Reinforcement learning for the privacy preservation and manipulation of eye tracking data

Wolfgang Fuhl, Efe Bozkir, Enkelejda Kasneci
Human Computer Interaction
Sand 14, 72076 Tubingen, Germany
wolfgang.fuhl@uni-tuebingen.de, efe.bozkir@uni-tuebingen.de, enkelejda.kasneci@uni-tuebingen.de

Abstract

In this paper, we present an approach based on reinforcement learning for eye tracking data manipulation. It is based on two opposing agents, where one tries to classify the data correctly and the second agent looks for patterns in the data, which get manipulated to hide specific information. We show that our approach is successfully applicable to preserve the privacy of a subject. For this purpose, we evaluate our approach iteratively to showcase the behavior of the reinforcement learning based approach. In addition, we evaluate the importance of temporal, as well as spatial, information of eye tracking data for specific classification goals. In the last part of our evaluation we apply the procedure to further public data sets without re-training the auoencoder nor the data manipulator. The results show that the learned manipulation is generalized and applicable to other data too.

Introduction

Due to the spread of the eye tracking technology over many fields and its use in everyday life, the specific information content in the eye tracking signal becomes more and more important (Bulling and Gellersen 2010; Majaranta and Bulling 2014). This is mainly due to the fact that the gaze signal is very rich in information and on the other hand that it cannot be turned off or easily controlled by a human (Hansen et al. 2003; Stellmach and Dachselt 2012). Many applications use this signal, however, still little value is placed on the anonymization of the signal. This is partly due to the fact that the topic of differential privacy has come into the focus of eye tracking research last year (Steil et al. 2019b; 2019a; Liu et al. 2019), but also to the challenge of finding specific patterns in the signal itself that make a person identifiable.

Initially in 2014 the problem of personal information in the eye tracking signal was mentioned for the first time as well as the person specific patterns contained in the signal (Liebling and Preibusch 2014). They mentioned critical attributes that are contained in the eye tracking data like age, gender, personal preference or health (Liebling and Preibusch 2014). This information poses a new challenge to modern eye tracking systems, which must now learn to hide this information. The basic approach of differential privacy is based on adding random noise to the signal: to cover up people specific data. However, this only works in the case of prefabricated features, since modern machine learning techniques such as convolutional neuronal networks are able to adapt their feature extractors. Furthermore, it would be more interesting to find specific patterns either in the stimulus itself or, as in this paper, in the scan path, which we can remove from the signal. On the one hand, this offers an insight into important characteristics which are interesting for science. On the other hand, it can be used in many other areas such as gaze guidance (Latif et al. 2014; Kano and Tomonaga 2011) or expertise evaluation (Gegenfurtner, Lehtinen, and Säljö 2011; Kunze et al. 2013).

In this paper, we present an approach that is able to learn an image manipulation to hide specific information while preserving other information. Our approach uses reinforcement learning on the sparse representation learned by an autoencoder. This combination allows to manipulate general patterns in an image, since the autoencoder has to reconstruct it based on a reduced set of values. This reduced set can be found in the central part of the autoencoder. It is also called bottleneck, and the following transposed convolutions of the autoencoder reconstruct the image on the basis of this reduced set. Meaning, that those values represent patterns in an image that are manipulated by an agent in our approach. This agent tries to hide specific information by manipulating those values. Another agent tries to train new classifiers to adapt to the manipulated data. This retraining allows our approach to diminish all personal patterns in the data since the classifiers adapt to the manipulated data too.

Contribution of this work:

1 A novel approach to remove patterns from eye tracking data that contain personal information which achieves a similar goal as differential privacy.

2 Independent of static features due to the iterative usage of CNNs.

3 Identification of general patterns in the data instead of adding random noise as it is done in differential privacy.

4 With our method it is possible to specify the information type which has do be hidden in the data.

5 Evaluation of the importance of the spatial and temporal information in eye tracking data.

6 Evaluation on unseen data.

Related Work

In this work, we deal with three topics. The first is differential privacy, in which people try to hide specific information. Also other information should remain in the data. To achieve this we use modern machine learning approaches to get a reduced representation of the input images, which ensures that general patterns are manipulated. For this we use deep autoencoders. For the classification, we also use deep nets in combination with the softmax loss function. The manipulation of the data itself is based on statistics and can be understood as Markov process (Smyth 1997). At the end of this paper we describe a possible application of our approach in the area of gaze guidance. For this reason, we have decided to split the related work into three parts, which are described below.

Differential privacy

This section contains the range of information contained in eye tracking data, the biometric properties of eye movements, and the general case of differential privacy. In the last part we move on to the modern approaches to differential privacy in eye tracking that appeared last year.

The rich information content available in human eye movements has been shown in several studies. One example is the pupil dilation of a subject. It holds information about the cognitive load (Matthews et al. 1991) as well as the attention to the scene or personal interest in the scene (Hess and Polt 1960). Mental disorders such as Alzheimer (Hutton, Nagel, and Loewenson 1984), Parkinson (Kuechenmeister et al. 1977), or schizophrenia (Holzman et al. 1974) can be detected in the eye movement behavior as well. Additionally, the eye movements hold information about the activity of the human (Bulling, Weichel, and Gellersen 2013; Steil and Bulling 2015), the cognitive state (Marshall 2007) and personal attributes [Hoppe et al. 2018]. While all of this information is already critical, several researchers have shown that the gender and age can be estimated from the eye movements as well (Cantoni et al. 2015; Sammaknejad et al. 2017). Of course, it is useful for diagnosis or security tasks to be able to extract this information, but this information should not be available to everybody just by receiving your eye tracking data or measuring your gaze behavior.

However, the high and unique information content in the eye tracking signal only becomes clear when the application for biometrics is considered. Here, it is possible to unambiguously identify the person by means of the eye behavior. The first approaches required a moving point stimulus which was followed by the user (Kasprowski 2004; Kasprowski and Ober 2004; 2005) or static images (Maeder and Fookes 2003). In 2010, the first approach that was able to distinguish users with a task independent approach was presented (Bednarik et al. 2005). At the same time, model based approaches that map the gaze behavior on a oculomotor model appeared too (Komogortsev et al. 2010; Komogortsev and Holland 2013). This approach was further developed to distinguish users even if they perform different tasks like browsing, writing, reading, or watching movies (Eberz et al. 2016). This was achived by computing twenty features on the gaze signal and measuring the difference of these features. For virtual reality head sets, a user authentication was proposed in (Zhang et al. 2018) using different stimuli and analyzing the gaze signal.

All of these publications show the potential threat to a human by revealing his gaze data. This means also that raw eye tracking data has to be handled with care for storage and for transmission. This topic falls into the field of differential privacy, which has a large theoretical foundation. Practical applications fall into the realm of localization (Pyrgelis, Troncoso, and De Cristofaro 2017), biomedical data (Saleheen et al. 2016), and continuous time series signal (Fan and Xiong 2012) as it is the case for eye tracking data. The main goal of differential privacy is to hide the private information while keeping the utility of the signal as high as possible. Here, utility is the measure of how good the original signal can be reconstructed. For the purpose of hiding information, random noise is added to the signal (Fredrikson et al. 2014) either to the raw data or in the frequency domain (Nikolov, Talwar, and Zhang 2013; Kasiviswanathan et al. 2013). Adding too much noise to the signal perserves privacy, but makes the signal itself useless (Fredrikson et al. 2014). In general, the utility and privacy tradeoff is tailored around a specific use case (Pyrgelis, Troncoso, and De Cristofaro 2017), which can be understood as a classification target in the eye tracking world. For further information, we refer to survey papers (Zhu et al. 2017; Ligett and Roth 2012).

For differential privacy related to eye tracking data is only covered by three papers so far. The first publication focuses on head mounted eye trackers (Steil et al. 2019b). It proposes a field camera that is able to avoid the recording of other persons. While this is not directly related to the research field of differential privacy it falls into the scope because it considers the problem from another perspective. The second paper tries to hide information in the eye tracking signal of the user itself (Steil et al. 2019a). They use the approach from (Dwork, Roth, and others 2014), which adds random noise to the signal. The third paper is about the private information included into heatmaps that are usually used for visualization (Liu et al. 2019). They found that those heatmaps still contain information about a subject and should therefore be used with caution.

Reinforcement learning

Reinforcement learning in the area of machine learning refers to one or more agents trying to learn a strategy that maximizes their reward (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013). The agent in this scenario has different actions that it can perform and after each action it receives a certain reward. For this, different cases have to be considered. The first case are temporal actions similar to a walk through a labyrinth where the agent receives his reward after it tried to go through the labyrinth (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013). This means that after executing several actions, the agent receives his final reward. In the second case, the agent has several possible actions without temporal dependency (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013). In the following, we only deal with the temporally independent application, because we also pursue this in this work. The scenario of several actions without temporal dependency can also be understood as a multi-armed bandit (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013). Here, the agent has the possibility to activate any number of levers after which it expects a reward. The strategy to be learned here is the optimal combination, whereby the consideration of all possible combinations exceeds the computing capacity of modern computer systems (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013). In the case of the multi-armed bandit, where each leaver has only two states, this would be 2^{Levers}. In order to learn this strategy and the optimal combination of levers, there is exploration on the one hand and exploitation on the other. In the exploration, the bandit is tested with new lever combinations regarding the reward, and the learned strategy is adapted. In case of exploitation, the learned strategy is used to get the maximum reward. If the exploration does not reveal possibilities for a greater reward, the process is saturated and the final strategy is learned (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013).

In order to learn complex strategies, there are basically two approaches; one is model based where a statistical model is given. This model is formulated as a Markov decision problem and is described by states and transitions that are known in advance. For the training of a model based approach, a multitude of action selection strategy algorithms have been proposed. The first approach is called the greedy algorithm and usually used together with an optimistic initialization (Kaelbling, Littman, and Moore 1996; Kober, Bagnell, and Peters 2013). The second approach in reinforcement learning is called model free. Here the algorithm learns a strategies on how to behave under different circumstances. Therefore, the model is not known in advance, but estimated through exploration. The most famous approach herefore is called the Q-learning algorithm (Luong et al. 2019). This algorithm learns policies for possibly an infinite amount of states, whereby each state can have a different amount of actions. It consists of a learning rate and a table that holds the information gathered so far. This table is updated with new observations and new actions are chosen using the same selection algorithms as described in the area of the model-based approaches. A disadvantage of the Q-learning algorithm is that it is only applicable if the state and action space is small. Therefore, the deep neuronal networks are employed to replace the table and output the best action by observing the current state. This is called the Deep Q-Learning algorithm (DQL) (Luong et al. 2019). In contrast to the tables the DQL approach has the disadvantage that the neuronal networks are nonlinear function approximators that only receive the reward for training. This means that the network may not be stable or even diverge (Mnih et al. 2015). To solve this issue, multiple approaches have been proposed and combined (Luong et al. 2019). The first is called the experience replay mechanism (Luong et al. 2019). For this approach, the algorithm initializes a replay memory. The initialization is done using the ϵ-greedy algorithm. Out of this memory, mini batches are selected and used for training (Luong et al. 2019). Afterwards, the neuronal network is used to make new experiences, which are stored in the memory. Therefore, the network can always learn on old and new experiences and is thus, stable to train (Luong et al. 2019). The second approach to stabilize the training of the neuronal network is called fixed target Q-network (Luong et al. 2019). For this approach, two neuronal networks are used. The first one is trained based on the memory and, afterwards, used to slowly update the second network after a fixed set of steps of the learning process (Luong et al. 2019). This is especially helpful if the initial exploration is not sufficient. Newer extensions for the DQL are Double Deep Q-Learning (Van Hasselt, Guez, and Silver 2016), Deep Q-Learning with Prioritized Experience Replay (Schaul et al. 2015), Dueling Deep Q-Learning (Wang et al. 2015), Asynchronous Multi-step Deep Q-Learning (Mnih et al. 2016), Distributional Deep Q-learning (Bellemare, Dabney, and Munos 2017), Deep Q-learning with Noisy Nets (Fortunato et al. 2017), and Rainbow Deep Q-learning (Hessel et al. 2018) together with extension in the area of combinatorical approaches of DQL and the Markov decision problem to be able to handle a infinite action space too. For a more detailed overview, we refer the reader to the survey paper (Luong et al. 2019).

Method

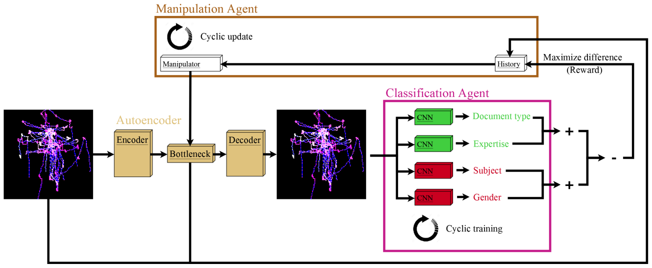

Figure 1: The workflow used for our approach. Classification Agent holds and uses the classifiers and Manipulation Agent the manipulator. Both agents are retrained after a fixed set of steps and have a buffer to hold old and new examples.

Figure 1 shows the general workflow of our approach. The autoencoder is trained preliminary to reconstruct the image. In its central part, it holds values that correspond to general patterns for the reconstruction of the image (Bottleneck in Figure 1). The idea behind using the autoencoder is that it reduces the input data ($64 * 64 * 3 = 12.228$ to $4 * 4 * 256 = 4096$) and thus also the possible action combinations of Manipulation Agent. Furthermore, it ensures that in the end, an image is still generated that is similar to the input image or consists of general patterns compared to a direct manipulation of the image by Manipulation Agent. Manipulation Agent is the reinforcement part of our approach. It learns a manipulation of the bottleneck from the autoencoder based on previous seen input images and the classification result from Classification Agent. This classification result is only the difference between the good (Green classifiers in Figure 1) and bad (Red classifiers in Figure 1) information revealed by the classifiers. This difference is used as reward in Manipulation Agent for the performed manipulation, whereas the image itself is the state. The different classification objectives (Document type, expertise, subject, gender) in Figure 1 are intended to indicate that our approach supports any number of classifiers. Manipulation Agent tries to worsen the accuracy of the red classifiers and to keep the accuracy of the green classifiers high. In contrast to this, Classification Agent tries to adapt the classifiers to the new image manipulation by retraining them. In the following each part is described in detail.

Table 1 first row shows the architecture of the used autoencoder. Each convolution block is followed by a rectifier linear unit (ReLu) and max pooling for size reduction. For the decoder of the autoencoder, we used transposed convolutions instead of pooling. The input to the network is an image with size $64 \times 64 \times 3$. The bottleneck in the autoencoder is the block with size $4 \times 4 \times 256$. For the training, we used stochastic gradient decent with an initial learning rate of 10^{-2}, decreasing each 200 epochs by a factor of 10^{-1}. The training stops at a learning rate of 10^{-7}. Weight decay was set to $5 * 10^{-4}$ and momentum to $9 * 10^{-1}$. During training, we used a batch size of 40 and the L2 loss formulation. This autoencoder is trained only once before starting our reinforcement learning approch.

The classifiers used in the Classification Agent (Table 1 second and third row) use a similar structure as the autoencoder. A detailed view of the classifiers can be seen in Table 1. Each convolution block uses a ReLu together with a max pooling operation. Before the first fully connected layer, we used a dropout, which deactivates 50% randomly. A and B in Table 1 have the same structure except for the last fully connected layer, which has either eight (Subject) or four (Stimulus image) output neurons. For the training, we used stochastic gradient decent with an initial learning rate of 10^{-4} decreasing each 500 epochs by a factor of 10^{-1}. The training stops at a learning rate of 10^{-7}. Weight decay was set to $5 * 10^{-4}$ and momentum to $9 * 10^{-1}$. During training, we used a batch size of 50 and the log multi class loss with softmax.

Since these classifiers are subject to the cyclic training of Classification Agent, they are always re-trained once the reinforcement learning has stabilized. This new training is done with a random initialization. The idea behind this is that the convolutions, which learn new feature extractors, adapt to the new image manipulation and thus improve the classification result. The training itself is done using the not manipulated and all the manipulated images seen so far (only from the training set).

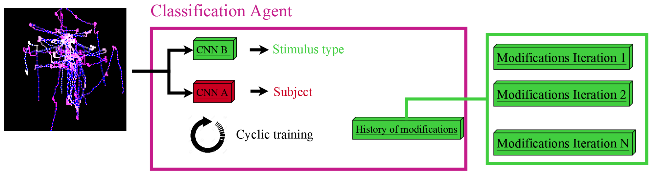

Figure 2: Used setup of Classification Agent with a memory for manipulated data seen in the past.

Figure 2 shows the workflow for Classification Agent with the memory. In comparison to Figure 1, which is a general overview, it can be seen that we now have only two classes. Those two classes are also used in our experiment for the evaluation section which is why we decided to insert them in the detailed view of the Classification Agent. In the memory (Figure 2) are all the seen manipulated images from the training set together with their labels. Images from the validation set are discarded and therefore, not stored in the memory of Classification Agent. For the training and test set, we made a 50% to 50% split. We seperated the data to produce equal amounts of stimulus and subject classes. As can be seen in this description, Classification Agent does not use reinforcement learning. This agent can be understood as a supervised learner, which retrains its classifiers.

In contrast to the Classification Agent, the Manipulation Agent uses reinforcement learning for training. The used DQL model can be seen in Table 1 fourth row. It consists of three convolution blocks and a fully connected output layer. The input of this model is the current image, which is called the state and the output of this model (4096 fully connected neurons) are the actions. Between each convolution block, we used ReLu and max pooling as in the models before. The output of the last layer was set 1 if it was greater or equal to 0.5, otherwise it was set to 0. Meaning, our model could either deactivate a feature in the bottleneck of the autoencoder or let it unchanged. For the training we used stochastic gradient decent with a fixed learning rate of 10^{-4}. The training stops after ten epochs of training on the entire memory of Manipulation Agent. Weight decay was set to $1 * 10^{-5}$ and momentum to $9 * 10^{-1}$. During training, we used a batch size of 100 and the L2 loss formulation for reinforcement learning $(predicted - actual)^2$. The parameter $predicted$ in this context means the result of DQL1 from the current input image. Since there is no ground truth in reinforcement learning, the parameter $actual$ is computed based on a second network (DQL2) and the reward R. Therefore, the ground truth is formulated as $actual = R + y * DQL2$. As mentioned before, R is the reward (Result of Classification Agent), $DQL2$ is the output of a second network and y is the discount factor, which is adjusted through training so that the net explores more in the beginning. This usage of two neuronal networks is called fixed target Q-network (Luong et al. 2019). Therefore, after ten training runs of DQL1, we set $DQL2 = DQL1$ since DQL1 has stabilized.

In addition to the fixed target network, we use the experience replay mechanism (Luong et al. 2019) as can be seen in Figure 3. As mentioned in the related work, this concept describes the memory which holds all examples (Stimulus, actions, and classification result). In this memory, we only store examples from the training set, since we want to eval-

Table 1: The configuration of all used models in our work. The autoencoder is used for extracting high level features from the input image. Classifier A is the network to classify the subject and Classifier B the network to classify the stimulus image. The DQL model is used in the Manipulation Agent as Deep Q-Learning algorithm (DQL).

Autoencoder	Classifier A	Classifier B	DQL
Input $64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$
CONV $32 \times 7 \times 7$	$32 \times 7 \times 7$	$32 \times 7 \times 7$	$32 \times 7 \times 7$
ReLu, Max Pooling	ReLu, Max Pooling	ReLu, Max Pooling	ReLu, Max Pooling
CONV $64 \times 7 \times 7$	$64 \times 7 \times 7$	$64 \times 7 \times 7$	$64 \times 7 \times 7$
ReLu, Max Pooling	ReLu, Max Pooling	ReLu, Max Pooling	ReLu, Max Pooling
CONV $128 \times 5 \times 5$	$128 \times 5 \times 5$	$128 \times 5 \times 5$	$128 \times 5 \times 5$
ReLu, Max Pooling	ReLu, Max Pooling	ReLu, Max Pooling	ReLu, Max Pooling
CONV $256 \times 5 \times 5$	$256 \times 5 \times 5$	$256 \times 5 \times 5$	Fully 4096
ReLu	ReLu, Max Pooling	ReLu, Max Pooling	-
TCONV $128 \times 5 \times 5$	Fully 512	Fully 512	-
ReLu	ReLu	ReLu	-
TCONV $64 \times 5 \times 5$	Fully #Classes	Fully #Classes	-
ReLu	-	-	-
TCONV $32 \times 7 \times 7$	-	-	-
ReLu	-	-	-
TCONV $3 \times 7 \times 7$	-	-	-

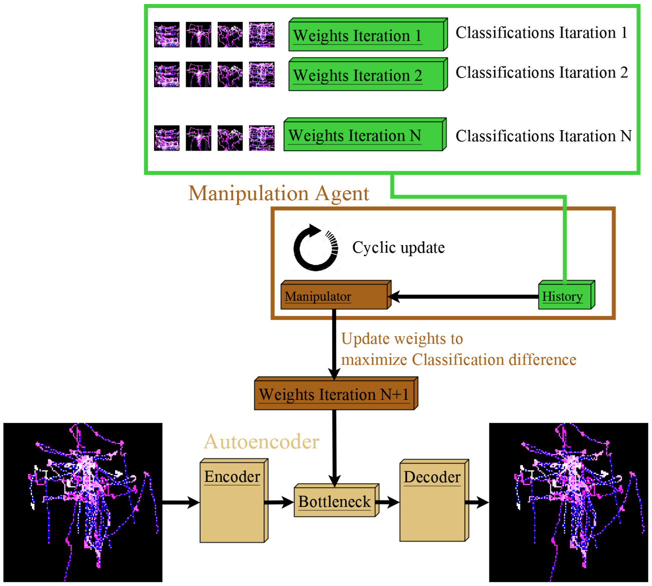

Figure 3: Memory and setup of Manipulation Agent.

uate our approach especially for unseen data. This memory is initialized before starting the entire approach and the networks DQL1 and DQL2 are trained on it. For this initialization, we compute the change of each value in the bottleneck on the classification and store it in the memory of Manipulation Agent. In addition, we compute one hundred random changes of 2-100 values in the bottleneck. This means that for the change of two values, we compute one hundred random changes and the same for three values, four values, and so on.

For data augmentation of all models, we used random noise which was in the range of 0-20%, cropping and shifting the scanpath. Cropping in this context means that we extracted randomly 60-100% of the scanpath and draw it on the input image. With shifting, we mean a randomly selected constant shift of the entire scanpath. This shift was selected in the range of 0-30% of the stimulus size.

In order to be able to show a comparison to the state often the art, we have evaluated an approach to differential privacy. Therefore, Laplacian-distributed noise based on the sensitivity (δf) of the signal is added to the raw eye tracking data (DP Raw) or to the generated image (DP Image) (Dwork 2008; 2011; Sarathy and Muralidhar 2011). As sensitivity (δf) for the raw eye tracking date we used the average Manhattan distance between all gaze point recordings (Rastogi and Nath 2010). Since those do not have equal length we cut off the last part of the larger recording. Therefore, the scale parameter of the Laplacian-distribution is $\frac{\delta f}{\epsilon}$ (Sarathy and Muralidhar 2011). For our evaluation we apllied the Laplacian Noise one hundred times to an image and computed the class using the networks. Afterwards, we selected the maximum vote as detected class. In case of the images as data we used the average maximum pixel distance (Manhatten distance between the red, green, and blue channel) between all image pairs (Fan 2019). The same formula was used to compute the scale parameter Laplacian-distribution. For a fair comparison we evaluated different values of ϵ in 0.01 steps and present the results of the best found ϵ based on the adapted classifiers. The search range of the optimal ϵ was $[0.01 - 15.0]$ for the images and for the noise injection into the raw gaze data the search range was $[10.00 - 500.0]$. For the images the ϵ value has to be multiplied by the image resolution (64×64) due to the sequential composition theorem and the independecy of the image pixels (McSherry 2009). Therefore, the search range for the ϵ was $[40.96 - 61440]$ but the 0.01 search steps are made based on the single pixel search range ($[0.01 - 15.0]$). It also has to be mentioned that we skipped ϵ values for the raw eye tracking data if there were less than three gaze points remaining on the image. The optimal epsilon was selected based on the maximal distance between the stimulus and subject classification, where the subject classification had to be at chance level. For the DOVES (Bovik et al. 2009) dataset we selected the best epsilon based on the minimal subject classification but the stimulus classification had to be over chance level. This was done for the DOVES (Bovik et al. 2009) data set, because all algorithms did not work as desired for this data set.

In addition, we have also evaluated a supervised trained approach to justify the use of reinforcement learning for the manipulation. The same models as shown in Table 1 were used and trained as a Generative Adversarial Network (GAN). The autoencoder is the generator and the classifiers A and B are used as discriminators. Before we trained the GAN we initially trained the Autoencoder, Classifier A, and Classifier B for one hundred epochs with the training parameters already provided. This was done to stabilize the training of the GAN afterwards. To adapt the initial training to the training of a GAN we added the logarithmic loss from the generated image as it was done in the original GAN paper (Goodfellow et al. 2014) with the difference, that the classifiers still had to predict the correct class.

For the generator (G) we used also the formulation of the original paper ($log(1 - D(G(I)))$) (Goodfellow et al. 2014) but in our case the discriminator (D) consists of two networks. Therefore, Classifier A and Classifier B can only contribute 0.5 each but in inverse directions. This means if Classifier A is correct it contributes 0.5 and if Classifier B is wrong it additionally contributes 0.5 since we want the GAN to learn to perserve the information classified by Classifier A and hide the information important for Classifier B. Based on the softmax output we can simply compute the probability for the correct class for Classifier A and Classifier B and weight both with 0.5.

Evaluation

For our first two experiments, we used the data provided with the ETRA 2019 challenge (Otero-Millan et al. 2008; McCamy et al. 2014). In this data, 8 subjects with 120 trials per subject are recorded. Therefore, it consists of 960 trials with a length of 45 seconds per trial. They recorded different tasks namely visual fixation, visual search, and visual exploration. Additionally, four different stimuli were presented; Which are blank, natural, where is waldo, and picture puzzle. For the image generation out of the raw gaze data files, we used the approach from (Fuhl et al. 2019). This means that the raw gaze data is in the red channel as dots, the green channel holds the time by adjusting the intensity of the dot, and the blue channel holds the relation ship of the gaze points by connecting them as lines. This can be seen in Figure 4.

The first experiment shows the results of our approach for different iterations, as well as before and after the adaption of the classifiers (Classification Agent). This experiment shows that our approach is capable of removing unwanted information in the scanpath. In this scenario, it is the information of the subject. In the second experiment, we evaluate the importance of different channels of the input image for different iterations of our approach. This experiment shows the advantage of our approach to other differential privacy methods since the feature extractors (Neuronal networks in Classification Agent) adapt to the new image manipulation as well as our image manipulation technique. For all experiments, we used a 50% split of the data where the test and validation set contain always equal amounts of subjects and stimuli samples.

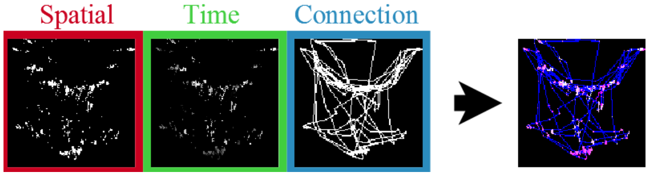

Figure 4: The encoding of eye tracking data as image. In the red channel all locations of gaze points are marked as dot. The time is encoded using the intesity in the green channel. Sequence of gaze points are embedded in the blue channel.

Table 2: Accuracy of the classifiers after each iteration and before as well as after the adaption of Classification Agent. RL is the proposed approach, GAN the same models trained supervised, DP Raw is the Differential Privacy applied to the raw gaze data, and DP Image is the differential Privacy applied to the image. The best results are in bold.

	No Adaption		Adaption	
Iteration	Stim	Sub	Stim	Sub
RL Initial	-	-	0.96	0.93
RL 1	0.95	0.13	0.96	0.93
RL 2	0.95	0.11	0.95	0.91
RL 5	0.91	0.12	0.93	0.52
RL 10	0.88	0.14	0.91	0.31
RL 15	0.78	0.12	0.86	0.22
RL 20	0.81	0.13	**0.83**	**0.15**
GAN	0.75	0.13	0.81	**0.15**
DP Raw $\epsilon = 223.21$	0.27	0.15	0.30	**0.15**
DP Image $\epsilon = 10485.76$	0.41	0.11	0.59	**0.15**
Chance level	0.25	0.12	0.25	0.12

Table 2 shows the classification results per iteration. With iteration we mean that the reinforcement learning (Manipulation Agent) has stabilized, which are approximately one thousand training runs. After each iteration, Classification Agent starts to retrain the classifiers, which is indicated by the adaption rows. The first line in Table 2 shows the initial results of the pretrained classifiers. At the bottom of Table 2, the chance level is shown. As can be seen Manipulation Agent always succeeds in dropping the classification accuracy for the subject close to the chance level. Afterwards, Classification Agent adapts the classifiers, but with less success for the subject classification if the process over all iterations is considered. In the last iteration (20), the training of the subject classifier fails and is close to the chance level.

This is also true for DP RAW (Differential privacy applied to the raw eye tracking data), DP Image (Differential privacy applied to the image), and the GAN (generative adversarial network) approach. When comparing the results of the different approaches, it is clear that our reinforcement learning approach is much better than differential privacy in terms of receiving the stimulus information. Compared to the GAN approach our approach gives slightly better results.

Table 3: Importance of spatial (R, G; red and blue channel) and temporal (B; intensity in green channel) features for the classification per iteration. The importance is measured in percent of values changed in total per channel.

Iteration	Red (Spat.)	Green (Temp.)	Blue (Spat.)
1	72%	16%	12%
2	19%	12%	69%
5	36%	33%	31%
10	41%	20%	39%
15	38%	22%	40%
20	37%	28%	35%

Table 3 shows the percentage amount of changed values per channel normalized over the total amount of changed values. Due to the construction of the image with raw dots in the red channel, connected dots in the blue channel, and the time as intensity value per dot in the green channel, we can estimate the importance of their contribution. For iteration one, it can be seen that the subject information was mainly extracted out of the red channel, which holds only spatial information. In the second iteration, this swaps to the blue channel, which holds the interconnections between the gaze points and therefore the spatial information. After five iterations, the amount of changes have balanced per channel. If we compare this result to Table 2, it can be seen that this already had an significant impact on the adaption of the subject classifier. After the last iteration, the amount of changes has again nearly balanced, where the green channel is the lowest. Since the green channel is the only channel that has temporal information, it could be argued that it is less important for the subject information since the green channel also contains spatial information. This statement is of course purely hypothetical and requires further experiments and research as well as another construction of input data.

In the next experiment we will use the data manipulation (learned with reinforcement learning) and the autoencoder on other public data sets without further training. The classifiers, however, will be re-trained on the output of the autoencoder and additionally adapted to the data manipulation in a further step. But before we come to the evaluation of this experiment, we describe the individual data sets used.

Gaze (Dorr et al. 2010): Is a huge data set with eye tracking data on dynamic scenes. The data was recorded using an SR Research EyeLink II eye tracker with 250 Hz. For our experiment we only used the data provided for static images where each static image of a video was considered the same image. In addition, we had to exclude subject V01 since there was only one recording for it available. Therefore, we used the eye tracking data of 10 subjects on 9 images for our experiment with an average recording length of 2 seconds. The training and test split was done using 50% for the training and 50% for the validation with a random selection. To treat both classifiers equally, the training set contained data from each subject as well as from each image.

WherePeopleLook (Judd et al. 2009) (WPL): Is a eye tracking data set and published under the intention to integrate top down features into saliency map generation. It consists of 1003 static images with eye tracking data of 15 subjects per image with an average recording length of 3 seconds. For our experiment we use a 50% to 50% split where in the training data all subjects and all images are available at least once to treat both classifiers equally.

DOVES (Bovik et al. 2009): Contains eye tracking data of 29 subjects on 101 natural images with an average recording length of 5 seconds. The recordings were performed with a 200 Hz high-precision dual-Purkinje eye tracker. Similar to the WherePeopleLook (Judd et al. 2009) data set we made a 50% to 50% training and testing split. In the training data each subject and each images was at least once to treat both classifiers equally.

For the classification on the data sets Gaze (Dorr et al. 2010), WherePeopleLook (Judd et al. 2009), and DOVES (Bovik et al. 2009) we used the same model as in Experiment 1 (Table 1). For training, we set the initial learning rate to 10^{-2} and reduced it by a factor of 10^{-1} every 100 epochs until we reached 10^{-7}. The optimizer used was stochstic gradient decent with weight decay of $5 * 10^{-4}$ and momentum of 0.9. For the data set Gaze (Dorr et al. 2010) we used a batch size of twice the number of classes and made sure that there were always 2 examples of each class in a batch. For WherePeopleLook (Judd et al. 2009) we also used double the number of classes for the subject classification. For the Stimulus Classification we used only the single class number as batch size. For the last data set DOVES (Bovik et al. 2009) we used twice the number of classes as batch size for both classifiers as for Gaze (Dorr et al. 2010) and also made sure that there were always two examples of each class per batch.

In the first column of Table 4 the results without data manipulation on the data sets Gaze, WherePeopleLook, and DOVES can be seen. Comparing these with the results on the Challenge data set in Table 2 it can be seen that the results are significantly lower. One reason for this is that there are many more classes which increases the challenge for the classification, but the main reason is the significantly lower recording time. For the Challenge data set the average recording time is 45 seconds. In comparison, Gaze has an average of 2 seconds, WherePeopleLook an average of 3 seconds, and DOVES an average of 5 seconds. This shows that the Challenge data set provides a multiple of the information for the neural networks. This means that the data from the Challenge data set contains significantly more personal information as well as more information about the structure of the stimuli. It is also interesting how little eye tracking data is sufficient to classify a subject. If for example the data set DOVES is compared with Gaze and WherePeopleLook (Table 4 first column) it can be seen that DOVES has a higher accuracy for the subject classification although DOVES has

Table 4: Accuracy on new unseen data sets with retrained classifiers but the same data manipulation learned from experiment 1 and 2 as well as the same weights for the autoencoder. RL is the proposed approach, GAN the same models trained supervised, DP Raw is the Differential Privacy applied to the raw gaze data, and DP Image is the differential Privacy applied to the image. The best results are in bold.

		None		Manipulation		Adapted	
Data set	Method	Stim	Sub	Stim	Sub	Stim	Sub
Gaze	RL	75%	31.66%	40%	8.88%	**71.11%**	**13.33%**
	GAN	75%	31.66%	37.24%	14.32%	61.64%	19.53%
	DP Raw $\epsilon = 31.28$	75%	31.66%	15.44%	12.54%	16.25%	13.96%
	DP Image $\epsilon = 34938.88$	75%	31.66%	21.22%	11.81%	59.83%	13.61%
	Chance	11.11%	10%	11.11%	10%	11.11%	10%
WPL	RL	31.23%	30.06%	21.54%	6.39%	**30.48**	**8.28%**
	GAN	31.23%	30.06%	18.47%	14.76%	26.74%	20.37%
	DP Raw $\epsilon = 68.20$	31.23%	30.06%	0.19%	7.09%	0.4%	8.93%
	DP Image $\epsilon = 16547.84$	31.23%	30.06%	7.15%	6.72%	8.41%	8.91%
	Chance	0.099%	6.66%	0.099%	6.66%	0.099%	6.66%
DOVES	RL	10.86%	44.90%	4.3%	6.69%	**9.15%**	13.66%
	GAN	10.86%	44.90%	5.68%	6.73%	8.26%	19.55%
	DP Raw $\epsilon = 425.39$	10.86%	44.90%	1.81%	3.95%	1.42%	**12.45%**
	DP Image $\epsilon = 13762.56$	10.86%	44.90%	1.14%	5.01%	1.5%	22.11%
	Chance	0.99%	3.44%	0.99%	3.44%	0.99%	3.44%

a lower chance level but a 2-3 seconds longer recording time. In contrast to this the detection rate for the stimuli classification is significantly lower compared to the other data sets.

The second column in Table 4 shows the results after the data manipulation by Manipulation Agent. Manipulation Agent has not been retrained and neither has the autoencoder. As you can see, this data manipulation has a significant impact on the accuracy of the classifiers. This is true for the stimulus as well as for the subjects, although the subject classification is more influenced (except for the DOVES data set, everything is reduced below the chance level). Since this can also be purely due to data augmentation, we have also adapted the classifiers to the data manipulation via training. For this purpose, the training examples were manipulated with Manipulation Agent and both the unaltered and the manipulated data were used for the training. The results can be senn in the third column of Table 4. While the subject classification in the DOVES data set is still significantly above the chance level (13.66%), the personal information was mainly removed in the other two data sets. What can also be seen is that the stimulus information was mainly retained for all data sets. This shows, at least empirically, that our approach has found generalized patterns to hide specific information.

Conclusion

In this work, we showed the applicability of reinforcement learning for removing personal information from eye tracking data. In addition, it can be used to evaluate the features and is able to adapt to an adaptive attacker (Classification Agent in Figure 1). Our approach is theoretically also capable of removing as well as preserving the information of multiple classification targets but this was not shown in our evaluation and is part of future research. In Table 4 we empirically showed that our approach has generalized and is also applicable to unseen data sets. This is especially interesting since it could mean that our approach can be applied to improve the robustness of neuronal networks as a preprocessing module or during training as an adversarial attack generator. Both possibilities are purely hypothetical and have to be tested and evaluated fist which is also part of future work.

References

Bednarik, R.; Kinnunen, T.; Mihaila, A.; and Fränti, P. 2005. Eye-movements as a biometric. In *Scandinavian conference on image analysis*, 780–789. Springer.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 449–458. JMLR. org.

Bovik, A.; Cormack, L.; Van Der Linde, I.; and Rajashekar, U. 2009. Doves: a database of visual eye movements. *Spatial vision* 22(2):161–177.

Bulling, A., and Gellersen, H. 2010. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing* 9(4):8–12.

Bulling, A.; Weichel, C.; and Gellersen, H. 2013. Eyecontext: recognition of high-level contextual cues from human visual behaviour. In *Proceedings of the sigchi conference on human factors in computing systems*, 305–308. ACM.

Cantoni, V.; Galdi, C.; Nappi, M.; Porta, M.; and Riccio, D. 2015. Gant: Gaze analysis technique for human identification. *Pattern Recognition* 48(4):1027–1038.

Dorr, M.; Martinetz, T.; Gegenfurtner, K. R.; and Barth, E. 2010. Variability of eye movements when viewing dynamic natural scenes. *Journal of vision* 10(10):28–28.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9(3–4):211–407.

Dwork, C. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, 1–19. Springer.

Dwork, C. 2011. A firm foundation for private data analysis. *Communications of the ACM* 54(1):86–95.

Eberz, S.; Rasmussen, K. B.; Lenders, V.; and Martinovic, I. 2016. Looks like eve: Exposing insider threats using eye movement biometrics. *ACM Transactions on Privacy and Security (TOPS)* 19(1):1.

Fan, L., and Xiong, L. 2012. Adaptively sharing time-series with differential privacy. *arXiv preprint arXiv:1202.3461*.

Fan, L. 2019. Differential privacy for image publication.

Fortunato, M.; Azar, M. G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; et al. 2017. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*.

Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.; Page, D.; and Ristenpart, T. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 17–32.

Fuhl, W.; Bozkir, E.; Hosp, B.; Castner, N.; Geisler, D.; Santini, T. C.; and Kasneci, E. 2019. Encodji: Encoding gaze data into emoji space for an amusing scanpath classification approach ;). In *Eye Tracking Research and Applications*.

Gegenfurtner, A.; Lehtinen, E.; and Säljö, R. 2011. Expertise differences in the comprehension of visualizations: A meta-analysis of eye-tracking research in professional domains. *Educational Psychology Review* 23(4):523–552.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hansen, J. P.; Johansen, A. S.; Hansen, D. W.; Itoh, K.; and Mashino, S. 2003. Command without a click: Dwell time typing by mouse and gaze selections. In *Proceedings of Human-Computer Interaction–INTERACT*, volume 3, 121–128.

Hess, E. H., and Polt, J. M. 1960. Pupil size as related to interest value of visual stimuli. *Science* 132(3423):349–350.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Holzman, P. S.; Proctor, L. R.; Levy, D. L.; Yasillo, N. J.; Meltzer, H. Y.; and Hurt, S. W. 1974. Eye-tracking dysfunctions in schizophrenic patients and their relatives. *Archives of general psychiatry* 31(2):143–151.

Hutton, J. T.; Nagel, J.; and Loewenson, R. B. 1984. Eye tracking dysfunction in alzheimer-type dementia. *Neurology* 34(1):99–99.

Judd, T.; Ehinger, K.; Durand, F.; and Torralba, A. 2009. Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)*.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4:237–285.

Kano, F., and Tomonaga, M. 2011. Perceptual mechanism underlying gaze guidance in chimpanzees and humans. *Animal cognition* 14(3):377–386.

Kasiviswanathan, S. P.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2013. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*, 457–476. Springer.

Kasprowski, P., and Ober, J. 2004. Eye movements in biometrics. In *International Workshop on Biometric Authentication*, 248–258. Springer.

Kasprowski, P., and Ober, J. 2005. Enhancing eye-movement-based biometric identification method by using voting classifiers. In *Biometric Technology for Human Identification II*, volume 5779, 314–323. International Society for Optics and Photonics.

Kasprowski, P. 2004. Human identification using eye movements. *Institute of Computer Science*.

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32(11):1238–1274.

Komogortsev, O. V., and Holland, C. D. 2013. Biometric authentication via complex oculomotor behavior. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 1–8. IEEE.

Komogortsev, O. V.; Jayarathna, S.; Aragon, C. R.; and Mahmoud, M. 2010. Biometric identification via an oculomotor plant mathematical model. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, 57–60. ACM.

Kuechenmeister, C. A.; Linton, P. H.; Mueller, T. V.; and White, H. B. 1977. Eye tracking in relation to age, sex, and illness. *Archives of General Psychiatry* 34(5):578–579.

Kunze, K.; Kawaichi, H.; Yoshimura, K.; and Kise, K. 2013. Towards inferring language expertise using eye tracking. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 217–222. ACM.

Latif, N.; Gehmacher, A.; Castelhano, M. S.; and Munhall, K. G. 2014. The art of gaze guidance. *Journal of experimental psychology: human perception and performance* 40(1):33.

Liebling, D. J., and Preibusch, S. 2014. Privacy considerations for a pervasive eye tracking world. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 1169–1177. ACM.

Ligett, K., and Roth, A. 2012. Take it or leave it: Running a survey when privacy comes at a cost. In *International Workshop on Internet and Network Economics*, 378–391. Springer.

Liu, A.; Xia, L.; Duchowski, A.; Bailey, R.; Holmqvist, K.; and Jain, E. 2019. Differential privacy for eye-tracking data. *arXiv preprint arXiv:1904.06809*.

Luong, N. C.; Hoang, D. T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.-C.; and Kim, D. I. 2019. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*.

Maeder, A. J., and Fookes, C. B. 2003. A visual attention approach to personal identification.

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

Marshall, S. P. 2007. Identifying cognitive state from eye metrics. *Aviation, space, and environmental medicine* 78(5):B165–B175.

Matthews, G.; Middleton, W.; Gilmartin, B.; and Bullimore, M. 1991. Pupillary diameter and cognitive load. *Journal of Psychophysiology*.

McCamy, M. B.; Otero-Millan, J.; Di Stasi, L. L.; Macknik, S. L.; and Martinez-Conde, S. 2014. Highly informative natural scene regions increase microsaccade production during visual scanning. *Journal of neuroscience* 34(8):2956–2966.

McSherry, F. D. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 19–30.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Nikolov, A.; Talwar, K.; and Zhang, L. 2013. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 351–360. ACM.

Otero-Millan, J.; Troncoso, X. G.; Macknik, S. L.; Serrano-Pedraza, I.; and Martinez-Conde, S. 2008. Saccades and microsaccades during visual fixation, exploration, and search: foundations for a common saccadic generator. *Journal of vision* 8(14):21–21.

Pyrgelis, A.; Troncoso, C.; and De Cristofaro, E. 2017. Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*.

Rastogi, V., and Nath, S. 2010. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 735–746.

Saleheen, N.; Chakraborty, S.; Ali, N.; Rahman, M. M.; Hossain, S. M.; Bari, R.; Buder, E.; Srivastava, M.; and Kumar, S. 2016. msieve: differential behavioral privacy in time series of mobile sensor data. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 706–717. ACM.

Sammaknejad, N.; Pouretemad, H.; Eslahchi, C.; Salahirad, A.; and Alinejad, A. 2017. Gender classification based on eye movements: A processing effect during passive face viewing. *Advances in cognitive psychology* 13(3):232.

Sarathy, R., and Muralidhar, K. 2011. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Trans. Data Priv.* 4(1):1–17.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Smyth, P. 1997. Belief networks, hidden markov models, and markov random fields: a unifying view. *Pattern recognition letters* 18(11-13):1261–1268.

Steil, J., and Bulling, A. 2015. Discovery of everyday human activities from long-term visual behaviour using topic models. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 75–85. ACM.

Steil, J.; Hagestedt, I.; Huang, M. X.; and Bulling, A. 2019a. Privacy-aware eye tracking using differential privacy. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, 27. ACM.

Steil, J.; Koelle, M.; Heuten, W.; Boll, S.; and Bulling, A. 2019b. Privaceye: privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, 26. ACM.

Stellmach, S., and Dachselt, R. 2012. Look & touch: gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2981–2990. ACM.

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*.

Wang, Z.; Schaul, T.; Hessel, M.; Van Hasselt, H.; Lanctot, M.; and De Freitas, N. 2015. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*.

Zhang, Y.; Hu, W.; Xu, W.; Chou, C. T.; and Hu, J. 2018. Continuous authentication using eye movement response of implicit visual stimuli. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1(4):177.

Zhu, T.; Li, G.; Zhou, W.; and Philip, S. Y. 2017. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering* 29(8):1619–1638.