [25] was extracted from the classified events. Feature distributions were transformed to like-normal using the Box-Cox transformation, and only sufficiently normal features were kept. Redundant features were removed next, either due to high correlations or due to multiple features measuring similar aspects (e.g., both mean and median of the same underlying feature). Lastly, principal component analysis (PCA) was performed, and the optimal set of features and principal components was determined through an iterative process. We note that STAR was designed to use the full duration of each recording, so results for STAR are based on the full recording duration.

*2) DeepEyedentificationLive baseline (DEL):* Our other baseline is the current state-of-the-art, DEL [4]. The code for DEL is publicly available[1], and we reimplemented the network in PyTorch to keep all models under a single framework. Excluding classification layers, our implementation of DEL has 209,085,024 learnable parameters (69,052,160 in the slow subnet, 139,933,408 in the fast subnet, and 99,456 in the final set of layers after concatenation).

To enable a more fair comparison against our own model, DEL underwent the same signal pre-processing, training, hyperparameter tuning, and test set evaluation as our own model, with only some minor differences. We used an input length of 1024 time steps and the AdamW optimizer to match our model, instead of the original implementation's use of 1000 time steps and the Adam optimizer. The input channels to the slow subnet were from Equations 3–4, and the input channels to the fast subnet were from Equations 5–6 (we did not include stimulus position as an input, and we only had monocular gaze signals). DEL was trained with PyTorch's `CrossEntropyLoss` instead of MS loss, and we trained the subnets individually before freezing their weights and training the final few fully-connected layers. After training, we removed the classification layer but still applied batch normalization and ReLU after the embedding layer, as is done in the public implementation. We did our best to match the weight initialization from Keras/Tensorflow (e.g., Keras uses a truncated normal distribution while PyTorch does not). Due to memory constraints, we used minibatches of 40 samples (with $k = 4$ instead of 8) constructed in the manner described in Section III-H. Only 2 hyperparameters needed tuned for DEL: learning rate and weight decay.

[1]https://osf.io/8es7z/

### L. Hardware & software

All models (except the STAR and DEL baselines) were trained inside Docker containers on two Lambda Labs workstations. One workstation was equipped with dual NVIDIA GeForce RTX 2080 Ti GPUs (11 GB VRAM), an Intel i9-10920X CPU @ 3.50 GHz (12 cores), and 128 GB RAM. The other workstation was equipped with dual NVIDIA GeForce RTX 3080 GPUs (10 GB VRAM), an Intel i9-10900X CPU @ 3.70 GHz (20 cores), and 64 GB RAM. The Docker containers ran Ubuntu 18.04 and were set up with Python 3.7.10, PyTorch 1.9.0, and PML [41] version 0.9.99.

Due to memory constraints, we needed to train the DEL baseline models on a different machine equipped with quad NVIDIA GeForce RTX A5000 GPUs (24 GB VRAM), an AMD Ryzen Threadripper PRO 3975WX CPU @ 3.5 GHz (32 cores), and 256 GB RAM. The slow and fast subnets were trained concurrently on separate GPUs to save time. DEL was trained inside a Docker container running Ubuntu 18.04 and set up with Python 3.7.11, PyTorch 1.10.0, and PML version 1.1.0.

Each of our models took up to 1 hour to train for each fold on the RTX 3080 (24.0 training iterations per second), and up to 2 hours on the RTX 2080 Ti (13.3 training iterations per second). The DEL baseline took an average of 2.8 hours to train for each fold on the RTX A5000, with the fast subnet often requiring more time to train than the slow subnet.

STAR was run on a Windows 10 computer, equipped with an Intel i7-6700K CPU @ 4.00 GHz (4 cores) and 16 GB RAM. The code was written in MATLAB 2020a and ran serially on the CPU.

Our full source code and trained models are available on the Texas State Digital Collections Repository at https://dataverse.tdl.org/dataverse/eky/.

## IV. Results & Discussion

Due to prevalent usage of reading data in the eye movement biometrics literature, we use TEX @ 1000 Hz as our representative dataset. The average performance measures of our approach on the held-out test set for TEX @ 1000 Hz are

given in Table II, using each $n \in \{1, 5, 10\}$ (for Equation 8) and each of R1–9 as the authentication set. In Table III, we present results for all tasks and sampling rates using $n = 10$ and using R1 or R6 as the authentication set. R1 has the shortest test-retest interval (approx. 20 minutes) and should, therefore, result in the best performance. R6 has a test-retest interval of approx. 1 year, is exclusive to the test set, and contains all 59 unique test-set subjects. Table IV shows results for the baseline models with TEX @ 1000 Hz, using R1 and R6 for authentication. We note that the full duration of each recording was used for STAR, while only the first 10.240 s were used for DEL and our approach.

For a more comprehensive view of each model's performance, refer to the tables in the supplementary materials.

In addition to the quantitative results in the aforementioned tables, we also present some qualitative results. For these figures, each model uses $n = 10$ (except STAR, which uses the full recording duration) and R1 for authentication. Figure 4 shows a comparison between the genuine and impostor similarity score distributions. An ROC curve for each model is presented in Figure 5.

### A. Comparison to baselines

*1) STAR:* Looking at Table IV, compared to STAR, our approach results in lower EERs and is more stable across folds.

The genuine vs impostor distributions (Figure 4B) and the ROC curve (Figure 5C) for STAR deserve additional discussion. The genuine and impostor distributions are not unimodal in the presented figure, but this is because the figure includes 4 separate genuine/impostor distributions (one per model). The mean ROC curve shows a sharp increase in FRR when FAR is approximately 0.08, and another around 0.005 FAR. We believe these abnormalities are due to different features and principal components being included for each fold, resulting in vastly different performance between models. Perhaps it would have been better to use a consistent set of features and principal components across folds.

STAR requires event classification and manual feature extraction, making it much harder to employ in different datasets and for different tasks than our end-to-end approach. It was also designed to use the full duration of each recording, limiting its use in practice.

Regarding the discrepancy between our results with STAR and the original results from [16] (where an EER of 10% was achieved on data separated by approximately 1 year), there are many contributing factors.

First, our EER estimates are based on different data than the original study. We evaluated with as many as 59 subjects, did not remove signal after the end of reading, and did not exclude any subjects from the analysis regardless of data quality. In contrast, the original study used either 149 subjects (SBA-ST) or 34 subjects (SBA-LT), removed signal after the end of reading, and screened subjects with "low recording quality" or who had "excessively noisy recordings."

Second, we made several changes to the original approach: we simplified the normality transformations by always using Box-Cox instead of trying several different standard transformation functions; we did not winsorize the distributions to

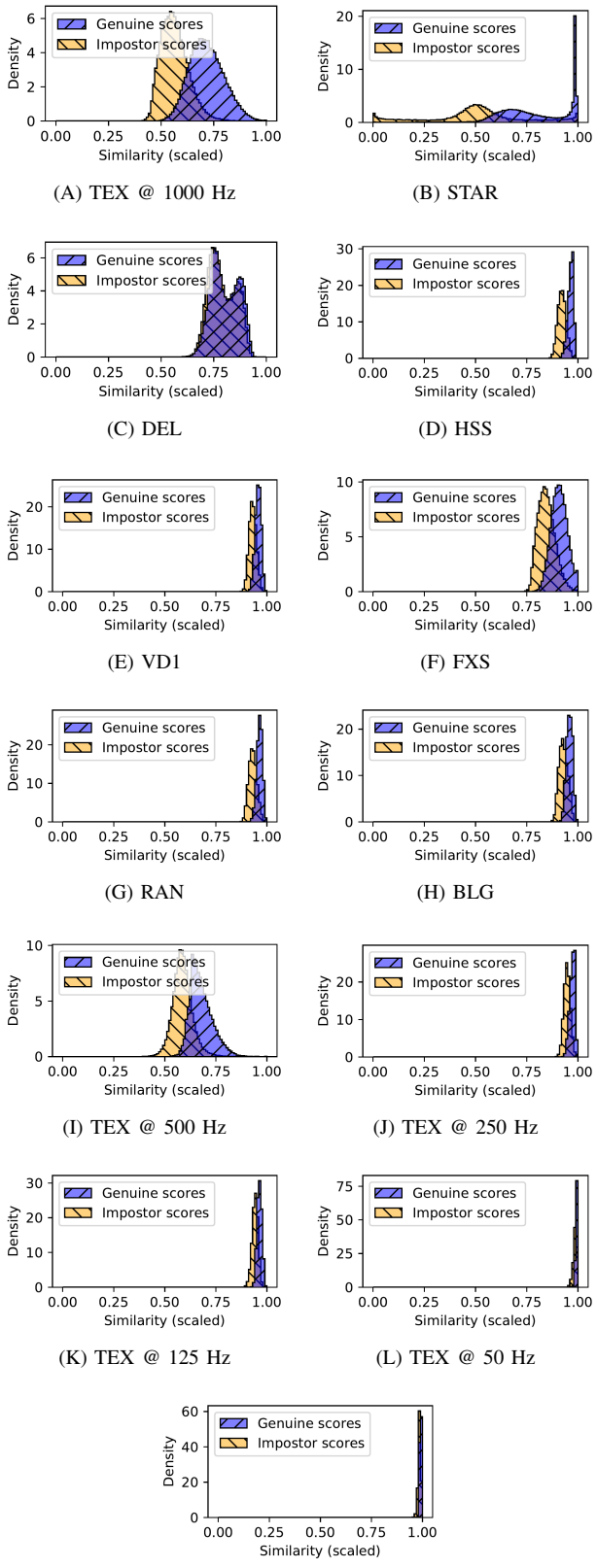



Fig. 4. Plots of the similarity distributions for genuine and impostor pairs. Each plot contains the similarities on the held-out test set computed separately for each of the 4 models trained with 4-fold cross-validation, using $n = 10$ for Equation 8 and R1 for authentication. Since cosine similarity is bounded from -1 to 1, we scaled the similarities to lie between 0 and 1 before plotting. A bin width of 0.01 was used, and the area under each curve sums to 1.

TABLE II
RESULTS ON THE HELD-OUT TEST SET FOR TEX @ 1000 HZ, VARYING THE $n$ USED FOR EQUATION 8 AND THE ROUND USED FOR THE AUTHENTICATION SET. VALUES ARE PRESENTED AS MEAN (STANDARD DEVIATION) ACROSS THE 4 MODELS TRAINED WITH 4-FOLD CROSS-VALIDATION.

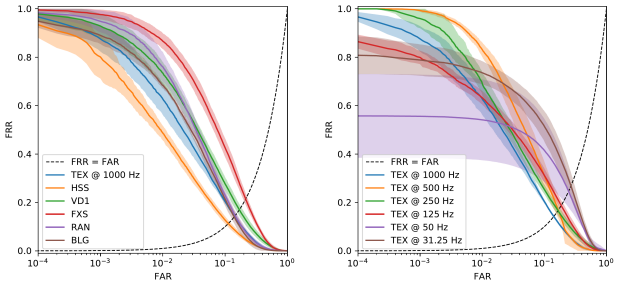
| $n$ | Round | EER | FRR @ FAR | | | |
|---|---|---|---|---|---|---|
| | | | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ |
| 1 | 1 | 0.2237 (0.0115) | 0.3866 (0.0256) | 0.7483 (0.0377) | 0.9370 (0.0378) | 0.9870 (0.0168) |
| | 2 | 0.2903 (0.0214) | 0.5415 (0.0402) | 0.8584 (0.0295) | 0.9673 (0.0183) | 0.9947 (0.0044) |
| | 3 | 0.3045 (0.0214) | 0.5582 (0.0339) | 0.8684 (0.0121) | 0.9775 (0.0082) | 0.9979 (0.0032) |
| | 4 | 0.3103 (0.0170) | 0.5754 (0.0334) | 0.8697 (0.0241) | 0.9714 (0.0128) | 0.9959 (0.0028) |
| | 5 | 0.2760 (0.0143) | 0.4741 (0.0120) | 0.7965 (0.0224) | 0.9738 (0.0159) | 0.9977 (0.0040) |
| | 6 | 0.2783 (0.0249) | 0.5134 (0.0288) | 0.8434 (0.0303) | 0.9640 (0.0204) | 0.9939 (0.0078) |
| | 7 | 0.3051 (0.0260) | 0.5349 (0.0521) | 0.8272 (0.0382) | 0.9549 (0.0218) | 0.9904 (0.0068) |
| | 8 | 0.3036 (0.0253) | 0.5502 (0.0529) | 0.8765 (0.0317) | 0.9942 (0.0101) | 0.9995 (0.0009) |
| | 9 | 0.2841 (0.0361) | 0.5623 (0.0427) | 0.9027 (0.0617) | 0.9787 (0.0191) | 0.9962 (0.0042) |
| 5 | 1 | 0.1488 (0.0075) | 0.2206 (0.0193) | 0.6284 (0.0579) | 0.8790 (0.0543) | 0.9619 (0.0240) |
| | 2 | 0.1952 (0.0100) | 0.3449 (0.0188) | 0.7801 (0.0429) | 0.9577 (0.0237) | 0.9955 (0.0039) |
| | 3 | 0.2260 (0.0154) | 0.3981 (0.0290) | 0.7709 (0.0284) | 0.9318 (0.0292) | 0.9827 (0.0111) |
| | 4 | 0.2008 (0.0144) | 0.3694 (0.0266) | 0.7831 (0.0211) | 0.9492 (0.0191) | 0.9920 (0.0042) |
| | 5 | 0.2037 (0.0144) | 0.3638 (0.0335) | 0.7451 (0.0089) | 0.9162 (0.0238) | 0.9679 (0.0213) |
| | 6 | 0.2178 (0.0105) | 0.4058 (0.0259) | 0.8030 (0.0389) | 0.9608 (0.0233) | 0.9895 (0.0060) |
| | 7 | 0.2367 (0.0166) | 0.4440 (0.0276) | 0.8070 (0.0251) | 0.9455 (0.0286) | 0.9792 (0.0186) |
| | 8 | 0.2432 (0.0285) | 0.4433 (0.0446) | 0.8192 (0.0434) | 0.9679 (0.0179) | 0.9988 (0.0018) |
| | 9 | 0.1815 (0.0125) | 0.3537 (0.0732) | 0.8955 (0.0762) | 0.9793 (0.0215) | 0.9939 (0.0070) |
| 10 | 1 | 0.1420 (0.0032) | 0.2038 (0.0084) | 0.6272 (0.0646) | 0.8758 (0.0524) | 0.9665 (0.0200) |
| | 2 | 0.1924 (0.0103) | 0.3426 (0.0184) | 0.7782 (0.0437) | 0.9464 (0.0263) | 0.9865 (0.0080) |
| | 3 | 0.2110 (0.0147) | 0.3847 (0.0392) | 0.7735 (0.0425) | 0.9371 (0.0295) | 0.9829 (0.0158) |
| | 4 | 0.1952 (0.0133) | 0.3397 (0.0342) | 0.7530 (0.0142) | 0.9254 (0.0213) | 0.9802 (0.0167) |
| | 5 | 0.1889 (0.0112) | 0.3354 (0.0368) | 0.7073 (0.0205) | 0.8899 (0.0301) | 0.9490 (0.0336) |
| | 6 | 0.2110 (0.0148) | 0.3915 (0.0364) | 0.7900 (0.0438) | 0.9451 (0.0294) | 0.9845 (0.0111) |
| | 7 | 0.2277 (0.0140) | 0.4263 (0.0286) | 0.7834 (0.0335) | 0.9315 (0.0271) | 0.9829 (0.0133) |
| | 8 | 0.2156 (0.0169) | 0.4217 (0.0340) | 0.8586 (0.0399) | 0.9803 (0.0102) | 0.9983 (0.0019) |
| | 9 | 0.2017 (0.0035) | 0.3798 (0.0391) | 0.8766 (0.0866) | 0.9716 (0.0280) | 0.9922 (0.0077) |

TABLE III
RESULTS ON THE HELD-OUT TEST SET FOR EVERY TASK AND SAMPLING RATE. FOR BREVITY, RESULTS ARE ONLY SHOWN WHEN USING R1 AND R6 FOR THE AUTHENTICATION SET, AND ONLY WHEN USING $n = 10$ IN EQUATION 8. VALUES ARE PRESENTED AS MEAN (STANDARD DEVIATION) ACROSS THE 4 MODELS TRAINED WITH 4-FOLD CROSS-VALIDATION. THE BEST RESULT FOR EACH ROUND IS BOLDED.
*: FOR BLG, 3 SUBJECTS (1, 120, AND 180) WERE EXCLUDED AT TEST TIME FOR HAVING A RECORDING WITH A DURATION LESS THAN 10.240 S.

| Task @ Rate | Round | EER | FRR @ FAR | | | |
|---|---|---|---|---|---|---|
| | | | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ |
| HSS @ 1000 Hz | 1 | **0.1125 (0.0071)** | **0.1273 (0.0154)** | **0.4867 (0.0299)** | 0.8010 (0.0891) | 0.9343 (0.0542) |
| | 6 | 0.2310 (0.0082) | 0.4630 (0.0174) | 0.8377 (0.0302) | 0.9495 (0.0210) | 0.9808 (0.0096) |
| VD1 @ 1000 Hz | 1 | 0.1694 (0.0112) | 0.2770 (0.0276) | 0.7343 (0.0460) | 0.9288 (0.0498) | 0.9789 (0.0298) |
| | 6 | 0.3151 (0.0179) | 0.6099 (0.0297) | 0.9125 (0.0016) | 0.9901 (0.0080) | 0.9979 (0.0030) |
| FXS @ 1000 Hz | 1 | 0.2136 (0.0102) | 0.4260 (0.0389) | 0.8664 (0.0329) | 0.9763 (0.0205) | 0.9953 (0.0064) |
| | 6 | 0.3654 (0.0107) | 0.7113 (0.0218) | 0.9518 (0.0040) | 0.9949 (0.0021) | 0.9994 (0.0005) |
| RAN @ 1000 Hz | 1 | 0.1459 (0.0079) | 0.2293 (0.0210) | 0.7602 (0.0619) | 0.9481 (0.0445) | 0.9836 (0.0175) |
| | 6 | 0.2626 (0.0194) | 0.5294 (0.0385) | 0.9188 (0.0160) | 0.9927 (0.0081) | 0.9987 (0.0017) |
| *BLG @ 1000 Hz | 1 | 0.1404 (0.0112) | 0.2140 (0.0273) | 0.6835 (0.0125) | 0.8902 (0.0355) | 0.9488 (0.0272) |
| | 6 | 0.2833 (0.0132) | 0.5578 (0.0313) | 0.8490 (0.0181) | 0.9315 (0.0282) | 0.9528 (0.0330) |
| TEX @ 1000 Hz | 1 | 0.1420 (0.0032) | 0.2038 (0.0084) | 0.6272 (0.0646) | 0.8758 (0.0524) | 0.9665 (0.0200) |
| | 6 | **0.2110 (0.0148)** | **0.3915 (0.0364)** | 0.7900 (0.0438) | 0.9451 (0.0294) | 0.9845 (0.0111) |
| TEX @ 500 Hz | 1 | 0.1667 (0.0155) | 0.3110 (0.0577) | 0.8315 (0.0293) | 0.9928 (0.0051) | 1.0000 (0.0000) |
| | 6 | 0.2764 (0.0158) | 0.5618 (0.0317) | 0.9280 (0.0179) | 0.9975 (0.0020) | 0.9999 (0.0001) |
| TEX @ 250 Hz | 1 | 0.1661 (0.0091) | 0.2540 (0.0201) | 0.6916 (0.0439) | 0.9597 (0.0399) | 0.9999 (0.0001) |
| | 6 | 0.2660 (0.0157) | 0.4845 (0.0280) | 0.8560 (0.0304) | 0.9790 (0.0196) | 0.9959 (0.0067) |
| TEX @ 125 Hz | 1 | 0.1875 (0.0106) | 0.3123 (0.0281) | 0.6327 (0.0277) | 0.7990 (0.0276) | 0.8642 (0.0289) |
| | 6 | 0.2499 (0.0165) | 0.4780 (0.0359) | 0.7994 (0.0335) | 0.9159 (0.0219) | 0.9504 (0.0201) |
| TEX @ 50 Hz | 1 | 0.2371 (0.0436) | 0.4152 (0.1301) | 0.5411 (0.1669) | **0.5558 (0.1720)** | **0.5577 (0.1730)** |
| | 6 | 0.2466 (0.0394) | 0.4742 (0.1329) | 0.7981 (0.1169) | **0.8150 (0.1075)** | **0.8173 (0.1064)** |
| TEX @ 31.25 Hz | 1 | 0.2666 (0.0246) | 0.5147 (0.0628) | 0.7356 (0.0476) | 0.7881 (0.0645) | 0.8079 (0.0779) |
| | 6 | 0.3047 (0.0281) | 0.5902 (0.0591) | **0.7796 (0.0368)** | 0.8282 (0.0537) | 0.8404 (0.0600) |

TABLE IV
BASELINE PERFORMANCE MEASURES COMPUTED ON THE HELD-OUT TEST SET USING TEX @ 1000 HZ. FOR BREVITY, RESULTS ARE ONLY SHOWN WHEN USING R1 AND R6 FOR THE AUTHENTICATION SET, AND ONLY WHEN USING $n = 10$ IN EQUATION 8 (FULL RECORDING DURATION USED FOR STAR). VALUES ARE GIVEN AS MEAN (STANDARD DEVIATION), AGGREGATED ACROSS 4 MODELS TRAINED VIA 4-FOLD CROSS-VALIDATION. THE BEST RESULT FOR EACH ROUND IS BOLDED.

| Model | Round | EER | FRR @ FAR | | | |
|---|---|---|---|---|---|---|
| | | | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ |
| STAR | 1 | 0.1563 (0.0487) | 0.2249 (0.1101) | 0.6240 (0.2479) | **0.8040 (0.1962)** | 0.9107 (0.0917) |
| | 6 | 0.2461 (0.0620) | 0.4161 (0.1188) | 0.8284 (0.1727) | 0.9237 (0.0765) | 0.9727 (0.0281) |
| DEL | 1 | 0.4295 (0.0213) | 0.8001 (0.0240) | 0.9706 (0.0165) | 0.9955 (0.0033) | 0.9992 (0.0005) |
| | 6 | 0.4748 (0.0197) | 0.8465 (0.0301) | 0.9867 (0.0172) | 0.9973 (0.0034) | 0.9991 (0.0010) |
| DEL (slow) | 1 | 0.1559 (0.0394) | 0.2226 (0.0854) | **0.5822 (0.1083)** | 0.8558 (0.1072) | 0.9550 (0.0483) |
| | 6 | 0.2314 (0.0336) | 0.4053 (0.0807) | **0.7619 (0.0662)** | 0.9241 (0.0438) | 0.9687 (0.0230) |
| DEL (fast) | 1 | 0.2111 (0.0377) | 0.3631 (0.0737) | 0.6995 (0.0371) | 0.8208 (0.0766) | **0.8691 (0.0973)** |
| | 6 | 0.2673 (0.0326) | 0.5296 (0.0612) | 0.8208 (0.0189) | **0.8967 (0.0317)** | **0.9146 (0.0403)** |
| Ours | 1 | **0.1420 (0.0032)** | **0.2038 (0.0084)** | 0.6272 (0.0646) | 0.8758 (0.0524) | 0.9665 (0.0200) |
| | 6 | **0.2110 (0.0148)** | **0.3915 (0.0364)** | 0.7900 (0.0438) | 0.9451 (0.0294) | 0.9845 (0.0111) |




(A) Varying tasks (B) Varying sampling rate

(C) Baselines

Fig. 5. ROC curves to provide a qualitative assessment of model performance using $n = 10$ and R1 for authentication. The horizontal axis is log-scaled false acceptance rate (FAR). The vertical axis is false rejection rate (FRR). Each ROC curve represents the mean performance across 4 models trained with 4-fold cross-validation, and each shaded region indicates $\pm 1$ SD about the mean. The point where the dashed line intersects each ROC curve indicates the EER for that curve.

try to improve normality; we tested normality by checking skewness and kurtosis instead of using the chi-square test; we measured reliability on a set of data disjoint from our test set; and we determined the best set of features and number of principal components using mean EER across rounds rather than rank-1 identification rate.

Third, we did not have access to several oculomotor plant characteristic (OPC) features that were present in the original study, some of which were found to be highly reliable in the original study.

*2) DEL:* We were unable to get reasonable performance with the full DEL model. Since the fast and slow subnets both performed well on their own, we included their performance in our results. Looking at Table IV, we see that the full DEL model had near-random performance with 42.95% R1 EER. The slow and fast subnets individually performed better than the full DEL model, achieving R1 EERs of 15.59% and 21.11%, respectively. Our proposed model outperforms the DEL baseline, though the difference between the slow subnet and our model may not be statistically significant.

As with STAR, the genuine and impostor distributions (Fig. 4C) are not unimodal, but this is because the figure includes 4 separate genuine/impostor distributions (one per model).

We note that our approach outperforms DEL, despite ours having 440x fewer learnable parameters (145x fewer than the slow subnet). This significant reduction in model complexity may enable more power-efficient implementations in certain target settings, such as embedded environments.

Regarding the discrepancy between our results with DEL and the original results from [4] (where an EER @ 10 s of 5% was achieved on data separated by $\geq$1–4 weeks), we note that we used a different data set and evaluated the model on a held-out test set of as many as 59 subjects (compared to 25 subjects used in the original study). Additionally, we had only 1 enrolled recording per subject in the test set, whereas the original study had multiple. We also determined similarity differently. The original DEL study [4] checks if *any* window from *any* enrolled sequence is sufficiently similar to *any* window from the presented sequence during authentication. In contrast, we measured the mean similarity across each pair of temporally-aligned windows from the enrolled sequence and the presented sequence during authentication.

### B. Authentication accuracy vs test-retest interval

Looking at Table II, we see that performance when authenticating on R1 (test-retest interval of approx. 20 minutes) is significantly better than later rounds. This matches our expectation.

We note that R1–6 all have 59 subjects, R7 has 35, R8 has 31, and R9 has 14. The reduction in subject count for R7–9 may partially explain the reduction in EER despite the increase in test-retest interval. We also note that data for other subjects from R1–5 were present in the training and validation set, while R6–9 were exclusive to the test set.

### C. Authentication accuracy vs recording duration

Looking at Table II, we see that our model still performs better than chance (50% EER) using just the first 1.024 s ($n = 1$) from each recording for both enrollment and authentication. Performance drastically improves when the first 5.120 s ($n = 5$) are used but does not improve much further when using the first 10.240 s ($n = 10$).

Although we focused on low data requirements, it may be worth mentioning that in an additional analysis, we evaluated the TEX @ 1000 Hz model using (roughly) the full recording duration ($n = 58$, or 59.392 s) for both enrollment and authentication. In this higher data setting, our model achieved a mean R1 EER of 10.52%—an improvement of nearly 4 percentage points compared to using $n = 10$.

### D. Authentication accuracy vs sampling rate

Looking at the bottom half of Table III, we see that R1 EER worsened as sampling rate was reduced, which aligns with our expectations. However, R6 EER monotonically improved (slightly) as sampling rate was degraded, starting from 500 Hz down to as low as 50 Hz. Figure 4 shows that as sampling rate is degraded and fewer time steps are present in the input, the models gradually become less capable of producing embeddings that are highly dissimilar.

Interestingly, R1 FRR @ FAR $10^{-4}$ was its lowest at 50 Hz (see Table III). Of course, a FRR of 56% is still unusable in practice, and the variance across folds was largest for 50 Hz; but this was an interesting result nonetheless.

### E. Authentication accuracy vs task

Looking at the top half of Table III, we find that HSS, a low-cognitive-load task, resulted in the best R1 performance across tasks. TEX resulted in the best R6 performance across tasks, but HSS was still competitive despite requiring significantly less mental effort from the participants. Unsurprisingly, FXS resulted in the worst performance of all the tasks, but still did better than chance (50% EER) even on R6.

### F. Explanation of high error rates

The results presented herein are nominally worse than those of many prior works in the literature. We highlight, for the following reasons, that this relatively poor performance is due to our work attempting to solve a harder problem that is more practically relevant and thus would be more indicative of real-world performance with the selected architecture. Within this more realistic scenario, our proposed architecture outperformed all state-of-the-art models.

**Low data setting.** Our approach for authentication requires very little data (up to 10.240 s collected during one sitting), during both enrollment and verification. The majority of prior works require significantly more data. For example, DEL [4] uses 9 separate trials (totaling 26.250 s) collected over a period of at least 3 weeks for enrollment. The statistical method by Friedman et al. [16] uses 60 s of data during both enrollment and verification. We believe that it is important for future studies to focus on requiring less data to improve the practical utility of eye movements as a biometric.

**No data screening.** We did not clean the data set at all (beyond any prior screening employed for the GazeBase data set itself). As a result, the data we used included noisy signals littered with NaN values, and some windows of data used during training and evaluation had a significant amount of missing data.

**Held-out test set.** We used a separate held-out set of data for the final evaluation of our model. This held-out set contains nearly 50% of all recordings in GazeBase, resulting in much less data available for training. It also contains data from 59 subjects which is a larger population than many prior studies consider. Most prior studies do not use a separate held-out set of data, so their estimates of model performance may be more biased (in their favor).

**Pessimistic resampling.** Our approach of resampling the similarity scores with a Pearson family distribution tended to produce pessimistic estimates of model performance that are likely more indicative of real-world performance. While our measures of FRR @ FAR $10^{-4}$ are very high to the point of limited practical use, we note that our study is the first in the field to report measures of FRR @ FAR $10^{-4}$.

### G. Limitations

There is an implicit assumption that the embeddings of each subsequence within a recording come from the same distribution. This assumption is necessary for the metric learning model to learn a well-clustered embedding space, as the subsequence embeddings for a given subject should follow some central tendency. However, during TEX, for instance, this assumption is almost certainly violated for whichever subsequence inevitably contains the large return saccade that occurs when a participant finishes reading the text and starts re-reading it. It may be advantageous to exclude such anomalous subsequences during training.

We note that degrading sampling rate alone is not sufficient to emulate other eye trackers with worse signal quality. There would also be differences in other eye tracking signal quality metrics including spatial accuracy, spatial precision, temporal precision, linearity, and crosstalk.

The EERs presented in the present study (and virtually every other study in the field) are based on a threshold determined on the test set data. We note that doing so essentially leaks test set data into the decision process. Of course, in a real-world scenario, it would be necessary to determine the threshold using the training and validation data and then later apply that threshold to the similarity scores produced on unseen (test) data.

Our method of scoring the "goodness" of a model using $\text{mean(EER)} + 1.96 \times \text{SD(EER)}$ may have erroneously favored

worse-performing models. It may be suggested for future works to exclude the SD term.

## V. Conclusion

We presented a metric learning approach for end-to-end biometric authentication via eye movements. Our proposed model employed exponentially-dilated convolutions to exponentially increase the receptive field of subsequent layers while only linearly increasing the number of parameters. Our approach was validated on the publicly available GazeBase dataset [14] using recordings collected as much as 37 months apart, and we compared our approach against a statistical baseline and the current state-of-the-art, DEL.

When authenticating on R1 with a test-retest interval of approx. 20 minutes and using the first 10.240 s of each recording, we achieved an EER as low as 11.25% on HSS @ 1000 Hz and a FRR @ FAR $10^{-4}$ as low as 55.77% on TEX @ 50 Hz. Even on R9 with a 37-month test-retest interval, we were able to achieve an EER as low as 18.15% using the first 5.120 s of each recording (on a reduced pool of 14 subjects), but FRR @ FAR $10^{-4}$ was consistently above 99%.

We have defined a testing scenario that is more realistic than most prior works, and we outperform the current state-of-the-art, DEL, under that testing scenario despite having 440x fewer learnable parameters and requiring a fraction of the time to train. Our work rehighlights the applicability of CNN architectures for eye movement biometrics and shows that even efficient architectures can achieve good performance. We believe that the scope and diversity (not only in terms of tasks, but also participant characteristics) of the GazeBase dataset gives it potential to serve as a unifying dataset for future biometrics research. We also encourage future eye movement biometrics studies to report FRR @ FAR $10^{-4}$ in a combined effort to eventually achieve the FIDO Alliance's recommendation of 5% FRR @ FAR $10^{-4}$.

## Acknowledgment

The authors would like to thank Dr. Lee Friedman for his suggestion of using the Pearson family of distributions for resampling similarities. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1144466. The study was also funded by 3 grants to Dr. Komogortsev: (1) National Science Foundation, CNS-1250718 and CNS-1714623, www.NSF.gov; (2) National Institute of Standards and Technology, 60NANB15D325, www.NIST.gov; (3) National Institute of Standards and Technology, 60NANB16D293. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the National Institute of Standards and Technology.

## References

[1] C. Katsini, Y. Abdrabou, G. E. Raptis, M. Khamis, and F. Alt, "The Role of Eye Gaze in Security and Privacy Applications: Survey and Future HCI Research Directions," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, Apr. 2020, pp. 1–21. [Online]. Available: https://doi.org/10.1145/3313831.3376840

[2] G. Bargary, J. M. Bosten, P. T. Goodbourn, A. J. Lawrance-Owen, R. E. Hogg, and J. D. Mollon, "Individual differences in human eye movements: An oculomotor signature?" *Vision Research*, vol. 141, pp. 157–169, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698917300391

[3] O. V. Komogortsev, A. Karpov, and C. D. Holland, "Attack of mechanical replicas: Liveness detection with eye movements," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 716–725, 2015.

[4] S. Makowski, L. A. Jäger, P. Prasse, and T. Scheffer, "Biometric identification and presentation-attack detection using micro- and macro-movements of the eyes," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, 2020, pp. 1–10.

[5] S. Eberz, K. Rasmussen, V. Lenders, and I. Martinovic, "Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics," 2015.

[6] P. Kasprowski and K. Harezlak, "Fusion of eye movement and mouse dynamics for reliable behavioral biometrics," *Pattern Analysis and Applications*, vol. 21, no. 1, pp. 91–103, Feb. 2018. [Online]. Available: https://doi.org/10.1007/s10044-016-0568-5

[7] A. George and A. Routray, "A score level fusion method for eye movement biometrics," *Pattern Recognition Letters*, vol. 82, pp. 207–215, oct 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865515004067

[8] P. Kasprowski and J. Ober, "Eye movements in biometrics," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3087, pp. 248–258, 2004.

[9] C. Li, J. Xue, C. Quan, J. Yue, and C. Zhang, "Biometric recognition via texture features of eye movement trajectories in a visual searching task," *PLoS ONE*, vol. 13, no. 4, p. e0194475, apr 2018. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0194475

[10] S. Jia, D. H. Koh, A. Seccia, P. Antonenko, R. Lamb, A. Keil, M. Schneps, and M. Pomplun, "Biometric recognition through eye movements using a recurrent neural network," in *Proceedings - 9th IEEE International Conference on Big Knowledge, ICBK 2018*. Institute of Electrical and Electronics Engineers Inc., dec 2018, pp. 57–64.

[11] A. Abdelwahab and N. Landwehr, "Deep Distributional Sequence Embeddings Based on a Wasserstein Loss," *arXiv:1912.01933 [cs, stat]*, Dec. 2019, arXiv: 1912.01933. [Online]. Available: http://arxiv.org/abs/1912.01933

[12] D. J. Lohr, S. Aziz, and O. Komogortsev, "Eye movement biometrics using a new dataset collected in virtual reality," in *ACM Symposium on Eye Tracking Research and Applications*, ser. ETRA '20 Adjunct. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3379157.3391420

[13] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5017–5025.

[14] H. Griffith, D. Lohr, E. Abdulin, and O. Komogortsev, "Gazebase: A large-scale, multi-stimulus, longitudinal eye movement dataset," 2020.

[15] C. Holland and O. V. Komogortsev, "Biometric identification via eye movement scanpaths in reading," in *2011 International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1–8.

[16] L. Friedman, M. S. Nixon, and O. V. Komogortsev, "Method to assess the temporal persistence of potential biometric features: Application to oculomotor, gait, face and brain structure databases," *PLOS ONE*, vol. 12, no. 6, pp. 1–42, 06 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0178501

[17] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," vol. 3, no. 1, pp. 5–24.

[18] L. A. Jäger, S. Makowski, P. Prasse, S. Liehr, M. Seidler, and T. Scheffer, "Deep eyedentification: Biometric identification using micro-movements of the eye," in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds. Cham: Springer International Publishing, 2020, pp. 299–314.

[19] P. Prasse, L. A. Jäger, S. Makowski, M. Feuerpfeil, and T. Scheffer, "On the relationship between eye tracking resolution and performance of oculomotoric biometric identification," *Procedia Computer Science*, vol. 176, pp. 2088–2097, 2020, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920321487

[20] C. Galdi, M. Nappi, D. Riccio, and H. Wechsler, "Eye movement analysis for human authentication: a critical survey," vol. 84, pp. 272–283. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865516303154
[21] I. Rigas and O. V. Komogortsev, "Current research in eye movement biometrics: An analysis based on BioEye 2015 competition," vol. 58, pp. 129–141.
[22] R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nyström, "One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms," *Behavior Research Methods*, vol. 49, no. 2, pp. 616–637, 2017. [Online]. Available: https://doi.org/10.3758/s13428-016-0738-9
[23] H. K. Griffith and O. V. Komogortsev, "Texture feature extraction from free-viewing scan paths using gabor filters with downsampling," in *ACM Symposium on Eye Tracking Research and Applications*, ser. ETRA '20 Adjunct. Association for Computing Machinery, pp. 1–3. [Online]. Available: https://doi.org/10.1145/3379157.3391423
[24] L. Friedman, I. Rigas, E. Abdulin, and O. V. Komogortsev, "A novel evaluation of two related and two independent algorithms for eye movement classification during reading," *Behavior Research Methods*, vol. 50, no. 4, pp. 1374–1397, 08 2018. [Online]. Available: https://doi.org/10.3758/s13428-018-1050-7
[25] I. Rigas, L. Friedman, and O. Komogortsev, "Study of an extensive set of eye movement features: Extraction methods and statistical analysis," *Journal of Eye Movement Research*, vol. 11, no. 1, 2018.
[26] S. Makowski, L. A. Jäger, A. Abdelwahab, N. Landwehr, and T. Scheffer, "A discriminative model for identifying readers and assessing text comprehension from eye movements," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 209–225.
[27] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015. IEEE Computer Society, oct 2015, pp. 815–823.
[28] H. Griffith, D. Lohr, and O. V. Komogortsev, "Gazebase data repository," 9 2020. [Online]. Available: https://figshare.com/articles/dataset/GazeBase_Data_Repository/12912257
[29] D. J. Lohr, L. Friedman, and O. V. Komogortsev, "Evaluating the data quality of eye tracking signals from a virtual reality system: Case study using smi's eye-tracking htc vive," 2019.
[30] "Magic Leap 1," https://www.magicleap.com/en-us/magic-leap-1, accessed: 2021-04-07.
[31] "Vive Pro Eye," https://www.vive.com/us/product/vive-pro-eye/overview/, accessed: 2021-04-07.
[32] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
[33] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2016.
[34] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.
[35] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018.
[36] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," 2020.
[37] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 681–699.
[38] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," mar 2017. [Online]. Available: http://arxiv.org/abs/1703.07737
[39] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1735–1742.
[40] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019*. International Conference on Learning Representations, ICLR, nov 2019.
[41] K. Musgrave, S. Belongie, and S.-N. Lim, "Pytorch metric learning," 2020.
[42] J. Mockus, V. Tiesis, and A. Zilinskas, *The application of Bayesian methods for seeking the extremum*, 1978, vol. 2, pp. 117–129.
[43] L. Friedman, H. S. Stern, V. Prokopenko, S. Djanian, H. K. Griffith, and O. V. Komogortsev, "Biometric performance as a function of gallery size," 2020.
[44] S. Schuckers, G. Cannon, and N. Tekampe, "FIDO biometrics requirements," https://fidoalliance.org/specs/biometric/requirements/, accessed: 2021-04-04.
[45] M. Becker and S. Klößner, *PearsonDS: Pearson Distribution System*, 2017, r package version 1.1. [Online]. Available: https://CRAN.R-project.org/package=PearsonDS

**Dillon Lohr** was born in Phoenix, AZ, USA. He received the B.S. degree in computer science from Texas State University, San Marcos, TX, USA, in 2018. He is currently a Ph.D. student in the Department of Computer Science at Texas State University with an anticipated graduation in 2023.

Mr. Lohr is a 2014 Terry Scholar and a 2018 awardee of the National Science Foundation Graduation Research Fellowship Program (NSF GRFP). He has been a member of Dr. Oleg Komogortsev's research lab at Texas State University since 2015. His research centers around eye tracking with a focus on eye movement biometrics.

**Henry Griffith** (Senior Member, IEEE) was born in Lorain, OH, USA. He received the Ph.D. degree from Michigan State University, East Lansing, MI, USA, the M.S. degree from The University of Dayton, Dayton, OH, USA, and the M.B.A. degree from Wright State University, Dayton, OH, USA.

Dr. Griffith has over 15 years' experience as both an educator and practicing engineer. He is currently an Assistant Professor and Program Coordinator of Engineering at San Antonio College, and a Lecturer in the Department of Electrical and Computer Engineering at the University of Texas at San Antonio.

**Oleg Komogortsev** is currently a Professor of Computer Science at Texas State University. Dr. Komogortsev has received his B.S. in Applied Mathematics from Volgograd State University, and M.S./Ph.D. degree in Computer Science from Kent State University. He was previously a faculty or a scientist at such institutions as Johns Hopkins University, Notre Dame University, Michigan State University, and Meta. Dr. Komogortsev conducts research in eye tracking with a focus on health assessment, cyber security (biometrics), bioengineering, human computer interaction, and usability. This work has thus far yielded multitude of publications and patents.

Dr. Komogortsev's research was covered by the national media including NBC News, Discovery, Yahoo, Livescience and others. Dr. Komogortsev is a recipient of four Google and four Facebook Faculty Research Awards. Dr. Komogortsev has also won National Science Foundation CAREER award and Presidential Early Career Award for Scientists and Engineers (PECASE) from President Barack Obama on the topic of cybersecurity with the emphasis on eye movement-driven biometrics and health assessment. In addition, his research was supported by the National Science Foundation, National Institute of Health, National Institute of Standards, Sigma Xi the Scientific Research Society, and various industrial sources. Dr. Komogortsev's current grand vision is to push forward eye movement-driven user understanding with a very strong privacy backbone in the future virtual and augmented reality platforms.