Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Improving Cognitive-State Analysis from Eye Gaze with Synthetic Eye-Movement Data

Paul Prasse[a,1,*], David R. Reich[a,1], Silvia Makowski[a], Tobias Scheffer[a], Lena A. Jäger[a,b]

[a] *University of Potsdam, Potsdam, Germany*
[b] *University of Zurich, Zurich, Switzerland*

ARTICLE INFO

Article history:
Received February 27, 2024

Keywords: eye tracking, generative adversarial networks, scanpath, gaze generation, reading comprehension, biometric verification, adhd detection, gender classification

ABSTRACT

Eye movements can be used to analyze a viewer's cognitive capacities or mental state. Neural networks that process the raw eye-tracking signal can outperform methods that operate on scan paths preprocessed into fixations and saccades. However, the scarcity of such data poses a major challenge. We therefore develop SP-EyeGAN, a neural network that generates synthetic raw eye-tracking data. SP-EyeGAN consists of Generative Adversarial Networks; it produces a sequence of gaze angles indistinguishable from human ocular micro- and macro-movements. We explore the use of these synthetic eye movements for pre-training neural networks using contrastive learning. We find that pre-training on synthetic data does not help for biometric identification, while results are inconclusive for the detection of ADHD and gender classification. However, for the eye movement-based assessment of higher-level cognitive skills such general reading comprehension, text comprehension, and the distinction of native from non-native readers, pre-training on synthetic eye-gaze data improves the models' performance and even advances the state-of-the-art for reading comprehension. The SP-EyeGAN model, pre-trained on GazeBase, along with the code for developing your own raw eye-tracking machine learning model with contrastive learning, is available at `https://github.com/aeye-lab/sp-eyegan`.

1. Introduction

Eye tracking data has a wide range of applications, including the assessment of linguistic and cognitive skills [1, 2], the detection of conditions such as dyslexia [3, 4, 5] or attention deficit hyperactivity disorder (ADHD) [6], and even identifying individuals based on their unique patterns of eye movements [7, 8]. In the context of biometric identification, using the raw eye-tracking signal of yaw and pitch degrees of visual angles (dva) at the tracker's sampling rate as input to a deep neural network rather than the preprocessed and possibly aggregated saccades and fixations has been shown to improve performance by an order of magnitude and enables the use of shorter input sequences [9, 8, 10, 11].

However, data scarcity is a major challenge for developing such neural networks; collecting eye-tracking data is costly in terms of labor, equipment and time. There is also a risk that personal information such as gender, identity, or ethnicity may be extracted from eye movements, creating a major privacy concern [12, 13, 14]. Both of these problems could potentially be

*Corresponding author: Tel.: +49-331-9773-829
e-mail: `prasse@uni-potsdam.de` (Paul Prasse)
[1] Both authors contributed equally to this research.

mitigated by using synthetic instead of real-world data to train (or pre-train) machine-learning models. Former approaches to generating synthetic eye-tracking data are limited in their ability to create realistic data; most known approaches only generate fixation positions and/or durations [15, 16, 17, 18, 19] or use statistical models with relatively few parameters [20, 21, 22]. Hence, the data generated by these former methods cannot be used to (pre-)train state-of-the-art models that directly process the raw eye-tracking signal.

In computer vision, biometrics, and other fields, the development of generative adversarial networks (GANs) for generating synthetic data has shown promising results [23]. In this paper, we develop *SP-EyeGAN*, a model that creates **S**can **P**aths for **Eye**-tracking data using **GAN**s. This system generates synthetic eye-tracking data that closely mimic real-world data, and that can be used to overcome the challenges of data scarcity and privacy.

We investigate the potential of pre-training with synthethic data on different challenging downstream tasks: assessment of general reading comprehension, text comprehension, text difficulty, nativeness of a reader, biometric verification, gender classification, and ADHD detection. We develop and evaluate two pre-training strategies: (a) fine-tuning different task-specific state-of-the-art neural-network architectures after a task-independent embedding layer has been pre-trained on synthetic gaze data, and (b) training a random-forest classifier that uses the task-independent, pre-trained embedding layer as input features. In both cases, the embedding layer is trained solely on synthetic data generated with SP-EyeGAN using *contrastive learning* [24, 25].

This paper extends a previous conference publication Prasse et al. [26]. Its main contributions are as follows.

- We develop SP-EyeGAN, a model based on generative adversarial networks that generates a sequence of horizontal and vertical gaze angles.
- We show that SP-EyeGAN generates raw eye-tracking data that closely resembles human data, outperforming various statistical and machine-learning-based baseline models.
- We investigate the effect of using synthetic data generated by SP-EyeGAN for pre-training of neural networks on four downstream tasks: assessing general reading comprehension, text comprehension, text difficulty, and deciding whether a reader is a native speaker.

In addition, this extended manuscript makes the following additional original contributions.

- In addition to fine-tuning neural network classifiers whose embedding layer has been pre-trained on syntheric data, we develop and evaluate the strategy of using the embedding layer of a pre-trained network as input features to a random-forest classifier.
- As an additional reference method, we include a random-forest classifier on engineered features showing the classification performance of models not using embeddings.
- We evaluate the random-forest strategy on all downstream tasks, and the neural-network strategy on three downstream tasks, namely biometric identity verification, gender classification, and detection of ADHD, in addition to the tasks included in Prasse et al. [26].
- We investigate the impact of different training set sizes during fine-tuning on the model's performance for the task of gender classification.
- We use a set of five different data sets with different eye tracking devices and different sampling frequencies to investigate how the properties of different data sets effect the usefulness of synthetic data.
- We establish a new state-of-the-art performance on the task of assessing text comprehension and on two out of four videos for ADHD detection.

The rest of this paper is structured as follows. In Section 2, we discuss related work on creating synthetic data using machine learning models. Subsequently, in Section 3, we describe

SP-EyeGAN , a neural-network to create human-like raw eye-tracking data, and how data generated with SP-EyeGAN can be used for contrastive pre-training of a neural network. Section 4 details our experimental results, which are examined for limitations in Section 5. In Section 6, we provide a more extensive analysis and discussion of the results.

2. Related Work

Existing methods for generating human-like eye-tracking data can be divided into training-free statistical models and trained machine-learning models.

Statistical models. Lee et al. [27] and Duchowski and Jörg [28] presented statistical approaches that generate eye movements for rendered, animated faces. Ma and Deng [29] developed a method that synthesizes human eye gaze from a head-motion sequence by statistically modeling the relationship between gaze and head movements. Le et al. [30] generated realistic head motion, eye gaze, and eyelid motion simultaneously based on speech input. Wood et al. [31] presented a method that generates eye crops together with gaze vectors. Yeo et al. [32] proposed a statistical model that generates an eye-tracking sequence of saccades and smooth pursuits for an agent catching a ball. All of these approaches aim at making rendered faces more realistic rather than creating realistic eye-tracking data that include micro- and macro-movements as well as a noise component.

Campbell et al. [20] generated realistic eye-tracking data based on a statistical model of jointly estimated dynamic properties of eye movements for a known saliency map of the stimulus. Duchowski et al. [33, 21] added micro-saccadic jitter, noise, simulated measurement error and pupil unrest to a previously generated eye-tracking sequence. Fuhl and Kasneci [22] and Fuhl et al. [34] simulate saccadic movements by gamma distributions and smooth pursuit onsets with the sigmoid function. EyeSyn, introduced by Lan et al. [35] generates fixational movement using Gaussian and pink noise. We use these two statistical models by Fuhl et al. [34] and Lan et al. [35] as reference models in our evaluation as they are able to generate stimulus independent synthetic fixational and saccadic eye movements.

Machine-learning models. Simon et al. [36] employed a convolutional neural network (CNN) and long short-term memory (LSTM) modules to generate raw eye-tracking samples; this model is limited to generating eye-tracking data for static images. Assens et al. [37] proposed a GAN that consumes images as input and generates fixation points, but is unable to model saccadic movements. Fuhl and Kasneci [38] use a hierarchical k-means algorithm, *HPCGen*, that generates raw eye-tracking data. HPCGen generates random eye-tracking samples not following a specific stimulus with no constant sampling rate, which is not suitable to generate micro-movements and fixations. Fuhl et al. [39] devised a variational autoencoder (VAE) that generates stimulus-independent eye-tracking data. We use this model as one of the baselines in our evaluation as comparison to the previous state-of-the-art neural network approach.

This paper extends our previous work [26] in which we introduced SP-EyeGAN, a model to generate synthetic eye-tracking data using two GANs. SP-EyeGAN generates synthetic eye-tracking data that mimics real-world data, and that can be used in situations where real-world data is limited or unavailable, or privacy concerns don't allow the use of real-world data. This model can generate raw eye-tracking data relative to the head position, and applies to experimental viewing conditions in which the head is kept immobile (e.g., reading on a computer screen with a chin rest), or the viewer is wearing eye-tracking glasses.

3. Method

This section introduces *SP-EyeGAN* a framework to generate synthetic eye movement data, and a contrastive pre-training scheme to use the generated data to pre-train a neural embedding for eye movement sequences.

3.1. SP-EyeGAN

SP-EyeGAN is composed of two independent, structurally identical GANs [40] for generating the raw velocity of fixations

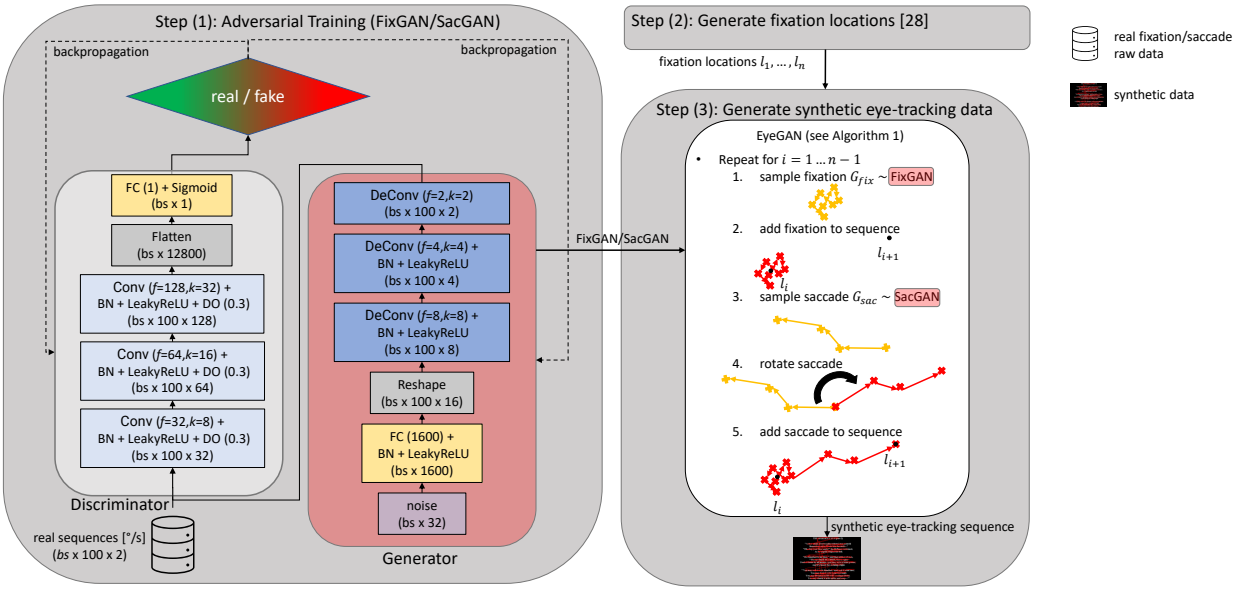


Fig. 1: Overview of SP-EyeGAN. The figure shows the complete pipeline where two GANs are first trained using real data to generate fixations (FixGAN) and saccades (SacGAN) (step 1), and, resorting to generated fixation locations (step 2), are subsequently used to create synthetic data (step 3). The two GANs (SacGAN and FixGAN) consist of fully connected layers (denoted as FC), batch normalization layers (denoted as BN), convolutional/deconvolutional layers (denoted as Conv/DeConv with filter size f, kernel size k and dilation d), and the Leaky Rectified Linear Unit (LeakyReLU) activation function. The batch size (bs) determines the memory consumption of the two GANs. Details about the generation of the synthetic eye-tracking data (step 3) can be found in Algorithm 1

Algorithm 1 The SP-EyeGAN algorithm generates a synthetic eye-movement sequence for given fixation locations.

Require: $\mu_{fix}, \sigma_{fix}, \mu_{sac}, \sigma_{sac}$, FixGAN, SacGAN, fixation locations $F = l_1, \ldots, l_n$
Ensure: Synthetic eye movement sequence $S = s_1, \ldots, s_m$

$S = [\,]$ ▷ start location is first fixation location
for $i \in [1 \ldots n-1]$ **do**
  $a_{sac} = computeSaccadeAmplitude(l_i, l_{i+1})$ ▷ compute saccade amplitude for jump from l_i to l_{i+1}
  $d_{fix} = \mathcal{N}(\mu_{fix}, \sigma_{fix})$ ▷ sample duration for next fixation
  $d_{sac} = \mathcal{N}(\mu_{sac}, \sigma_{sac})$ ▷ sample duration for next saccade
  $G_{fix} = generateFixation(\text{FixGAN}, d_{fix})$ ▷ generate fixation $[°/s]$ with duration d_{fix}
  $S = S + dva(G_{fix})$ ▷ add fixation converted to degrees of visual angle to sequence
  $G_{sac} = generateSaccade(\text{SacGAN}, d_{sac}, a_{sac})$ ▷ generate saccade $[°/s]$ with duration d_{sac} and amplitude a_{sac}
  $G^{rot}_{sac} = rotateSaccade(G_{sac}, S[-1], l_{i+1})$ ▷ rotate generated saccade to end at new fixation location l_{i+1}
  $S = S + dva(G^{rot}_{sac})$ ▷ convert rotated saccade to degrees of visual angle and add to eye movement sequence
end for
$d_{fix} = \mathcal{N}(\mu_{fix}, \sigma_{fix})$ ▷ sample duration for last fixation
$G_{fix} = generateFixation(\text{FixGAN}, d_{fix})$ ▷ generate fixation $[°/s]$ with duration d_{fix}
$S = S + dva(G_{fix})$ ▷ add fixation converted to degrees of visual angle to sequence

(FixGAN) and saccades (SacGAN), respectively, and a module that assembles the generated fixations and saccades into a gaze sequence (see Figure 1). SP-EyeGAN requires a sequence of fixation positions as input. The fixation positions depend on the stimulus and can be generated using any type of model that outputs fixation locations. For example, the fixation locations can be sampled from a saliency map (e.g., for video and image stimuli [41]), or a distribution over word positions [42] for text stimuli. It can also be obtained from a sophisticated cognitive or machine learning model that generates fixation locations on an image or video frame [43, 44] or on a textual stimulus [15, 16, 17, 18, 19].

Both GANs use the same architecture (but do not share parameters) consisting of a discriminator and a generator shown in the left box (step 1) in Figure 1. Both discriminators use the horizontal and vertical velocities of fixations and saccades, respectively, as input. The raw eye movement velocites are computed by computing the change (or displacement) in x- and y-

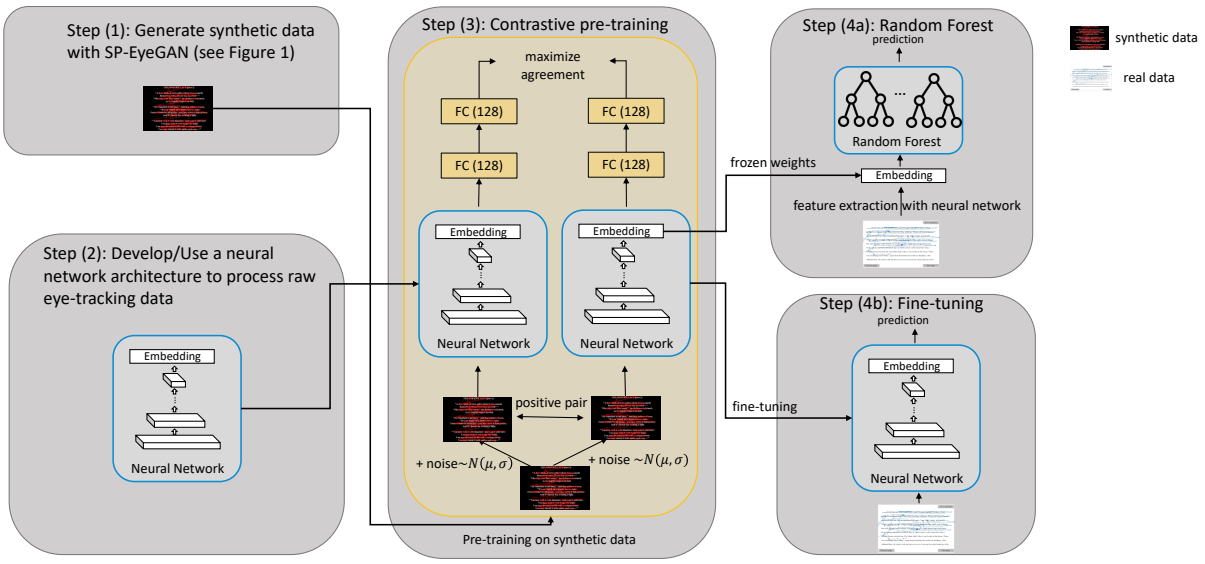


Fig. 2: Pre-training overview. This figure depicts the contrastive pre-training using synthetic data generated by SP-EyeGAN (step 1). The contrastive pre-training is shown in step 3 using any (neural network (step 2). Steps 4a/4b show how the pre-trained models can be used to solve downstream tasks.

positions per millisecond.

FixGAN generates low-amplitude fixational micro-movements, while SacGAN generates fast saccadic movements. Both GANs consist of a generative and a discriminative module. While the generator is used to create synthetic eye movements, the discriminator is trained to distinguish between real and generated synthetic data. Each GAN is trained by alternating the following steps: In the first step, the generator creates synthetic data. Back-propagation is employed to train the discriminator using this data, with the discriminator's performance quantified through cross-entropy loss. The generator's loss is determined by the discriminator's classification performance: it receives a reward when it manages to deceive the discriminator and incurs a penalty when it fails to do so [40].

The generator creates a synthetic eye-movement sequence by projecting a noise vector into a higher dimensional space using a fully-connected layer followed by batch normalization and LeakyReLU activation. This output is then reshaped to match the required sequence length. In this work we used 100 ms for fixations and 30 ms for saccades because this matches the mean durations of typical fixations and saccades in natural viewing tasks [45]. The reshaping layer is followed by three deconvolutional blocks. Each block consists of a deconvolution (filter size f, kernel size k) followed by batch normalization and a LeakyReLU activation.

The discriminator consumes a sequence of eye movements and decides whether this sequence was recorded from a human participant or whether it was synthesized by the generator. It consists of three convolutional blocks. Each convolutional block consists of a convolution (filter size f, kernel size k) followed by batch normalization and a LeakyReLU activation. The output of the last convolutional block is flattened and is fed into a fully connected layer, followed by a sigmoid activation to compute the estimated probability of the eye-movement sequence being real.

SP-EyeGAN, as shown in Algorithm 1, generates a complete eye movement velocity sequence S of fixations and saccades by sampling velocities that constitute fixations and saccades using the trained FixGAN and SacGAN generators. Raw fixation and saccade sequences of different durations can be created by sampling multiple times using the GANs and fusing or cutting the sequences to obtain the desired duration. The algorithm creates a synthetic eye-movement sequence given the mean μ_{fix} and standard deviation σ_{fix} for fixation durations, the mean μ_{sac} and standard deviation σ_{sac} for saccade durations and n fixation lo-

cations $F = l_1 \dots l_n$ as shown in Figure 1 (top right). The means and standard deviations of durations can be estimated on the entire training data. Algorithm 1 samples the fixation and saccade durations using a Normal distribution (lines 4 and 5). For use cases that require highly diverse synthetic data, durations can be sampled by other statistical models or cognitive models. Each sequence starts with fixation velocities on the first fixation location, with fixational movements created by the FixGAN and clipped or extended (line 6) to the sampled fixation duration d_{fix} (line 4). The velocities comprising each fixation are appended to the generated sequence S (line 7). The amplitude a_{sac} of the preceding saccade at iteration i is determined by the distance between the two fixation locations l_i and l_{i+1} (line 3). In the next step, we use the SacGAN generator to generate a saccade matching the amplitude (line 8) and the saccade duration d_{sac} (line 5). This saccade can move in another direction and therefore has to be rotated accordingly (line 9) before being added to the sequence (line 10). The generation ends with sampling the velocities for the last fixation location (lines 12-14).

3.2. Pre-Training and Task-Specific Fine-Tuning

SP-EyeGAN is able to generate raw eye-movement sequences (Figure 2 (step 1)) which can serve to pre-train neural networks (Figure 2 (step 2)) using the self-supervised technique *contrastive learning* [24, 25]. Contrastive learning can be applied without labels for any downstream task, making it a suitable approach for learning representations from synthetic data. The objective of contrastive learning, as shown in Figure 2 (step 3), is for the neural network to be able to differentiate between positive pairs that originate from the same sequence and negative pairs that originate from different sequences. In our study, we create positive pairs by augmenting a base sequence with Gaussian noise twice. Negative pairs are created by augmenting two different sequences with Gaussian noise. The two sequences that constitute a (positive or negative) pair are fed into the neural network, which computes a hidden representation whose dimension is then reduced using two bottleneck layers. The objective of the neural network training during the contrastive learning process is to maximize the agreement between the hidden representations of positive pairs and minimize the agreement in negative pairs.

We investigate two different strategies that exploit the pre-trained embedding for downstream tasks. The first strategy is to use the pre-trained embedding in a task-specific neural-network architecture and fine-tune the model parameters for a specific downstream task using a—potentially smaller—amount of real data labeled for the downstream task (see step 4b in Figure 2). The second strategy under investigation uses the feature embedding of an eye-tracking sequence as input to any machine learning model. In this paper, we use a simple random forest classifier that consumes the feature representation created by the pre-trained neural network (see step 4a in Figure 2).

To summarize, our method is comprised of the following steps:

1. Adversarial training of FixGAN and SacGAN using unlabeled human eye movement data[2];
2. Selection of a model (e.g., a cognitive model) that generates fixation locations for a given stimulus;
3. Generation of synthetic raw eye-tracking data using SP-EyeGAN together with the fixation location model;
4. Development or selection of a neural network architecture suitable to process raw eye-tracking data for the downstream task at hand;
5. Pre-training of the neural network on the synthetic data using contrastive learning;
6. Fine-tuning of the neural network or use of the pre-trained neural embedding as input to any machine learning model (e.g., random forest) on labeled data for the downstream task.

Note that for generating a synthetic gaze sequence there is no need to train the FixGAN or SacGAN from scratch (our trained weights are made publicly available with the code). Hence, for training a model for a new task, only the last step needs to be re-done.

[2]The data is not used for training in the subsequent classification task.

Table 1: Descriptive statistics for used datasets.

Dataset	Number of participants	Eye tracking device	Sampling frequency
GazeBase [46]	322 (151 female, 171 male)	EyeLink 1000	1,000 Hz
SB-SAT [47]	95	EyeLink 1000	1,000 Hz
JuDo1000 [48, 49]	150	EyeLink 1000	1,000 Hz
Gaze on Faces [50]	428 (223 female, 205 male)	EyeLink 1000	60 Hz
HBN [51]	67 (Video 1), 159 (Video 2), 316 (Video 3), 341 (Video 4)	iView-X Red-m (SMI)	120 Hz

3.3. Evaluation Metrics

We evaluate the quality of generated eye movement data by comparing properties commonly used to describe eye tracking data of generated fixations and saccades with the same properties of human eye-movement data [52] using the *Jensen-Shannon divergence*. The Jensen-Shannon divergence measures the similarity between two probability distributions and is a symmetric variant of the Kullback-Leibler divergence; its symmetry makes it more suitable for comparing distributions that may have different shapes or scales. For discrete probability distributions P and Q defined on the same sample space $\mathcal{X}$ the Jensen-Shannon divergence (JSD) is defined as $JSD(P\|Q) = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M)$, where $M = \frac{1}{2}(P + Q)$ and $KL(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log_2\left(\frac{Q(x)}{P(x)}\right)$.

We evaluate the performance of models trained on a downstream task in terms of the area under the receiver operating characteristic curve (AUC) for all classification tasks. The AUC is a quantitative indicator of classification performance. Independently of the class ratios, the AUC ranges from 0 for inverted predictions to 1 for perfect predictions, while 0.5 represents random guessing. For the downstream task of biometric identity verification we evaluate the performance using the equal error rate (EER), which is the rate of false matches and non-matches that is attained when the decision threshold is adjusted such that the risk of incorrectly verifying the identity of an impostor (false match rate) is equal to the risk of incorrectly denying verification (false non-match rate).

3.4. Data

To train SP-EyeGAN, we use eye movement data from a reading experiment taken from the *GazeBase* dataset [46]. *GazeBase* consists of gaze recordings from 322 college-aged participants recorded monocularly with an EyeLink 1000 eye tracker at a sampling frequency of 1,000 Hz. The participants were repeatedly recorded while reading a poem up to nine times over a period of 37 months.

We use the raw eye movement recordings of the Stony Brook Scholastic Assessment Test (*SB-SAT*) dataset [47] for four different downstream tasks(General Reading Comprehension, Text Comprehension, experienced Text Difficulty, and whether the reader is a Native Speaker). *SB-SAT* consists of eye movement data from 95 undergraduate students reading Scholastic Assessment Test (SAT) texts, followed by comprehension questions recorded at a sampling rate of 1,000 Hz.

For the downstream task of biometric identity verification, we use two different datasets recorded on different stimuli. We use the reading part of the *GazeBase* [46] dataset that was recorded while 322 participants were reading a text.[3]The *JuDo1000* [49] dataset consists of recordings from 150 participants who attended four experimental sessions with a lag of at least one week between any two sessions recorded at a sampling rate of 1,000 Hz. In each session, participants were presented with trials in which a black dot appeared consecutively at 5 random screen locations on a light gray background.

For the downstream task of gender classification, we use the *Gaze on Faces* [50] dataset. This dataset consists of recordings of 428 visitors to the Science Museum of London recorded at a sampling rate of 250 Hz. Stimuli consisted of video clips of eight different actors (four females, and four males). Each clip depicted the actor initially gazing toward the bottom of the screen for 500 ms, then gazing up at the participant for a vari-

[3]Please note that for downstream fine-tuning, we exclusively utilize the part of the data on which SP-EyeGAN was trained.

able amount of time, and finally gazing back at the bottom of the screen for 500 ms. We use the publicly available subset of the *Gaze on Faces* dataset that consists of data downsampled to 60 Hz.

For the downstream task of detecting ADHD we use the Healthy Brain Network (HBN) [51] dataset. This dataset consists of recordings from children (mean age 9.97 years ± 3 years) watching four different age-appropriate videos: (1) an educational video clip (*"Fun with Fractals"*), (2) a short animated film (*"The Present"*), (3) a short clip of an animated film (*"Despicable Me"*), and (4) a trailer for a feature-length movie (*"Diary of a Wimpy Kid"*). The eye gaze was recorded at a sampling rate of 120 Hz.

Table 1 shows descriptive statistics and eye-tracking devices used for recording the datasets used in this study.

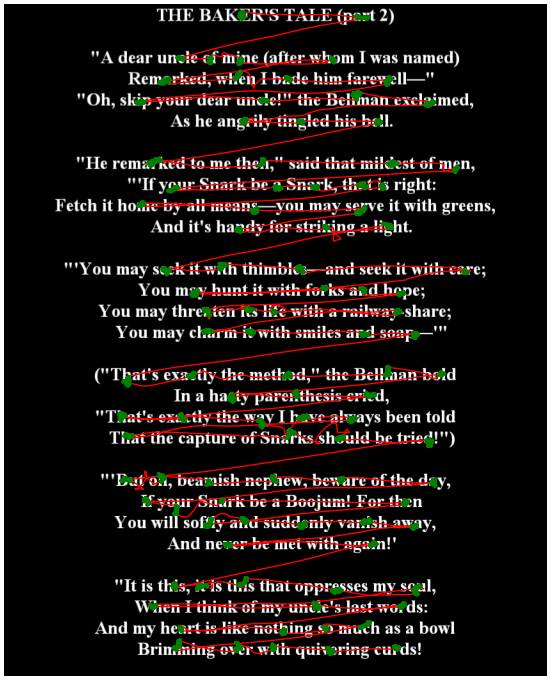


Fig. 3: Generated eye movement sequence using SP-EyeGAN. The fixation locations are sampled using a statistical model [42]. Raw saccadic and fixation samples are shown in red and green, respectively.

4. Results

This section reports on our experimental results. All code to reproduce the results and pre-trained SP-EyeGAN models to create synthetic eye movement data can be found online.[4]

4.1. Synthetic Data Quality

We evaluate the quality of generated synthetic data by comparing generated and real eye-movement events in terms of descriptive features by subsampling real and model-generated eye-movement data 10 times. In order to measure the quality of generated fixations, we calculate the Jensen-Shannon divergence between real and generated fixations in terms of velocities, mean velocities, and dispersion, respectively. The quality of generated saccades is determined by comparing the peak velocity, mean velocity, peak acceleration, mean acceleration, and the amplitude of a saccade. We compare SP-EyeGAN to the statistical models proposed by Fuhl et al. [34] and Lan et al. [35], and the neural network approach of Fuhl et al. [39]. Note that this section focuses solely on examining the two generative components of the SP-EyeGAN architecture: FixGAN for fixations and SacGAN for saccades.

Table 2 compares generated fixations. Note that the statistical model only creates absolute velocities without directions so we cannot compute the dispersions. Our FixGAN model outperforms all the other models and generates fixation profiles that are more similar to fixation profiles of real human eye-movement sequences.

The results for comparing model-generated saccades can be seen in Table 3.[5] From the results we can conclude that SacGAN creates saccades that are more similar to human saccades than the baseline methods. SacGAN performs best for all the attributes under investigation and significantly outperforms all the baseline models in four out of five attributes. Figure 3 shows a synthetic eye movement sequences generated using SP-EyeGAN that are similar to eye movement data generated by humans. SP-EyeGAN generates a sequence consisting of an alternation between fixations and saccades, which can be identified using the Dispersion-Threshold Identification algorithm [53]. To validate the data generated by SP-EyeGAN, we

[4] https://github.com/aeye-lab/sp-eyegan

[5] We cannot compute saccade amplitudes for the statistical model proposed by [34] because it only generates velocities without directions.

Table 2: Quality of generated *fixations* in terms of Jensen-Shannon divergence between human eye movement data and data generated by the model. Values show the Jensen-Shannon divergence ± standard error for 10 resamplings of the data. For reference, the table also shows the divergence between two different sequences of human eye movement data, denoted as *real*. Bold values indicate the best model. An asterisk ("*") denotes models with a performance significantly better than the second best performing model (paired t-test with $\alpha \leq 0.05$).

	Jensen-Shannon divergence ↓		
Method	Velocity	Mean velocity	Dispersion
Statistical model [34]	0.286 ± 0.001	0.692 ± 0.004	–
VAE [39]	0.204 ± 0.001	0.958 ± 0.003	0.742 ± 0.007
EyeSyn [35]	0.065 ± 0.001	0.793 ± 0.006	0.99 ± 0.002
SP-EyeGAN [26]	**0.03 ± 0.001***	**0.321 ± 0.007***	**0.326 ± 0.006***
Real	0.001 ± 0.0	0.073 ± 0.003	0.114 ± 0.004

Table 3: Quality of generated *saccades* in terms of Jensen-Shannon divergence between human data and data generated by the model. Values show the Jensen-Shannon divergence ± standard error for 10 resamplings of the data. For reference, the table also shows the divergence between two different sequences of human eye movement data, denoted as *real*. Bold values indicate the best model. An asterisk ("*") denotes models with a performance significantly better than the second best performing model (paired t-test with $\alpha \leq 0.05$).

	Jensen-Shannon divergence ↓				
Method	Peak velocity	Mean velocity	Peak acceleration	Mean acceleration	Amplitude
Statistical model [34]	0.426 ± 0.007	0.253 ± 0.007	0.922 ± 0.003	0.859 ± 0.004	–
VAE [39]	0.951 ± 0.004	0.906 ± 0.007	0.895 ± 0.003	0.871 ± 0.006	0.92 ± 0.008
SP-EyeGAN [26]	**0.399 ± 0.006***	**0.251 ± 0.004**	**0.293 ± 0.004***	**0.24 ± 0.004***	**0.267 ± 0.006***
Real	0.079 ± 0.005	0.052 ± 0.003	0.254 ± 0.007	0.046 ± 0.002	0.075 ± 0.003

checked the overlap between saccades and fixations identified using this algorithm and the output of SP-EyeGAN generating fixations and saccades. To do so, we sampled 10 scan paths for a text and checked for each training sample whether the outcome of the Dispersion-Threshold Identification algorithm matches the label assigned by SP-EyeGAN. We observe an accuracy of 0.971 ± 0.001, indicating that scan paths generated by SP-EyeGAN are plausible.

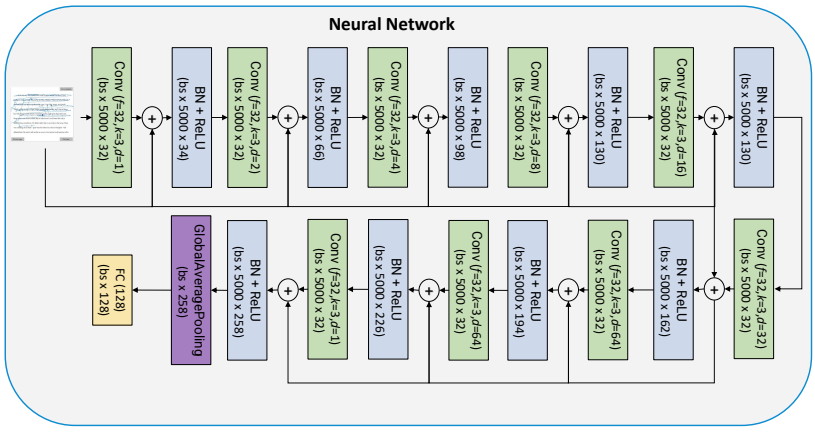
4.2. Evaluation on Downstream Tasks

In order to quantify the benefit of pre-training a model using synthetic data generated by SP-EyeGAN, we investigate the effectiveness on five publicly available datasets (see Table 1). Remember that SP-EyeGAN is trained on reading data extracted from *GazeBase* at 1,000 Hz. This means that we are creating data with a resolution of 1,000 Hz using Algorithm 1. To get lower sampling rates we simply downsample the data using linear interpolation.

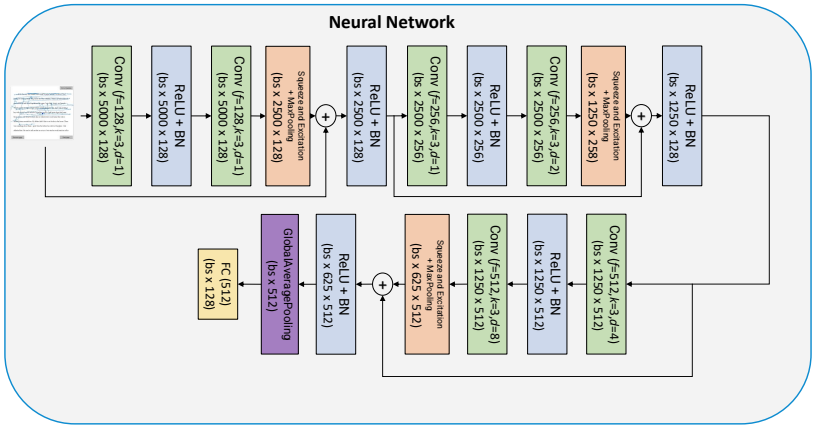
For all the downstream tasks we compare the performance of different deep neural models [8, 25] that can process raw eye-tracking data, and, for the task of biometric identity verification, have been shown to perform exceptionally well: CLRGaze [25] and EKYT [8] (see Figure 4). We evaluate the performance of these models on all downstream tasks in three settings: (1) when being trained from scratch on human data without any pre-training, (2) when being first pre-trained on synthetic eye movement sequences generated by SP-EyeGAN and then fine-tuned on the human data, and (3) when being pre-trained on synthetic eye movements generated by SP-EyeGAN and then used to compute feature embeddings of the human data which are then used as input to train a random forest. Furthermore, we compared against a baseline using a random forest on engineered features reported in the literature [54]. This baseline extracts features from eye tracking sequences by detecting the fixations and saccades and computing features like the saccade durations, peak velocities, gaze entropy, and many more [55, 56, 57, 58, 54].

4.2.1. Stony Brook Scholastic Assessment Test

This section reports on the results using the SB-SAT dataset for four different downstream tasks. The labels extracted for the different tasks are: overall comprehension score across all passages (General Reading Comprehension), text-based comprehension accuracy (Text Comprehension), a subjective difficulty



(a) EKYT model used for downstream tasks. See Lohr and Komogortsev [8] for more details.



(b) CLRGaze model used for downstream tasks. See Bautista and Naval [25] for more details.

Fig. 4: Neuronal Networks used to pre-train and fine-tune a model on the downstream task. Figure 4a and4b depict the model architectures trained with batch size *bs* consisting of fully connected layers (denoted as FC), batch normalization (denoted as BN), convolutional layers (denoted as Conv with filter size f, kernel size k and dilation d), and the rectified linear unit (ReLU) activation function. The numbers in brackets show the dimensions of the data after each layer.

Table 4: Comparison of a diverse set of downstream tasks on the SB-SAT dataset. AUC ± standard error reported for 5-fold CV. The model in brackets denotes the model used during pre-training/ fine-tuning. An asterisk ("*") denotes models with a performance significantly better than random guessing and a pre-trained model marked with a dagger ("†") indicates a model that is significantly better than its variant trained from scratch (paired t-test with $\alpha \leq 0.05$).

	Task			
Method	General Reading Comprehension	Text Comprehension	Text Difficulty	Native Reader
BEyeLSTM	0.608 ± 0.037*	0.542 ± 0.015*	**0.710 ± 0.017***	0.670 ± 0.025*
Random forest on engineered features	0.619 ± 0.039*	0.584 ± 0.039*	0.544 ± 0.013*	**0.737 ± 0.057***
Without pretraining (EKYT)	0.585 ± 0.015*	0.566 ± 0.020*	0.494 ± 0.021	0.550 ± 0.014*
Pre-training & fine-tuning (EKYT)	**0.622 ± 0.029***	0.574 ± 0.024*	0.545 ± 0.006*†	0.721 ± 0.061*†
Random forest on feature embeddings (EKYT)	0.575 ± 0.074	**0.597 ± 0.026***	0.501 ± 0.012	0.556 ± 0.057
Without pre-training (CLRGaze)	0.569 ± 0.065	0.560 ± 0.055	0.516 ± 0.034	0.528 ± 0.046
Pre-training & fine-tuning (CLRGaze)	0.577 ± 0.033	0.592 ± 0.032*	0.566 ± 0.018*	0.704 ± 0.050*†
Random forest on feature embeddings (CLRGaze)	0.507 ± 0.05	0.562 ± 0.026*	0.488 ± 0.016	0.577 ± 0.042

rating (Text Difficulty), and whether the presented text was the native (L1) language of the reader (Native Reader).

We compare the results obtained with our approach in combination with EKYT and CLRGaze to the performance of i) the current state-of-the-art BEyeLSTM model [1] (which is not able to process raw data, but only preprocessed fixations) and ii) the random forest on engineered features. We apply 5-fold cross-validation splitting the data such that no reader is simul-

Table 5: Equal error rates ± standard error for biometric verification on the GazeBase data. An asterisk ("*") denotes models with a performance significantly lower than 0.5 ($\alpha < 0.05$).

Method	EER
Without pre-training (EKYT)	**0.165 ± 0.004***
Pre-training & fine-tuning (EKYT)	0.169 ± 0.003*
Zero-shot with pre-training (EKYT)	0.495 ± 0.004
Without pre-training (CLRGaze)	0.181 ± 0.003*
Pre-training & fine-tuning (CLRGaze)	0.188 ± 0.003*
Zero-shot with pre-training (CLRGaze)	0.493 ± 0.003

Table 6: Equal error rates ± standard error for biometric verification on JuDo1000. An asterisk ("*") denotes models with a performance significantly lower than 0.5 ($\alpha < 0.05$).

Method	EER
Without pre-training (EKYT)	0.112 ± 0.003*
Pre-training & fine-tuning (EKYT)	0.114 ± 0.003*
Zero-shot with pre-training (EKYT)	0.49 ± 0.002*
Without pre-training (CLRGaze)	**0.109 ± 0.002***
Pre-training & fine-tuning (CLRGaze)	0.124 ± 0.004*
Zero-shot with pre-training (CLRGaze)	0.462 ± 0.002*

taneously included in both the training and test portions of the data [1]. Note that splitting along readers has been found to be the more challenging evaluation setting since it assesses the models' ability to generalize to new readers [59, 1].

An overview of the results can be found in Table 4. We find that a fine-tuned model that is based on a previously contrastively pre-trained model significantly improves over models trained without pre-training in three cases and numerically (though not significantly) improves the performance in the remaining five cases. When using the pre-trained network to compute neural feature representations (embeddings) of the human data, even a simple random forest establishes a new state-of-the-art for the Text Comprehension task. This highlights the utility of the learned embeddings. The random forest on engineered features establishes a new state of the art for the Native Reader classification task and outperforms the neural feature representations in all settings. In summary, for two out of four downstream tasks, a model using synthetic data generated by SP-EyeGAN establishes a new state-of-the-art. For one downstream task (Text Difficulty), BEyeLSTM—that processes engineered features of the fixated text which our models have no access to—remains the state of the art.

4.2.2. Biometric Verification

This section reports on the results of biometric verification using two state-of-the-art models: EKYT and CLRGaze. To evaluate the models we resample five times from the complete dataset. In each iteration, a training population of 100 subjects is selected and the remaining subjects (222 for GazeBase, 50 for JuDo1000) are used for evaluation. At application time, one subject is enrolled by calculating and storing the mean embedding of the input data (80 randomly selected training instances per subject), taken from the first session. Each subject serves as a test subject: a probe sequence of five seconds, taken from another recording session, is compared against the enrollment embedding. For each dataset, we compare a model without pre-training with a fine-tuned model that was contrastively pre-trained on data created by SP-EyeGAN, and a zero-shot model without fine-tuning. This model uses the embeddings created by the contrastively pre-trained model.[6]

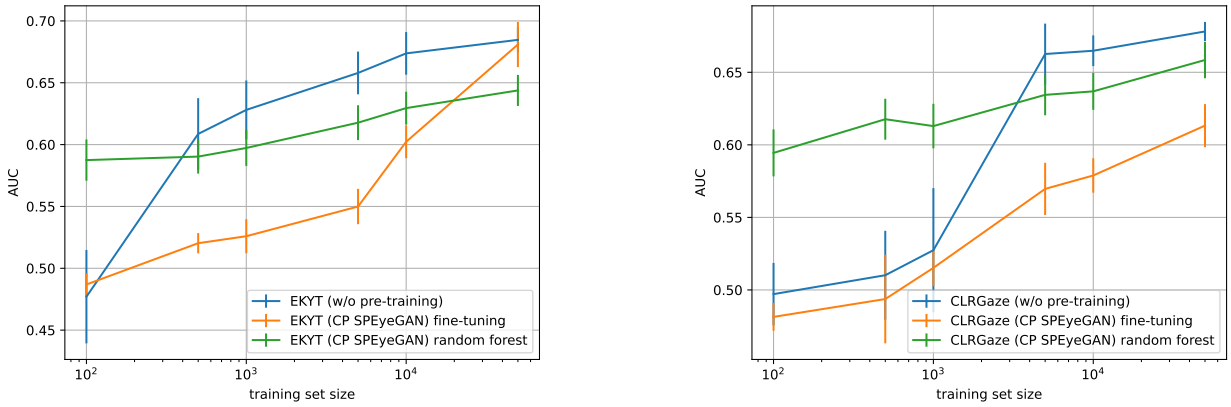
On both datasets, we observe that models pre-trained on data created by SP-EyeGAN perform worse than a model trained from scratch on human data without pre-training (see Table 5-5). For the GazeBase dataset, we observe that the zero-shot version does not perform significantly better than random guessing, whereas it performs significantly better for the JuDo1000 dataset.

4.2.3. Gender Classification

This section reports on the results for gender classification on the Gaze on Faces dataset. Since there is no state-of-the-art gender classification model that uses sequences of raw eye-tracking data, we evaluate EKYT and CLRGaze using 5-fold cross-validation splitting by subjects to make sure that the subjects in the training and test data are different and compare it to the random forest on engineered features.

According to the results presented in Table 7, models that have been contrastively pre-trained on synthetic data do not

[6] We cannot apply the random forest because it is not possible to train a classifier to distinguish between classes (here: participant identities) not exposed to during training.



(a) EKYT model.

(b) CLRGaze mode.

Fig. 5: AUC ± standard error for gender classification over the numer of labeled training instances on the Gaze on Faces dataset. The x-axis shows the number of instances of labeled human eye-tracking.

Table 7: AUC ± standard error for gender classificaton on the Gaze on Faces dataset, model names in parentheses. An asterisk ("*") denotes models with a performance significantly better than random guessing ($\alpha < 0.05$).

Method	**AUC**
Random forest on engineered features	**0.68 ± 0.02***
Without pre-training (EKYT)	**0.68 ± 0.01***
Pre-training & fine-tuning (EKYT)	**0.68 ± 0.01***
Random forest on feature embeddings (EKYT)	0.65 ± 0.01*
Without pre-training (CLRGaze)	0.66 ± 0.01*
Pre-training & fine-tuning (CLRGaze)	0.6 ± 0.02*
Random forest on feature embeddings (CLRGaze)	0.66 ± 0.01*

have any advantage over models without pre-training. We observe that neural networks on raw eye-tracking data perform similarly well as a model using engineered features. All models perform significantly better than random guessing ($\alpha \leq 0.05$).

The Gaze on Faces dataset is the largest dataset under investigation in terms of training instances, allowing us to investigate the influence of the number of training instances used for training. The results for different training set sizes are shown in Figure 5. We can conclude that for small training set sizes, the random forest that uses the pre-trained feature embeddings performs best. For 100 training instances with the EKYT model and for up to 1,000 instances with the CLRGaze model the random forest model performs significantly better than a fine-tuned neural network or a neural network that has not been pre-trained.

4.2.4. ADHD Detection

This section reports on the results of ADHD detection on the HBN dataset. We compare each model to the current state-of-the-art model [6], denoted as *SOTA CNN*, that uses a CNN architecture processing gaze events and saliency maps of the videos, and a random forest on engineered features. In contrast to the SOTA CNN, the models that we employ as the neural network component (step 2 in Figure 2) only use the raw eye-tracking data without any stimulus information. To compare to the state-of-the-art model we follow the same evaluation protocol as described in Deng et al. [6]: We randomly resample the data 10 times and in each iteration perform 10-fold cross-validation along subjects, such that no individual occurs in both the training and test data.

The results presented in Table 8 are inconclusive. The *SOTA CNN* [6] with the stimulus video clip *Fun with Fractals* remains the combination that allows for the most accurate detection of ADHD. For *Fun with Fractals*, only the random forest on the pre-trained embedding of CLRGaze improves over CLRGaze without pre-training. For *Despicable Me*, the random forest on EKYT embeddings improves over EKYT without pre-training and beats the SOTA CNN. For *Diary of a Wimpy Kid*, pre-training on synthetic gaze data improves the results for CLRGaze and the random forest on CLRGaze embeddings and again outperforms the SOTA CNN. In other cases, pre-training

Table 8: AUC ± standard error for ADHD detection with model names in parentheses. An asterisk ("*") denotes models with a performance significantly better than random guessing and a dagger ("†") marks model that are significantly better than their variant without pre-training.

	AUC			
Method	Fun with Fractals	The Present	Despicable me	Diary of a Wimpy Kid
SOTA CNN [6]	**0.646 ± 0.025***	**0.554 ± 0.016***	0.544 ± 0.01*	0.503 ± 0.01
Random forest on engineered features	0.587 ± 0.026*	0.407 ± 0.014	0.541 ± 0.009*	0.485 ± 0.009
Without pre-training (EKYT)	0.552 ± 0.026*	0.49 ± 0.016	0.544 ± 0.011*	0.513 ± 0.01
Pre-training & fine-tuning (EKYT)	0.516 ± 0.028	0.43 ± 0.017	0.533 ± 0.012*	0.509 ± 0.01
Random forest on feature embeddings (EKYT)	0.606 ± 0.025*	0.375 ± 0.015	**0.581 ± 0.01***†	0.467 ± 0.009
Without pre-training (CLRGaze)	0.49 ± 0.026	0.494 ± 0.017	0.549 ± 0.011*	0.494 ± 0.011
Pre-training & fine-tuning (CLRGaze)	0.514 ± 0.023	0.402 ± 0.016	0.541 ± 0.011*	0.528 ± 0.01*†
Random forest on feature embeddings (CLRGaze)	0.626 ± 0.024*†	0.368 ± 0.015	0.574 ± 0.011*	**0.545 ± 0.01***†

does not improve the model's accuracy. For three out of four videos, a random forest trained on a pre-trained embedding performs better than its counterpart trained on engineered features, highlighting the expressive power of the learned embeddings.

5. Limitations

Although SP-EyeGAN shows promising results in generating synthetic scan paths, there are still limitations to consider.

In the current implementation, SP-EyeGAN is not able to process the viewed stimulus as input and hence does not take it into account for the generation of the scan path. However, we would like to emphasize that the statistical model that we use for the generation of fixation locations can be replaced by any other model that generates fixation locations, including models that do take into account the stimulus such as cognitive models of scene viewing or reading [42, 43, 15, 16, 17, 18, 19]. In order to achieve this interaction, the stimulus would need to be used as input to the FixGAN and SacGAN, which could be implemented as stimulus-conditioned GANs. Similarly, rather than simply sampling fixation and saccade durations from the training data, one might consider estimating them relative to both the stimulus and the location of fixation.

Hence our approach is limited to stimuli that do not evoke smooth pursuits, which are an important oculomotor event that occurs while following a slowly moving stimulus. Given training data containing smooth pursuits, it is straightforward to extend our model by a third GAN component to include smooth pursuit movements.

Despite these limitations, our model represents a significant step forward in using machine learning to generate synthetic raw eye-tracking data. Future studies can build on our work to address these limitations and further improve the accuracy and generalizability of models generating eye movement data.

6. Discussion and Conclusion

We have introduced SP-EyeGAN, a method that generates realistic raw eye-tracking data. We investigated the utility of using synthetic data generated by SP-EyeGAN. Fixational micro-movements can be generated around fixation locations taken from any model of eye movement control—be it a statistical model, a machine-learning-based model, or a cognitive model. SP-EyeGAN connects these fixations with realistic saccadic movements. The synthetic raw eye gaze sequences can be used to pre-train any neural network that is designed to process raw eye movement data for any downstream task. In this pre-training step, the neural network learns to compute informative neural representations of eye movement sequences, so-called embeddings, independently of the downstream task. In a final step, the neural network can be fine-tuned with human eye-tracking data for any downstream task of the researcher's choice.

We also explored the possibility of using features generated by the pre-trained model as input to a random forest. This embedding-based workflow has the advantage that it can be used in situations where only a little human data for the downstream task is available.

We investigated the performance on seven downstream prediction tasks that have recently attracted attention in eye-tracking research. Although we used neural network architectures that were originally developed for the task of biometric verification, we found that pre-training on SP-EyeGAN-generated synthetic data improved their performance significantly in several of the investigated tasks and settings. For general reading comprehension, text comprehension, and detection of native readers, pre-training on synthetic data improves the models' performance. On the other hand, we also identified downstream tasks for which SP-EyeGAN is not suitable. For biometric verification, the use of a pre-trained model deteriorates the overall performance. We hypothesize that this might be due to the fact that during the pre-training, the model learns to extract generic, subject-independent patterns by abstracting away from idiosyncracies in the eye-tracking signal, which stands in contrast to learning behavioral biometric traits of the individual participants.

To date, most research has been focusing on methods that operate on preprocessed scan paths of fixations and saccades, often using engineered fixational and saccadic features. To compare against a baseline operating on such features extracted from fixations and saccades, we used a random forest on engineered features. Recent research in eye-tracking-based biometrics [9], however, has shown that the raw eye-tracking signal contains valuable information that is lost when preprocessing the data. Since neural networks that are designed to process raw eye-tracking data have an even larger number of parameters than neural networks operating on preprocessed data, data scarcity is a major obstacle to the development of such models. Our proposed approach opens the possibility of developing deep neural networks with large numbers of parameters since potentially infinite amounts of synthetic data are available for (pre)-training.

Besides our approach's advantages for training neural networks, it has also important advantages for *privacy*. In recent years, it has been shown that in many cases, it is possible to reconstruct the training data from a neural network's final parameters [60], which can violate the privacy of donors of training data: it may be possible to infer the training users' identity, gender or other sensitive attributes [8, 7, 61]. The inclusion of synthetic training data dilutes any potentially identifiable traits.

Acknowledgments

This work was partially funded by the German Federal Ministry of Education and Research under grant 01| S20043.

References

[1] Reich, DR, Prasse, P, Tschirner, C, Haller, P, Goldhammer, F, Jäger, LA. Inferring Native and Non-Native Human Reading Comprehension and Subjective Text Difficulty from Scanpaths in Reading. In: Proceedings of the 2023 Symposium on Eye Tracking Research and Applications. New York, USA: Association for Computing Machinery; 2022, p. 1–8.

[2] Berzak, Y, Katz, B, Levy, R. Assessing language proficiency from eye movements in reading. In: Walker, M, Ji, H, Stent, A, editors. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018, p. 1986–1996. URL: https://aclanthology.org/N18-1180. doi:10.18653/v1/N18-1180.

[3] Nilsson Benfatto, M, Öqvist Seimyr, G, Ygge, J, Pansell, T, Rydberg, A, Jacobson, C. Screening for dyslexia using eye tracking during reading. PloS one 2016;11(12):e0165508.

[4] Björnsdóttir, M, Hollenstein, N, Barrett, M. Dyslexia prediction from natural reading of Danish texts. In: Alumäe, T, Fishel, M, editors. Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa). Tórshavn, Faroe Islands: University of Tartu Library; 2023, p. 60–70. URL: https://aclanthology.org/2023.nodalida-1.7.

[5] Haller, P, Säuberli, A, Kiener, SE, Pan, J, Yan, M, Jäger, LA. Eye-tracking based classification of Mandarin Chinese readers with and without dyslexia using neural sequence models. In: Proceedings of the Workshop on Text Simplification, Accessibility, and Readability. Abu Dhabi, UAE: Association for Computational Linguistics; 2022, p. 111–118.

[6] Deng, S, Prasse, P, Reich, DR, Dziemian, S, Stegenwallner-Schütz, M, Krakowczyk, D, et al. Detection of ADHD based on eye movements during natural viewing. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Grenoble, France: Springer; 2022, p. 403–418.

[7] Makowski, S, Prasse, P, Reich, DR, Krakowczyk, D, Jäger, LA, Scheffer, T. DeepEyedentificationLive: Oculomotoric Biometric Identification and Presentation-Attack Detection using Deep Neural Networks. IEEE Transactions on Biometrics, Behavior, and Identity Science 2021;3(4):506–518.

[8] Lohr, D, Komogortsev, OV. Eye Know You Too: Toward Viable End-to-End Eye Movement Biometrics for User Authentication. IEEE Transactions on Information Forensics and Security 2022;17:3151–3164.

[9] Jäger, LA, Makowski, S, Prasse, P, Liehr, S, Seidler, M, Scheffer, T. Deep Eyedentification: Biometric identification using micro-movements of the eye. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. 2020, p. 299–314.

[10] Unger, M, Wedel, M, Tuzhilin, A. Predicting consumer choice from raw eye-movement data using the retina deep learning architecture. Available at SSRN 4341410 2023;.

[11] Qin, H, Zhu, H, Jin, X, Song, Q, El-Yacoubi, MA, Gao, X. Emmixformer: Mix transformer for eye movement recognition. arXiv preprint arXiv:240104956 2024;.

[12] Prasse, P, Reich, DR, Makowski, S, Jäger, LA, Scheffer, T. Fairness in oculomotoric biometric identification. In: Proceedings of the 2022 Symposium on Eye Tracking Research and Applications. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392525; 2022,.

[13] David-John, B, Butler, K, Jain, E. For your eyes only: Privacy-preserving eye-tracking datasets. In: 2022 Symposium on Eye Tracking Research and Applications. ETRA '22; New York, NY, USA: Association for Computing Machinery. ISBN 9781450392525; 2022,URL: https://doi.org/10.1145/3517031.3529618. doi:10.1145/3517031.3529618.

[14] Aziz, S, Komogortsev, O. Assessing the privacy risk of cross-platform identity linkage using eye movement biometrics. 2024. arXiv:2402.08655.

[15] Reichle, E, Rayner, K, A, P. The E-Z reader model of eye-movement control in reading: comparisons to other models. The Behavioral and Brain Sciences 2003;26:445–526.

[16] Kümmerer, M, Bethge, M. State-of-the-art in human scanpath prediction. 2021. arXiv:2102.12239.

[17] Engbert, R, Nuthmann, A, Richter, EM, Kliegl, R. SWIFT: a dynamical model of saccade generation during reading. Psychological review 2005;112(4):777.

[18] Deng, S, Reich, DR, Prasse, P, Haller, P, Scheffer, T, Jäger, LA. Eyettention: An attention-based dual-sequence model for predicting human scanpaths during reading. Proceedings of ACM Human-Computer Interaction 2023;7:1–24.

[19] Bolliger, L, Reich, D, Haller, P, Jakobi, D, Prasse, P, Jäger, L. ScanDL: A diffusion model for generating synthetic scanpaths on texts. In: Bouamor, H, Pino, J, Bali, K, editors. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics; 2023, p. 15513–15538. URL: https://aclanthology.org/2023.emnlp-main.960. doi:10.18653/v1/2023.emnlp-main.960.

[20] Campbell, DJ, Chang, J, Chawarska, K, Shic, F. Saliency-based bayesian modeling of dynamic viewing of static scenes. In: Proceedings of the Symposium on Eye Tracking Research and Applications. 2014, p. 51–58.

[21] Duchowski, AT, Jörg, S, Allen, TN, Giannopoulos, I, Krejtz, K. Eye movement synthesis. In: Proceedings of the Symposium on Eye Tracking Research and Applications. 2016, p. 147–154.

[22] Fuhl, W, Kasneci, E. Eye movement velocity and gaze data generator for evaluation, robustness testing and assess of eye tracking software and visualization tools. 2018. arXiv:1808.09296.

[23] Bowles, C, Chen, L, Guerrero, R, Bentley, P, Gunn, R, Hammers, A, et al. GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks. 2018. arXiv:1810.10863.

[24] Chen, T, Kornblith, S, Norouzi, M, Hinton, G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. 2020, p. 1597–1607.

[25] Bautista, LGC, Naval, PC. CLRGaze: Contrastive Learning of Representations for Eye Movement Signals. In: Proceedings of the 29th European Signal Processing Conference. 2021, p. 1241–1245.

[26] Prasse, P, Reich, DR, Makowski, S, Ahn, S, Scheffer, T, Jäger, LA. SP-EyeGAN: Generating synthetic eye movement data with generative adversarial networks. In: Proceedings of the 2023 Symposium on Eye Tracking Research and Applications. 2023, p. 1–9.

[27] Lee, SP, Badler, JB, Badler, NI. Eyes Alive. In: Proceedings of the Conference on Computer Graphics and Interactive Techniques. New York, NY, USA: Association for Computing Machinery; 2002, p. 637–644.

[28] Duchowski, AT, Jörg, S. Modeling physiologically plausible eye rotations. In: Proceedings of Computer Graphics International. 2015, p. 1–3.

[29] Ma, X, Deng, Z. Natural eye motion synthesis by modeling gaze-head coupling. In: Proceedings of the 2009 IEEE Virtual Reality Conference. Los Alamitos, CA, USA: IEEE Computer Society; 2009, p. 143–150.

[30] Le, B, Ma, X, Deng, Z. Live speech driven head-and-eye motion generators. IEEE Transactions on Visualization and Computer Graphics 2012;18(11):1902–1914.

[31] Wood, E, Baltrusaitis, T, Zhang, X, Sugano, Y, Robinson, P, Bulling, A. Rendering of eyes for eye-shape registration and gaze estimation. In: Proceedings of the 2015 IEEE International Conference on Computer Vision. 2015, p. 3756–3764.

[32] Yeo, SH, Lesmana, M, Neog, DR, Pai, DK. Eyecatch: Simulating visuomotor coordination for object interception. ACM Transactions on Graphics 2012;31(4):1–10.

[33] Duchowski, A, Jörg, S, Lawson, A, Bolte, T, undefinedwirski, L, Krejtz, K. Eye movement synthesis with 1/f pink noise. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games. New York, NY, USA: Association for Computing Machinery; 2015, p. 47–56.

[34] Fuhl, W, Santini, T, Kuebler, T, Castner, N, Rosenstiel, W, Kasneci, E. Eye movement simulation and detector creation to reduce laborious parameter adjustments. 2018. arXiv:1804.00970.

[35] Lan, G, Scargill, T, Gorlatova, M. Eyesyn: Psychology-inspired eye movement synthesis for gaze-based activity recognition. In: Proceedings of the 2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks. 2022, p. 233–246.

[36] Simon, D, Sridharan, S, Sah, S, Ptucha, R, Kanan, C, Bailey, R. Automatic scanpath generation with deep recurrent neural networks. In: Proceedings of the ACM Symposium on Applied Perception. 2016, p. 130–130.

[37] Assens, M, Giro-i Nieto, X, McGuinness, K, O'Connor, NE. PathGAN: Visual scanpath prediction with generative adversarial networks. In: Proceedings of the European Conference on Computer Vision Workshops. 2019, p. 406–422.

[38] Fuhl, W, Kasneci, E. HPCGen: Hierarchical k-means clustering and level based principal components for scan path genaration. In: Proceedings of the 2022 Symposium on Eye Tracking Research and Applications. 2022, p. 1–7.

[39] Fuhl, W, Rong, Y, Kasneci, E. Fully convolutional neural networks for raw eye tracking data segmentation, generation, and reconstruction. In: Proceedings of the 25th International Conference on Pattern Recognition. Los Alamitos, CA, USA: IEEE Computer Society; 2021, p. 142–149.

[40] Goodfellow, I, Pouget-Abadie, J, Mirza, M, Xu, B, Warde-Farley, D, Ozair, S, et al. Generative Adversarial Nets. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2014, p. 2672–2680.

[41] Fosco, C, Newman, A, Sukhum, P, Zhang, YB, Zhao, N, Oliva, A, et al. How much time do you have? modeling multi-duration saliency. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020, p. 4472–4481. doi:10.1109/CVPR42600.2020.00453.

[42] Rayner, K, McConkie, GW. What guides a reader's eye movements? Vision research 1976;16(8):829–837.

[43] Nuthmann, A, Smith, TJ, Engbert, R, Henderson, JM. CRISP: a computational model of fixation durations in scene viewing. Psychological Review 2010;117(2):382.

[44] Kümmerer, M, Bethge, M, Wallis, TSA. DeepGaze III: Modeling free-viewing human scanpaths with deep learning. Journal of Vision 2022;22(5):7–7. URL: https://doi.org/10.1167/jov.22.5.7. doi:10.1167/jov.22.5.7. arXiv:https://arvojournals.org/arvo/content_public/journal/jov/938

[45] Nyström, M, Holmqvist, K. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. Behavior research methods 2010;42(1):188–204.

[46] Griffith, H, Lohr, D, Abdulin, E, Komogortsev, O. GazeBase, a large-scale, multi-stimulus, longitudinal eye movement dataset. Scientific Data 2021;8(1):1–9.

[47] Ahn, S, Kelton, C, Balasubramanian, A, Zelinsky, G. Towards predicting reading comprehension from gaze behavior. In: Proceedings of the ACM Symposium on Eye Tracking Research and Applications. Stuttgart, Germany: Association for Computing Machinery; 2020, p. 1–5.

[48] Makowski, S, Jäger, LA, Prasse, P, Scheffer, T. Biometric identification and presentation-attack detection using micro-movements of the eyes. In: Proceedings of the 2020 IEEE International Joint Conference on Biometrics. 2020, p. 1–10.

[49] Makowski, S, Jäger, LA, Scheffer, T. JuDo1000 eye tracking data set. https://osf.io/5zpvk/; 2020.

[50] Coutrot, A, Binetti, N, Harrison, C, Mareschal, I, Johnston, A. Face exploration dynamics differentiate men and women. Journal of Vision 2016;16(14):16–16.

[51] Alexander, LM, Escalera, J, Ai, L, Andreotti, C, Febre, K, Mangone, A, et al. An open resource for transdiagnostic research in pediatric mental health and learning disorders. Scientific data 2017;4(1):1–26.

[52] Manning, C, Schütze, H. Foundations of statistical natural language processing. MIT press; 1999.

[53] Salvucci, DD, Goldberg, JH. Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 Symposium on Eye Tracking Research & Applications. New York, NY, USA: Association for Computing Machinery; 2000, p. 71–78.

[54] Makowski, S, Bätz, A, Prasse, P, Jäger, LA, Scheffer, T. Detection of alcohol inebriation from eye movements. Procedia Computer Science

2023;225:2086–2095.
[55] Schleicher, R, Galley, N, Briest, S, Galley, L. Blinks and saccades as indicators of fatigue in sleepiness warnings: looking tired? Ergonomics 2008;51(7):982–1010.
[56] Rigas, I, Friedman, L, Komogortsev, O. Study of an extensive set of eye movement features: Extraction methods and statistical analysis. Journal of Eye Movement Research 2018;11(1).
[57] Shiferaw, BA, Crewther, DP, Downey, LA. Gaze entropy measures detect alcohol-induced driver impairment. Drug and Alcohol Dependence 2019;204:107519.
[58] Doyle, M, Walker, R. Curved saccade trajectories: Voluntary and reflexive saccades curve away from irrelevant distractors. Experimental Brain Research 2001;139(3):333–344.
[59] Makowski, S, Jäger, LA, Abdelwahab, A, Landwehr, N, Scheffer, T. A discriminative model for identifying readers and assessing text comprehension from eye movements. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2019, p. 209–225.
[60] Carlini, N, Tramer, F, Wallace, E, Jagielski, M, Herbert-Voss, A, Lee, K, et al. Extracting training data from large language models. In: Proceedings of the 30th USENIX Security Symposium. 2021, p. 2633–2650.
[61] Lahey, JN, Oxley, DR. Discrimination at the intersection of age, race, and gender: Evidence from an eye-tracking experiment. Journal of Policy Analysis and Management 2021;40(4):1083–1119.