Medical Image Analysis 54 (2019) 179–192

Contents lists available at ScienceDirect

Medical Image Analysis

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

Patient-attentive sequential strategy for perimetry-based visual field acquisition

Şerife Seda Kucur[a,*], Pablo Márquez-Neila[a], Mathias Abegg[b], Raphael Sznitman[a]

[a] ARTORG Center for Biomedical Engineering Research, University of Bern, Bern, Switzerland
[b] Department of Ophthalmology, Bern University Hospital, Inselspital, Bern, Switzerland

ARTICLE INFO

Article history:
Received 22 November 2018
Revised 8 March 2019
Accepted 14 March 2019
Available online 23 March 2019

Keywords:
Perimetry strategy
Visual field
Neural network
Reinforcement learning
Image reconstruction
Sequential experimental design

ABSTRACT

Perimetry is a non-invasive clinical psychometric examination used for diagnosing ophthalmic and neurological conditions. At its core, perimetry relies on a subject pressing a button whenever they see a visual stimulus within their field of view. This sequential process then yields a 2D visual field image that is critical for clinical use. Perimetry is painfully slow however, with examinations lasting 7–8 minutes per eye. Maintaining high levels of concentration during that time is exhausting for the patient and negatively affects the acquired visual field. We introduce **PASS**, a novel perimetry testing strategy, based on reinforcement learning, that requires fewer locations in order to effectively estimate 2D visual fields. **PASS** uses a selection policy that determines what locations should be tested in order to reconstruct the complete visual field as accurately as possible, and then separately reconstructs the visual field from sparse observations. Furthermore, **PASS** is patient-specific and non-greedy. It adaptively selects what locations to query based on the patient's answers to previous queries, and the locations are jointly selected to maximize the quality of the final reconstruction. In our experiments, we show that **PASS** outperforms state-of-the-art methods, leading to more accurate reconstructions while reducing between 30% and 70% the duration of the patient examination.

1. Introduction

Estimating a subject's capacity to sense light is critical for diagnosing numerous ocular and neurological conditions. In the case of glaucoma, an eye condition that affects over 60 million people worldwide, the need to quantify how well patients perceive light is paramount to monitor the disease over years (Racette et al., 2016). Given the ageing world population and the growing number of glaucoma patients, the need for reliable methods is a significant public health concern.

To measure light perception, *perimetry* is the standard-of-care (Racette et al., 2016; Heijl et al., 2012). This non-invasive functional eye examination automatically quantifies a subject's sensitivity to light across the field of view. In most cases, the central 30° of the visual field is evaluated by sequentially projecting brief light stimuli of different brightness to an observer who fixates a central reference point. When a subject perceives a stimulus, they press a button to confirm the observation and a new stimulus is presented. By presenting stimuli at different regions of the visual periphery, perimetry yields a 2D image, or *visual field* (Fig. 1), quantifying the ability to perceive light at each location. Visual fields are then the basis for diagnosis and treatment (Racette et al., 2016).

While safe and inexpensive, perimetry suffers from a number of limitations for both patients and clinicians (Racette et al., 2016). In particular, the examination requires the patient to concentrate for up to 7–8 minutes per eye (King-Smith et al., 1994; Weber and Klimaschka, 1995; Bengtsson et al., 1998). For patients typically 60 to 90 years old, this is exhausting, unpleasant and leads to high drop-out rates in scheduled bi-annual monitoring appointments. More importantly, the quality and reliability of the visual field depends on maintained high attention of the patients throughout the exam. Long examinations time cause an increase in false positive and false negative responses, thus significantly reducing the quality of the examination. This in turn makes treatment planning by the clinician more challenging (Gonzalez-Hernandez et al., 2005; Wall et al., 2004). As such, there is an important need to speed up examinations in order to improve the overall quality of care.

Yet at the heart of perimetry lies an inherent accuracy *vs.* speed trade-off. That is, accurate visual fields could be produced if every location of the visual field were tested multiple times with different intensities. Doing so would however require unbearably long examinations. Conversely, testing just a few locations with no

* Corresponding author.
E-mail address: serife.kucur@artorg.unibe.ch (Ş.S. Kucur).

https://doi.org/10.1016/j.media.2019.03.002
1361-8415/

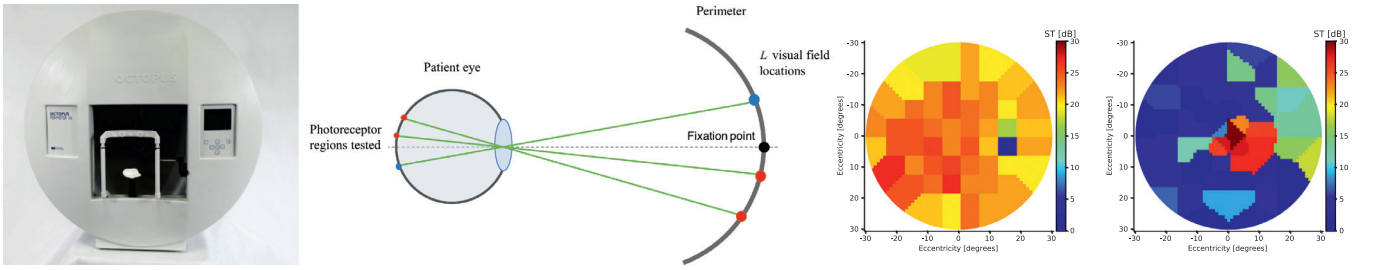


Fig. 1. (a) An OCTOPUS 900 perimeter (Haag-Streit AG, Switzerland), (b) A 1D schematic representation of the perimetry principle (generalizes to 2D). The patient eye (left) focuses on a fixation point in the perimeter (black point). Light of different intensities are sequentially shown to $L = 3$ locations in the perimeter (right) that stimulate photoreceptor regions in the patient's retina. A light sensitivity threshold at each evaluated photoreceptive region in the retina is recorded by the machine. In this example, two low thresholds (in red) and one high threshold (blue) are illustrated. (c) A 2D visual field map from a healthy subject acquired using a 24-2 pattern with $L = 54$ tested locations. The warm to cool color scale corresponds to high to low sensitivity thresholds. (d) A G-pattern visual field with $L = 59$ tested locations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

intensity variability would be extremely fast but yield clinically unusable visual fields.

Recent methods have been proposed to tackle this trade-off more efficiently. Chong et al. (2014, 2016) leverage the spatial relations of individual locations to improve the overall accuracy of visual fields. These methods rely on the fact that anatomical structures and pathological factors are indicators of the co-dependency between values at different locations (Weber et al., 1990; Anton et al., 1998). While potentially more precise, these methods remain slow, as each location still needs to be evaluated. As was later shown in (Wild et al., 2017; Kucur and Sznitman, 2017), exhaustive testing is unnecessary as the co-dependencies between locations can be exploited to reduce the number of evaluated locations. In particular, unobserved locations can be estimated using Markov Random Fields (Wild et al., 2017) or sparse reconstruction methods (Kucur and Sznitman, 2017). However, in terms of what locations to select, both methods are greedy in that they only pick what location should be next, either once for all eyes (Kucur and Sznitman, 2017) or dynamically during the examination (Wild et al., 2017). This yields locations that do not truly take advantage of the visual field co-dependencies and yields large inaccuracies at locations that have not been evaluated. Ultimately this strongly limits their usability in practice.

We present a *Patient Attentive Sequential perimetry Strategy* (**PASS**). **PASS** has three properties to overcome the above shortcomings: it is (i) *sparse*, examining only a limited number of the visual field locations; it is (ii) *patient-adaptive*, selecting the sequence of locations to examine in an online manner based on the previous answers from the patient; and it is (iii) *non-greedy*, picking the locations that jointly yield the most accurate visual field within the given duration of the exam. To do this, we separate the problem into (1) a visual field estimation problem and (2) a location selection problem. (1) enables the *sparsity* property, while (2) leads to the *patient-adaptiveness* and *non-greediness* properties. For (1) we propose to use a Neural Network (NN) or Least Squares approach to reconstruct the visual field from partial observations that are retrieved sequentially, while for (2) we use a separate NN that is trained to predict the best locations to pick given the history of observations. Our method then iterates between reconstructing visual fields from partial observations to selecting the next location to observe. As such, our work is a Reinforcement Learning instance (Sutton and Barto, 1998) and is related to attention models (Ranzato, 2014). Our main contributions are threefold:

- The presented framework is the first to show a policy-gradient reinforcement learning approach for the task of visual field reconstruction from sparse observations. At each time point of our sequential method, we determine what location to evaluate, using previous locations and the current best estimate of the entire visual field.
- Given a predefined number of locations to be evaluated, we show how to learn what locations need to be selected to approximate the global optimum. This is in stark contrast to state-of-the-art methods that exclusively rely on one-step look ahead criteria to select the following location to test.
- We compare the use of two different sparse reconstruction methods for the task of visual field estimation. The first relies on a sparse linear model, while the second involves a NN-based approach.

We show in our experimental section that **PASS** provides superior performances compared to state-of-the-art methods on two different datasets acquired with different perimeters. Ultimately, we show that our method provides better results with shorter examination times.

The remainder of this paper is organized as follows: In Section 2, we briefly describe perimetry and existing related works. In Section 3, we outline our proposed framework and thoroughly evaluate it in Section 4. We conclude with final remarks in Section 5.

2. Related work

2.1. Principles of perimetry: A brief overview

Perimetry quantifies the ability of retinal photoreceptors to perceive light (Racette et al., 2016). The basic principle is to present short (i.e. 200 ms) light stimuli of different intensities (in decibels) so that the brightness with which perception is detected 50% of the time can be estimated. This brightness value is called the *sensitivity threshold* and is a noisy measurement. Healthy and deteriorated retinal locations typically have low and high sensitivity thresholds, respectively.

To collect sensitivity threshold estimates over multiple retinal locations, the subject fixates a central point on a screen while the stimuli are presented. By varying where the stimuli appear in the field of view, sensitivity thresholds at different retinal locations can then be estimated (see Fig. 1b). For a given eye, this ultimately yields a 2D grid of sensitivity threshold values known as a visual field (VF) (see Fig. 1c-1d). Modern devices automatically evaluate 50–60 locations per eye using a variety of different coarse grid patterns. For more information on perimetry, we refer the reader to (Racette et al., 2016).

Given this setting, we can treat VFs as images whose pixels are sensitivity thresholds (i.e. with values ranging between 0–40 dB) as shown in Fig. 1c and d. Since perimeters can stimulate

photoreceptive regions in any order, with any given intensity, an important characteristic of the VF acquisition process is that each location can be observed independently from each other. In addition, the acquisition time is linear in the number of locations evaluated. This fact has led researchers to develop different testing strategies that choose what locations to stimulate, with what intensities and in what order, so to estimate VFs as quickly and accurately as possible. This problem is the focus of the present work and we describe the most relevant existing methods in what follows.

2.2. Perimetry testing strategies

Early research on speeding up perimetry focused on exploring methods that reduced the number of stimuli used for a single photoreceptor region at a time. These included methods that used fixed and varying intensity testing intervals (Racette et al., 2016; Weber and Klimaschka, 1995), as well as Bayesian testing strategies (King-Smith et al., 1994; Bengtsson et al., 1998; Tyrrell and Owens, 1988; Anderson and Johnson, 2006). In some cases, 50%-80% speedup gains are achieved when compared to brute-force testing that lasted over 15 minutes per eye. In this work, we use the clinically validated varying intensity testing method proposed in Weber and Klimaschka (1995) to determine the sensitivity thresholds at a specific location.

To further improve the acquisition speed, other methods used VF values of neighboring locations to reduce the number of stimuli presentations. The Dynamic Strategy (DS) (Weber and Klimaschka, 1995) and the Tendency Oriented Perimetry (TOP) strategy (Morales et al., 2000), use neighboring locations to seed initial stimuli values at yet to be tested locations. While the former fully estimated sensitivity thresholds before testing neighboring locations, the latter stimulated each location only once, yielding very fast VFs with low-accuracy (De Tarso Ponte Pierre-Filho et al., 2006).

More recent methods have looked to characterize VF locations using Markov Random Fields (MRF) and Bayesian inference to estimate VF values. Denniss et al. (2013) and Ganeshrao et al. (2015) both proposed similar schemes that used graph priors derived from anatomical retinal structures. While showing improvements in testing efficiency, the overall gains are limited and the need to explicitly model the graph structure is complex in itself. GOANNA (Chong et al., 2014) improved this by dynamically determining which locations to test. SWeLZ (Rubinstein et al., 2016) and SEP (Wild et al., 2017) also used MRFs to incrementally estimate VFs and dynamically select what locations to evaluate based on sparse estimates. While both methods brought speed improvements, their performances on datasets with wide ranges of subjects (i.e. datasets with healthy and pathological subjects) are inferior due to difficulties in estimating model parameters.

Sparse sensing (Donoho, 2006) techniques have been heavily used in image reconstruction problems, especially in medical applications such as Medical Resonance Imaging (Lai et al., 2016; Ravishankar and Bresler, 2011; Haldar et al., 2011; Huang et al., 2011; Chen et al., 2018). By inspring from the same image reconstruction idea, sparse sensing was used as the basis of the Sequentially Optimized Reconstruction Strategy (SORS) (Kucur and Sznitman, 2017). Here the problem was formulated as a sparse reconstruction problem where incremental basis matrices were used to estimate the VF after a location was observed. While being almost parameter-free and computationally simple, the order in which locations are tested is fixed for all patients no matter the state of the evaluated VF. We show in our experiments that this fixed testing policy is suboptimal and results in poor reconstructions at non-evaluated VF locations.

2.3. Reinforcement learning and attention models

Our approach is related to Reinforcement learning (RL) (Sutton and Barto, 1998) and Attention Models (Ranzato, 2014; Mnih et al., 2014; Xu et al., 2015). In RL, an agent is tasked to learn how to maximize a numerical reward by sequentially interacting with an *environment*. Recent progress in RL has been achieved in a variety applications such as strategies for playing Atari (Mnih et al., 2015) or Go (Silver et al., 2017). Various computer vision methods for object localization (Caicedo and Lazebnik, 2015), object detection (Mathe et al., 2016), classification (Wiering et al., 2011) have also used RL approaches as well. In medical image computing, the works of Sahba et al. (2008), Chitsaz and Seng Woo (2011), and Wang et al. (2013) for image segmentation, of Ghesu et al. (2016) for localization or of Neumann et al. (2015, 2016) for multi-physics computational model personalization stand out.

In the context of sequential decision problems, Attention Models (Ranzato, 2014; Mnih et al., 2014; Xu et al., 2015) have gained much interest. The underlying idea behind such models is to learn where to sequentially focus computational resources in an image so to gather as much information towards a specific task (e.g. image classification). This implies ignoring irrelevant parts of an image while concentrating on its most important parts.The Recurrent Attention Model (RAM) (Mnih et al., 2014) is perhaps the most relevant model to ours. Here, the authors propose a recurrent neural network (RNN) to process the history of states and actions to decide which local region in an image needs to be 'attended' in order to correctly classify the image content.

Similar to the RAM model, we introduce a framework for sequential decision making in the context of perimetry testing: our proposed method sequentially attends a number of VF locations, using previously visited locations to reconstruct the VF. As in Mnih et al. (2014), we train our method to minimize the final loss which in our case is the VF reconstruction loss.

3. Patient-attentive sequential strategy

We formulate **PASS** as a sequential experimental design problem. Our method looks to provide the best possible VF reconstruction from noisy observations acquired from a limited number of locations. Our key insight is that by selecting the best locations for each patient, better reconstructions can be attained in shorter examination times. In what follows, we begin by specifying the **PASS** model. We then describe how to train this model using data and then specify the implementation choices we have made.

3.1. Model

To acquire a VF from a given eye, our method proceeds iteratively for T steps. At each time step $t < T$, a measured VF location, ℓ_t, is selected from the set $\Omega = \{1, \dots, L\}$ of possible locations. We denote the patient examined as a function $\mathcal{Q} : \Omega \to \mathbb{R}$ that when queried at a location $\ell \in \Omega$, returns a noisy sensitivity threshold $q_\ell \in \mathbb{R}$ from that location. Note that the function $\mathcal{Q}$ is assumed to be non-deterministic as a patient may respond differently even when queried multiple times at the same location. At time step t, we denote the history of observations received so far as $\mathbf{h}_t \in (\mathbb{R} \cup \{\bullet\})^L$. Specifically, the ℓ-th element of $\mathbf{h}_t$ contains q_ℓ if location ℓ was queried, or the masked value $\bullet$ if the location has not yet been observed. Our method performs one measurement at a time, hence $\mathbf{h}_t$ contains $L - t$ masked values at any time step t.

After each observation, we estimate the complete VF, denoted $\hat{\mathbf{y}}_t \in \mathbb{R}^L$, using a reconstruction function $f : \mathbb{R}^L \to \mathbb{R}^L$ that receives as input the history of observations $\mathbf{h}_t$. To select the next location to evaluate, we use a policy function π that also uses the history of observations $\mathbf{h}_t$, together with the reconstructed VF $\hat{\mathbf{y}}_t$, to

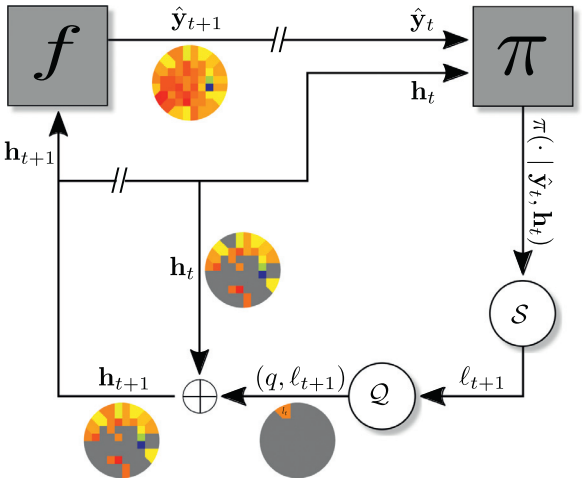


Fig. 2. Overview of the proposed method. At time step t, the policy π takes the reconstruction $\hat{\mathbf{y}}_t$, as well as the history of observations $\mathbf{h}_t$ and produces a probability distribution over the locations. $\mathcal{S}$ samples one location ℓ_{t+1} from this distribution, which is queried to the patient $\mathcal{Q}$, and yields an observation q. The history $\mathbf{h}_t$ is updated with the observation q and fed to the reconstruction function f to produce a new reconstruction $\hat{\mathbf{y}}_{t+1}$. The // symbol indicates the switch from one iteration to the next one. Algorithm 1 formalizes this procedure.

decide which location ℓ_{t+1} to query at the next time step $t+1$. This scheme is repeated T times, and we refer to this as the examination *horizon*. Fig. 2 depicts our model and Algorithm 1 formalizes this procedure. Here we use the notation $\mathbf{h} \oplus_\ell q$ to describe a vector equal to $\mathbf{h}$ but with the ℓ-th element set to q.

Algorithm 1 Patient Attentive Sequential Strategy (PASS).

Input: Patient $\mathcal{Q}$, policy function π, reconstruction function f
Output: Reconstruction of the VF $\hat{\mathbf{y}}_T$

$\mathbf{h}_0 \leftarrow (\bullet, \ldots, \bullet)$
$\hat{\mathbf{y}}_0 \leftarrow f(\mathbf{h}_0)$
for t **in** $(0, \ldots, T-1)$ **do**
  Sample next location $\ell_{t+1} \sim \pi(\cdot \mid (\mathbf{h}_t, \hat{\mathbf{y}}_t))$
  Query location from patient $q_{\ell_{t+1}} \leftarrow \mathcal{Q}(\ell_{t+1})$
  Update history $\mathbf{h}_{t+1} \leftarrow \mathbf{h}_t \oplus_{\ell_{t+1}} q_{\ell_{t+1}}$
  Reconstruct VF $\hat{\mathbf{y}}_{t+1} \leftarrow f(\mathbf{h}_{t+1})$
end for
return $\hat{\mathbf{y}}_T$

The key component of our method is the policy function π, that selects locations to query next. In particular, our policy function depends on the tuple $s_t = (\mathbf{h}_t, \hat{\mathbf{y}}_t)$ which we refer to as the current *state* of our model. If we let Σ be the set of all possible states, a policy function is formally defined as $\pi\colon \Omega \times \Sigma \to [0, 1]$, that takes the state s_t and builds a probability distribution over Ω. Thus, $\pi(\ell|s_t)$ is the probability of selecting the location ℓ when the state is s_t. A non-deterministic sampling procedure, denoted as $\mathcal{S}$ in Fig. 2, then chooses the location ℓ_{t+1} from this probability distribution.

In the following subsection, we describe how to learn the policy function from training data.

3.2. Training

Determining a good policy function is not trivial, as we wish to select locations that provide the best VF reconstruction after T iterations. Given that each selection depends on the history of previous selection, the main difficulty lies in attributing the value of individually picked locations when only observing the final VF reconstruction.

To this end, we will use a training phase to minimize a loss function over a training dataset $\mathcal{D} = \{(\mathbf{q}^{(i)}, \mathbf{y}^{(i)}) | i = 1, \ldots, N\}$ consisting of many pairs of noisy sensitivty thresholds $\mathbf{q}^{(i)} \in \mathbb{R}^L$ and the corresponding true VF values $\mathbf{y}^{(i)} \in \mathbb{R}^L$. Note that the noisy observations $\mathbf{q}$ serve to simulate the answers from the patient at training time. To evaluate the loss function for a pair $(\mathbf{q}, \mathbf{y})$, we compare the ground-truth $\mathbf{y}$ to the final reconstruction $\hat{\mathbf{y}}_T$ generated by our strategy in Algorithm 1. We can also quantify the difference between $\hat{\mathbf{y}}_T$ and $\mathbf{y}$ using the *mean squared error* (MSE),

$$P = \|\hat{\mathbf{y}}_T - \mathbf{y}\|_2^2, \tag{1}$$

that we refer to as the reconstruction *penalty*. Note that computing $\hat{\mathbf{y}}_T$ involves sampling locations from the policy function, thus both the reconstruction $\hat{\mathbf{y}}_T$ and the penalty P are in fact random variables that depend on the stochastic policy π. We thus define our loss as the expected penalty,

$$\mathcal{L}(\mathbf{q}, \mathbf{y}) = \mathbb{E}_{\boldsymbol{\ell} \sim \pi}[P], \tag{2}$$

where the expectation is taken over entire sequences of T locations. In contrast, Kucur and Sznitman (2017) defines their MSE penalty with respect to $\hat{\mathbf{y}}_{t+1}$, so that their expectation is only taken over individual locations. While computationally simple, such greedy or one-step lookahead strategies are rarely globally optimal with respect to $\hat{\mathbf{y}}_T$[1] For this reason, the goal of our training phase is to find a policy function π^* such that

$$\pi^* = \arg\min_\pi \mathbb{E}_{(\mathbf{q},\mathbf{y}) \sim \mathcal{D}}[\mathcal{L}(\mathbf{q}, \mathbf{y})], \tag{3}$$

where the loss is averaged over the training dataset.

To solve Eq. (3), we resort to a function approximation (Sutton et al., 2000). In particular, we model the policy function π_θ as a neural network with parameters θ and use a standard gradient-based method to minimize the loss. This requires computing the gradient of the expected penalty w.r.t. the parameters θ. The expected penalty however can not be computed in practice, as it would involve running over all possible sequences of T tested locations. Instead, we resort to Monte-Carlo sampling, whereby samples can be obtained from Algorithm 1 to compute approximations to the expected penalty. Unfortunately, doing so removes all differentiable structure w.r.t. θ and it is not possible to compute the gradient of the expected penalty from the Monte-Carlo samples. This is a well-known problem in reinforcement learning and can be solved using the REINFORCE rule (Williams, 1992). In our case, the gradient of the loss can be rewritten as

$$\begin{aligned}
\frac{\partial}{\partial\theta}\mathcal{L} &= \frac{\partial}{\partial\theta}\mathbb{E}_{\boldsymbol{\ell}\sim\pi_\theta}[P] = \frac{\partial}{\partial\theta}\sum_{\ell_1,\ldots,\ell_T} P\prod_{t=0}^{T-1}\pi_\theta(\ell_{t+1} \mid s_t) \\
&= \sum_{\ell_1,\ldots,\ell_T} P\frac{\partial}{\partial\theta}\prod_{t=0}^{T-1}\pi_{\theta t} = \sum_{\ell_1,\ldots,\ell_T} P\prod_{t=0}^{T-1}\pi_{\theta t}\cdot\frac{\partial}{\partial\theta}\log\prod_{t=0}^{T-1}\pi_{\theta t} \\
&= \mathbb{E}_{\boldsymbol{\ell}\sim\pi_\theta}\left[P\frac{\partial}{\partial\theta}\sum_{t=0}^{T-1}\log\pi_\theta(\ell_{t+1} \mid s_t)\right],
\end{aligned} \tag{4}$$

where we have used the fact that $\frac{\partial}{\partial\theta}p_\theta = p_\theta \cdot \frac{\partial}{\partial\theta}\log p_\theta$ for any function p_θ, and $\pi_{\theta t}$ is shorthand for $\pi_\theta(\ell_{t+1} \mid s_t)$. Eq. (4) provides a way to obtain the gradient of the expected penalty, which cannot be computed directly, as the expectation of a gradient. This can however be approximated with Monte-Carlo sampling,

$$\mathbb{E}_{\boldsymbol{\ell}\sim\pi_\theta}\left[P\frac{\partial}{\partial\theta}\sum_{t=0}^{T-1}\log\pi_{\theta t}\right] \approx \frac{1}{M}\sum_{m=1}^{M} P_{\tilde{\boldsymbol{\ell}}^{(m)}}\frac{\partial}{\partial\theta}\sum_{t=0}^{T-1}\log\pi_\theta(\tilde{\ell}_{t+1}^{(m)} \mid s_t), \tag{5}$$

[1] The greedy policy would need to satisfy Bellman's Optimality Principle (Bellman, 1957).

where M is the number of samples and $\tilde{\boldsymbol{\ell}}^{(m)}$ is a sequence of T locations sampled from π_θ. Note that this is an instance of a *policy gradient method* (Sutton et al., 2000) and that in practice we set the number of Monte-Carlo samples $M = 1$.

In addition, to reduce the variance of the REINFORCE estimator, it is a common to subtract a baseline b value from the penalty P (Williams, 1988):

$$\frac{\partial}{\partial\theta}\mathcal{L} \approx \frac{1}{M}\sum_{m=1}^{M}\left[\left(P_{\tilde{\boldsymbol{\ell}}^{(m)}} - b\right)\frac{\partial}{\partial\theta}\sum_{t=0}^{T-1}\log\pi_\theta(\tilde{\ell}_{t+1}^{(m)} \mid s_t)\right]. \tag{6}$$

Here, the optimal choice for b is the expected penalty (Williams, 1988), which is unknown and must be estimated. We do this by using a running average over all previously obtained penalties. We also include the entropy of the policy function, $H[\pi_\theta(\cdot\,|s_t)]$, as part of the gradient computation,

$$\frac{\partial}{\partial\theta}\mathcal{L} \approx \frac{1}{M}\sum_{m=1}^{M}\left[\left(P_{\tilde{\boldsymbol{\ell}}^{(m)}} - b\right)\frac{\partial}{\partial\theta}\sum_{t=0}^{T-1}\log\pi_\theta(\tilde{\ell}_{t+1}^{(m)} \mid s_t) - \lambda\frac{\partial}{\partial\theta}\sum_{t=0}^{T-1}H[\pi_\theta(\cdot \mid s_t)]\right], \tag{7}$$

where $\lambda > 0$ is the weight of the entropy term. This extra term has been used before (Xu et al., 2015) as a manner of increasing the entropy of the distributions generated by the policy. This avoids premature convergence and encourage exploration of the space of locations.

Algorithm 2 shows our implementation of Eq. (7) whereby the grad(a, b) operator returns the gradient of a w.r.t. b. This can be easily implemented with an automatic differentiation library (e.g. PyTorch (Paszke et al., 2017)). We use the special notation $\leftarrowtail$ to indicate a *detached* assignment that disables gradient computation by blocking backpropagation through it. During training, we iteratively call Algorithm 2 feeding random samples from $\mathcal{D}$ to compute approximations to the gradient of the loss, and use these gradients to update the parameters θ of the policy function. In the following two subsections, we specify the details of both the policy and reconstruction functions.

Algorithm 2 Computation of loss gradient with Monte-Carlo sampling.

Input: Sample $(\mathbf{q}, \mathbf{y}) \in \mathcal{D}$, policy function π_θ, reconstruction function f, baseline b
Output: Monte-Carlo approximation to gradient $\frac{\partial}{\partial\theta}\mathcal{L}$, updated baseline b

$\mathbf{h}_0 \leftarrowtail (\bullet, \ldots, \bullet)$, $\hat{\mathbf{y}}_0 \leftarrowtail f(\mathbf{h}_0)$
logp $\leftarrow 0$, entr $\leftarrow 0$
for t **in** $(0, \ldots, T-1)$ **do**
  Sample next location $\ell_{t+1} \sim \pi_\theta(\cdot \mid (\mathbf{h}_t, \hat{\mathbf{y}}_t))$
  logp $\leftarrow$ logp $+ \log\pi_\theta(\ell_{t+1} \mid (\mathbf{h}_t, \hat{\mathbf{y}}_t))$
  entr $\leftarrow$ entr $+ H\big[\pi_\theta(\cdot \mid (\mathbf{h}_t, \hat{\mathbf{y}}_t))\big]$
  Update history $\mathbf{h}_{t+1} \leftarrowtail \mathbf{h}_t \oplus_{\ell_{t+1}} q_{\ell_{t+1}}$
  Reconstruct VF $\hat{\mathbf{y}}_{t+1} \leftarrowtail f(\mathbf{h}_{t+1})$
end for
Compute penalty $P \leftarrowtail \|\hat{\mathbf{y}}_T - \mathbf{y}\|_2^2$
gradloss $\leftarrowtail (P - b)\cdot\text{grad}(\text{logp}, \theta) - \lambda\cdot\text{grad}(\text{entr}, \theta)$
Update baseline $b \leftarrowtail 0.99b + 0.01P$
return (gradloss, b)

3.3. Policy function

As mentioned above, we use a parameterized policy function π_θ modeled as an artificial neural network. This network takes the state $s_t = (\mathbf{h}_t, \hat{\mathbf{y}}_t)$ as input. The reconstruction $\hat{\mathbf{y}}_t$ and the history of observations $\mathbf{h}_t$ are processed separately in two independent streams and subsequently concatenated and processed by two fully connected layers. The output of the network is a L-dimensional vector normalized with a softmax operation to provide a probability distribution over the VF locations, which in practice models the policy function $\pi_\theta(\cdot\,|s_t)$. Table 1 summarizes the architecture of this network.

Table 1
Architecture for the policy network and the reconstruction network.

Policy network		Reconstruction network
History stream Input: $\mathbf{h}_t$	Reconstruction stream Input: $\hat{\mathbf{y}}_t$	Input: $\mathbf{h}_{t+1}$
linear ($2L \times 512$)	Linear ($L \times 512$)	Linear ($2L \times 256$)
ReLU	ReLU	ReLU
Linear (512×512)	Linear (512×512)	Linear (256×256)
ReLU	ReLU	ReLU
Concatenation		Linear (256×256)
Linear (1024×256)		ReLU
ReLU		Linear ($256 \times L$)
Linear ($256 \times L$)		
Softmax		

To speed up the learning process, we impose that the same locations can not be queried twice. To do this, the feature vector $\mathbf{w}$ of the last layer is modified using the mask vector and then transformed with softmax

$$\pi_\theta(\cdot \mid s_t) = \text{softmax}((1 - \mathbf{m}_t)\cdot\mathbf{w} - \mathbf{m}_t\cdot E), \tag{8}$$

where $E \gg 0$ is an arbitrarily large scalar and $\mathbf{m}_t = \mathbb{1}[\mathbf{h}_t \neq \bullet]$ is the mask of queried locations. This truncates the probability of location ℓ to zero if $(\mathbf{m}_t)_\ell = 1$ (*i.e.*, if it has been already queried).

3.4. Reconstruction function

From the history of observations $\mathbf{h}_t$, the reconstruction function $f : (\mathbb{R}\cup\{\bullet\})^L \to \mathbb{R}^L$ provides a reconstruction $\hat{\mathbf{y}}_t = f(\mathbf{h}_t)$ of the VF with the available information at time step t. In our experiments we propose and compare two different approaches for this reconstruction function. The first approach, based on SORS (Kucur and Sznitman, 2017), assumes a linear relationship between observations and the reconstructions. The second approach models f as a deep neural network.

3.4.1. Least squares (`LSTSQ`)

In this scheme, we assume that there is a linear mapping from the observations to the full VF reconstruction. Formally, if we let $\mathbf{M}_{\mathbf{h}_t}$ be the binary $t \times L$ matrix that removes the unobserved elements $\bullet$ from $\mathbf{h}_t$, we write the linear VF reconstruction as

$$\hat{\mathbf{y}}_t = \mathbf{B}_t\mathbf{M}_{\mathbf{h}_t}\mathbf{h}_t, \tag{9}$$

where $\mathbf{B}_t$ is a $L \times t$ matrix and we assume that $0\cdot\bullet = 0$. For each given history $\mathbf{h}_t$ we compute a suitable $\mathbf{B}_t$ for the specific set of queried locations of $\mathbf{h}_t$ by using a reconstruction training dataset $\mathcal{D}_{\text{rec}}$ and solving the quadratic problem

$$\mathbf{B}_t = \arg\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{B}\mathbf{M}_{\mathbf{h}_t}\mathbf{Q}\|_F^2, \tag{10}$$

where $\mathbf{Q}$ and $\mathbf{Y}$ are $L \times N_{\text{rec}}$ matrices containing the N_{rec} measurements $\{\mathbf{q}^{(i)}\}_{i=1}^{N_{\text{rec}}}$ and the real VFs $\{\mathbf{y}^{(i)}\}_{i=1}^{N_{\text{rec}}}$ from $\mathcal{D}_{\text{rec}}$. Eq. (10) has a closed form solution using least squares. Hence, the linear reconstruction function $f(\mathbf{h}_t)$ first performs a least squares to find the matrix $\mathbf{B}_t$ as the solution to Eq. (10) and then uses $\mathbf{B}_t$ for reconstruction as described in Eq. (9).

3.4.2. Neural network (`RNet`)

As more powerful alternative to the linear assumption, we model the reconstruction function as a neural network $f_\phi(\mathbf{h}_t)$ with parameters ϕ, which we will call the *reconstruction network*.

Table 2
Partitioning of the **Rotterdam** and **Bern** datasets into training, validation and test sets. Note that the training sets are further halved to train the policy network and the reconstruction function.

	# Tr. samples (Location Net. π)	# Tr. samples (Rec. function f)	# Val. samples	# Test samples
Rotterdam	2052	2077	470	509
Bern	4720	4780	1000	1180

Table 1 describes its architecture. We train this network by solving the optimization problem

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{(\mathbf{q},\mathbf{y})\sim\mathcal{D}_{\text{rec}},\mathbf{m}}\left[\|\mathbf{y} - f_\phi(\mathbf{h}(\mathbf{q},\mathbf{m}))\|_2^2\right]. \tag{11}$$

To compute this loss during training, the elements $\mathbf{q}$ and $\mathbf{y}$ are uniformly sampled from $\mathcal{D}_{\text{rec}}$. The masks $\mathbf{m}$ are randomly generated by simulating possible combinations of observed locations. Specifically, we generate each mask $\mathbf{m}$ by first sampling the number of observed locations from a uniform distribution $\mathcal{U}(1,T)$ and then sampling that number of elements from Ω. Sampled elements are set to 1 in the mask $\mathbf{m}$, and the rest are set to 0. The history $\mathbf{h}$ can be trivially constructed based on the observations $\mathbf{q}$ and the mask $\mathbf{m}$ as

$$h_\ell(\mathbf{q},\mathbf{m}) = \begin{cases} q_\ell & \text{if } m_\ell = 1, \\ \bullet & \text{if } m_\ell = 0. \end{cases} \tag{12}$$

Note that the reconstruction network and the policy network are trained independently in two separate phases. In a first phase we train the reconstruction network alone, which is then frozen during the subsequent training of the policy network. Also, both the linear and the neural network reconstruction functions are trained with a reconstruction training data set $\mathcal{D}_{\text{rec}}$ that differs from the data set $\mathcal{D}$ used to train the policy function.

4. Experimental results

In this section, we validate our approach **PASS** by evaluating it on two different datasets and by comparing its performance with a number of state-of-the-art methods.

4.1. Experimental set-up

We evaluated our approach on two separate datasets:

- **Rotterdam** dataset acquired at the Rotterdam Eye Institute (Netherlands) (Bryan et al., 2013; Erler et al., 2014). It includes 5108 visual field samples from 22 healthy and 139 glaucomatous patients. VFs were acquired using a 24-2 pattern (see Fig. 1c) with $L = 54$ locations by the Humphrey Visual Field Analyzer II (Carl Zeiss Meditec AG, Jena, Germany).
- **Bern** dataset containing 1108 visual fields from 538 patients collected at Inselspital Eye Clinic of Bern (Switzerland). VFs were collected with the G pattern (see Fig. 1d) with $L = 59$ locations using the OCTOPUS 900 Perimeter (Haag-Streit AG, Koeniz, Switzerland). We applied data augmentation to account for the low number of samples by using the Open Perimetry Interface (OPI) (Turpin et al., 2012) to simulate patient VFs. Using OPI, any perimetry strategy can be run on a patient model given a true VF measurement to generate additional instances of the same VF with natural variations. In our case, the VFs were simulated 10 times using the SimHenson model (Turpin et al., 2012; Henson et al., 2000), leading to 11080 samples in total.

Both datasets are partitioned in 80%, 10%, 10% splits corresponding to the training, validation and test sets, respectively. As previously mentioned, we train the policy function network and the reconstruction function on separate training sets. To do this, we split the training set into two halves. All splits are made in a patient-basis manner, so that VFs from the same patient are never present in two different splits. Table 2 summarizes the number of samples in the training, validation and test sets for both datasets.

We performed qualitative and quantitative comparison between our approach and the state-of-the-art methods **DS** (Weber and Klimaschka, 1995), **SORS** (Kucur and Sznitman, 2017), and **TOP** (Morales et al., 2000). We trained our **PASS** algorithm with both the **LSTSQ** and the **RNet** versions of the reconstruction function, which we denote **PASS+LSTSQ** and **PASS+RNet**, respectively. For fair comparison, the reconstruction network is pre-trained once for each dataset, and all the experiments with **PASS+RNet** use the same reconstruction network. At test time, the starting query stimulus at a next location is set to the value given by the previous reconstruction value, i.e. $q_{l_{t+1}} = (\hat{\mathbf{y}}_t)_{l_{t+1}}$. We then use the SimHenson model to simulate patient responses.

The horizon T is a hyperparameter of our policy. However, a policy trained for a specific horizon T might perform well when used with a different horizon at test time T_{test}. To assess this relation, we report performances of our method trained with horizons of $T = 8$, $T = 16$ and $T = 36$, and tested with horizons of $T_{\text{test}} = 8$, $T_{\text{test}} = 16$ and $T_{\text{test}} = 36$. The **DS** method has a fixed $T = T_{\text{test}} = L$, as it needs to query all possible locations. Similarly, **TOP** has a fixed horizon. For **DS** and **TOP**, we report results for their respective fixed horizons at training and testing time. **SORS** is trained for $T = L$, but its greedy nature allows us to stop the testing at any point T_{test}. Hence, we also use $T_{\text{test}} = 8$, $T_{\text{test}} = 16$ and $T_{\text{test}} = 36$ for **SORS**.

The experiments were implemented in Python and R. We used PyTorch (Paszke et al., 2017) as our automatic differentiation library. We perform mini-batch stochastic gradient optimization using the Adam (Kingma and Ba, 2014) optimizer to train the policy and reconstruction networks. The learning rate and batch size for the policy network were set to 10^{-5} and 256, respectively. For the reconstruction network, we used a batch size of 32 and a learning rate of 10^{-4}, which decayed by 0.1 every 300 epochs. The model that led to the best performance on the validation set was selected for evaluation on the test set.

4.2. Evaluation criteria and metrics

We use two different metrics to quantify the performance of different methods. The quality of the reconstructions is measured in terms of the MSE between the reconstruction $\hat{\mathbf{y}}$ and the ground-truth $\mathbf{y}$,

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2, \tag{13}$$

averaged over the testing dataset. We report the MSE computed both over all L locations and only in the $L - T_{\text{test}}$ unobserved locations, to assess the generalization power of our reconstruction methods. In particular, reconstruction errors on locations that have not been evaluated are of strong interest here as they indicate to what extent the clinician can believe the values presented.

We also provide the averaged *number of stimuli presentations* (#Pres.) required to acquire the VF by each method. #Pres. is proportional to the time taken to evaluate a patient. Note that, while related, #Pres. is not equal to the number of queries T_{test}, as each

individual location requires a number of stimuli presentations depending on its sensitivity threshold.

Since our overarching goal is to improve the quality of the reconstructions for a reduced amount of queries, we emphasize the importance of results obtained with small values of T_{test}, especially when $T_{\text{test}} = 8$. For larger horizons, such as $T_{\text{test}} = 36$, the quality of reconstructions is less interesting as the number of unobserved locations is smaller and there is a natural convergence in performance for all methods.

4.3. Quantitative performance

Table 3 compares the different methods on both the **Rotterdam** and **Bern** datasets. We observe that our approach brings improvements in terms of accuracy when compared to its counterparts, especially for unobserved locations. Moreover, our methods took similar or less examination time. For the **Rotterdam** dataset, PASS methods have remarkably lower MSEs than the baselines. When compared to TOP, PASS led to a huge increase in terms of accuracy and speed. For shorter horizons, PASS methods reached much better accuracy than TOP while requiring half the number of stimuli presentations. DS, having a fix horizon of $T_{\text{test}} = L$, provides better reconstructions than PASS for short test horizons, but it requires 125 more stimuli presentations than PASS on average. Interestingly, for longer horizons ($T_{\text{test}} = 36$), PASS outperforms DS, yielding higher quality reconstructions than DS in 30% less time. As for standard deviations (SD) of the MSEs, PASS methods, especially PASS+LSTSQ, had lower or similar SDs than TOP and DS for $T_{\text{test}} = 36$. For shorter horizons, they led to larger SD, which is expected since PASS leaves some locations untested and leads to higher variance in the acquisition process.

As expected, the gain in terms of accuracy decreases when the testing horizon grows. PASS+RNet could not perform better than SORS for $T_{\text{test}} = 36$, even though it had an excellent performance for shorter horizons. We attribute this lower performance for large horizons to limitations in the reconstruction network. Given that the number of possible combinations of queried locations increases exponentially with the horizon, the amount of training data required to properly train the network grows accordingly. This explains the drop in performance of reconstruction networks for larger testing horizons, and why LSTSQ outperforms RNet for $T = 36$. The fact that PASS+LSTSQ performs better than SORS while having the same reconstruction scheme shows how our patient-specific approach is more effective at selecting locations to query regardless of the reconstruction procedure. In terms of MSE SDs, PASS generally has lower or similar variance compared to SORS except at $T_{\text{test}} = 36$ where PASS+RNet led to higher MSE SD.

For the **Bern** dataset, PASS methods shows roughly the same performance as SORS in terms of mean MSE, even though slightly better accuracies are reached for $T_{\text{test}} = 8$ and $T_{\text{test}} = 36$ and the MSE SDs are higher than that of SORS. The **Bern** dataset includes relatively healthier patients — more than 70% — that can be easily estimated by any strategy. Still, our PASS methods reached better accuracy than DS and TOP while requiring between 36% and 70% fewer stimuli. Overall, PASS led to better accuracy on untested locations with similar or even less number of stimuli than SORS on both datasets.

We also compared our approach to another patient-specific scheme proposed in the literature, the *Spatial Entropy Pursuit* (SEP) (Wild et al., 2017) strategy, which selects VF locations according to uncertainty of the VF estimation in a greedy way and stops when the uncertainty is below a pre-defined threshold. For a fair comparison, we used the same training and testing datasets as in Wild et al. (2017) and compare the results with the SEP performance reported in Wild et al. (2017). We compare only PASS+RNet

Table 3
Performance of different perimetry strategies on **Rotterdam** and **Bern** datasets. Mean and standard deviations (SD) of the MSEs and of the number of presentations (#Pres) are provided for different test horizons $T_{\text{test}} = \{8, 16, 36\}$ for each strategy trained for horizons T. The MSEs are separately shown for tested and untested locations. The bold font shows the best performance among all PASS methods in terms of average MSEs and number of presentations.

	Method	MSE (SD) ($T_{\text{test}} = 8$)		#Pres.	MSE (SD) ($T_{\text{test}} = 16$)		#Pres.	MSE (SD) ($T_{\text{test}} = 36$)		#Pres.
		All	Untested		All	Untested		All	Untested	
Rotterdam	PASS+RNet ($T = 8$)	**27.84** (23.89)	**29.30** (25.54)	26.98 (4.21)	22.55 (18.17)	24.40 (20.44)	50.84 (7.17)	18.10 (14.63)	21.48 (19.98)	109.07 (16.54)
	PASS+LSTSQ ($T = 8$)	29.59 (24.22)	31.84 (26.68)	**25.80** (4.00)	21.27 (15.99)	24.15 (19.17)	**50.28** (7.26)	14.66 (9.75)	20.02 (15.33)	108.70 (15.29)
	PASS+RNet ($T = 16$)	28.42 (22.05)	30.12 (23.66)	27.99 (4.88)	**20.99** (17.04)	**22.38** (19.41)	51.93 (8.03)	16.45 (12.85)	16.08 (14.48)	111.06 (17.61)
	PASS+LSTSQ ($T = 16$)	33.02 (38.05)	35.58 (41.95)	27.41 (5.23)	21.67 (16.96)	24.14 (20.06)	51.69 (7.70)	14.13 (9.62)	16.59 (13.73)	109.50 (14.67)
	PASS+RNet ($T = 36$)	34.06 (27.33)	36.22 (30.08)	26.85 (4.08)	23.13 (18.52)	24.81 (21.47)	52.02 (7.25)	15.86 (12.46)	**14.51** (14.53)	111.52 (16.56)
	PASS+LSTSQ ($T = 36$)	40.47 (46.48)	44.40 (53.19)	27.08 (5.13)	22.74 (18.81)	25.54 (23.53)	50.87 (7.69)	**13.81** (9.57)	15.03 (14.87)	**108.07** (14.76)
	SORS ($T = 54$)	30.38 (25.22)	32.99 (27.72)	26.29 (4.60)	22.39 (17.02)	26.11 (20.56)	50.74 (7.69)	14.97 (9.90)	21.28 (16.08)	109.86 (16.13)
	TOP ($T = T_{\text{test}} = 54$)	32.35 (12.99)	-	54.00 (0.00)	32.35 (12.00)	-	54.00 (0.00)	32.35 (12.99)	-	54.00 (0.00)
	DS ($T = T_{\text{test}} = 54$)	15.17 (11.04)	-	156.31 (12.81)	15.17 (11.04)	-	156.31 (12.81)	15.17 (11.04)	-	156.31 (12.81)
Bern	PASS+RNet ($T = 8$)	**16.14** (17.28)	**16.77** (18.23)	23.66 (3.43)	13.39 (13.74)	14.44 (15.30)	46.65 (5.60)	11.19 (11.04)	12.89 (14.29)	104.24 (11.72)
	PASS+LSTSQ ($T = 8$)	17.02 (17.29)	17.96 (18.70)	23.43 (3.40)	13.17 (12.18)	14.53 (14.13)	46.61 (5.56)	9.84 (7.79)	11.80 (10.61)	104.38 (11.38)
	PASS+RNet ($T = 16$)	16.90 (19.17)	17.60 (20.79)	**22.63** (3.05)	13.27 (13.43)	**14.03** (15.02)	46.10 (5.41)	10.72 (10.54)	11.83 (13.25)	104.01 (11.16)
	PASS+LSTSQ ($T = 16$)	17.36 (17.81)	18.23 (19.29)	22.97 (3.01)	**13.07** (12.07)	14.08 (14.10)	46.33 (5.08)	9.73 (7.72)	11.35 (11.41)	103.82 (10.67)
	PASS+RNet ($T = 36$)	17.94 (22.30)	18.63 (24.04)	23.47 (3.54)	14.64 (15.23)	15.52 (17.03)	46.72 (5.73)	10.88 (10.86)	11.68 (13.56)	105.13 (11.74)
	PASS+LSTSQ ($T = 36$)	18.61 (22.42)	19.82 (25.05)	22.69 (2.98)	13.82 (13.23)	15.31 (15.80)	**45.46** (45.46)	**9.67** (7.81)	**11.13** (11.10)	**103.77** (10.83)
	SORS ($T = 59$)	16.67 (15.99)	17.67 (17.44)	23.50 (3.34)	12.98 (11.07)	14.48 (12.96)	47.48 (6.25)	9.82 (7.66)	12.28 (11.58)	104.89 (12.65)
	TOP ($T = T_{\text{test}} = 59$)	15.64 (8.67)	-	69.00 (0.00)	15.64 (8.67)	-	69.00 (0.00)	15.64 (8.67)	-	69.00 (0.00)
	DS ($T = T_{\text{test}} = 59$)	12.23 (7.60)	-	160.37 (13.20)	12.23 (7.60)	-	160.37 (13.20)	12.23 (7.60)	-	160.37 (13.20)

Table 4
Performances of **PASS** and **SEP** for $T_{\text{test}} = 20$.

Method	MSE (Mean, Median)	#Pres. (Mean, Median)
PASS+RNet ($T = 20$)	13.81, 5.86	68.34, 69
SEP	12.85, 10.72	88.87, 73

trained for $T = 20$. Table 4 shows the MSE means and medians, as well as the number of presentations achieved by each method. **PASS** performs better than **SEP** in terms of median MSE with fewer presentations. This implies that our scheme suggests better locations to query and estimates VFs more accurately. **PASS** is also more flexible than **SEP** when working with different VF patterns: **SEP** requires an explicit description of the VF layout and neighborhood, while **PASS** learns to cope with this information implicitly.

4.4. Error distributions

In this section, we present the error distributions of **PASS** and compare them to that of **SORS** in order to better analyze the average quantitative performance discussed in Section 4.3.

Fig. 3 presents the error distributions for the **Rotterdam** dataset, as well as their mean and standard deviations (SD). Compared to **SORS**, both **PASS**+RNet and **PASS**+LSTSQ methods for $T = 8$ have lower mean and SD. For longer horizons, **PASS**+RNET performed the worst potentially due to the confined training of

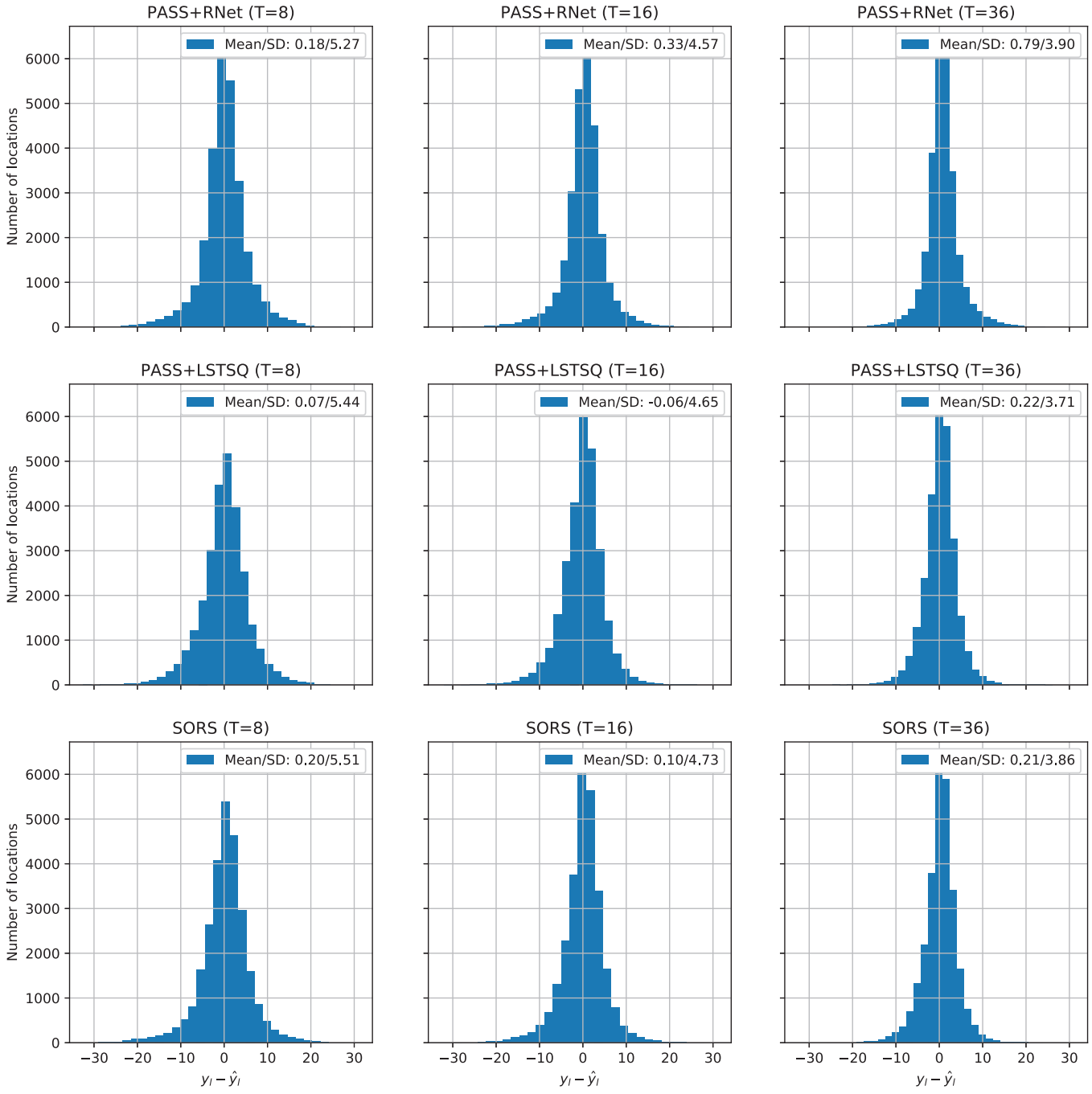


Fig. 3. Error distributions of **PASS** and **SORS** method for the **Rotterdam** dataset. Mean and standard deviations (SD) are given for each plot, and $T = T_{\text{test}}$ for each model.

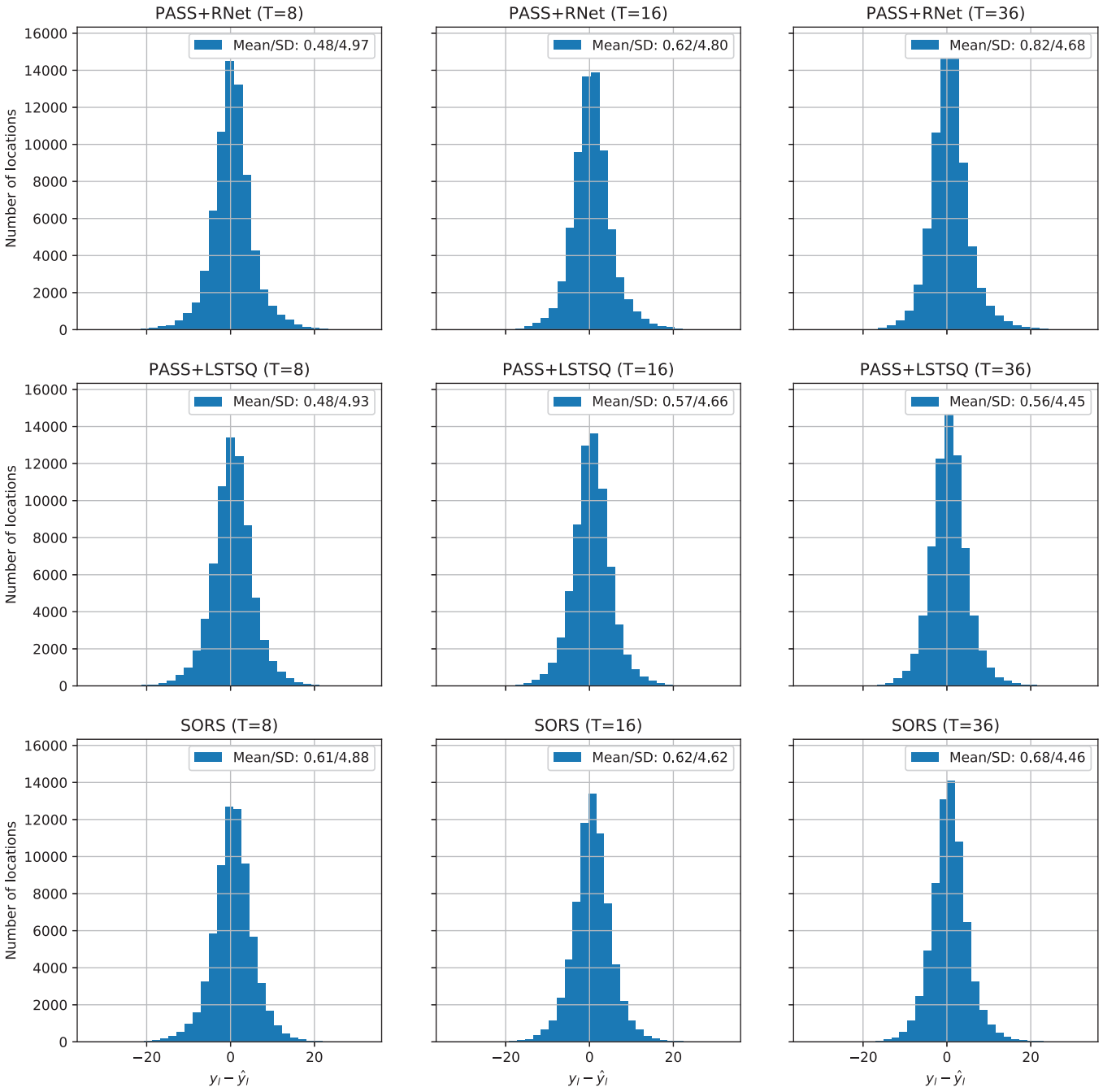


Fig. 4. Error distributions of PASS and SORS method for the **Bern** dataset. Mean and standard deviations (SD) are given for each plot, and $T = T_{\text{test}}$ for each model.

the reconstruction network for long horizons. This is supported by the fact that PASS+LSTSQ almost performed the best for every horizon whereby better performance could be achieved with PASS+LSTSQ compared to SORS even though they share the same reconstruction scheme. Similar observations can be made for the error distributions of the **Bern** dataset in Fig. 4, where the error variances for PASS are similar or slightly higher to those of SORS.

4.5. Performance on sub-populations

To better understand how PASS performs on VFs as a function VF health level, we disantagle the overall performance given in Table 3 and show the performances on sub-populations grouped by VF *mean defect* (MD)[2] We use Hodapp classification scheme Hodapp et al. (1993) to group VFs into three classes: VFs with MD ≥ -6 belong to the *early defect group* (EG); VFs with $-12 \leq$ MD < -6 belong to the *moderate defect group* (MG); and VFs with MD < -12 belong to *advanced defect group* (AG). Table 5 presents experimental results on thsse three groups for the **Rotterdam** and **Bern** datasets and compares the performance of PASS and SORS.

In most cases of the **Rotterdam** dataset, PASS methods outperformed SORS in terms of accuracy and number of presentations

[2] MD is the mean of the deviations of the sensitivity thresholds from the age-matched healthy normal values, and is a clinically well-accepted indicator of the severity of VF defects. A lower MD implies more severe VF defects.

Table 5
Performances on early defect group (EG), moderate defect group (MG) and advanced defect group (AG) sub-populations for both the **Rotterdam** and **Bern** dataset. Mean MSEs for each method and sub-population are given, with mean number of presentations in parenthesis. Percentages of each group in the training set is also shown. Bold font is used whenever any PASS method performance (in terms of error or number of presentations) is higher than SORS for the corresponding T. $T_{test} = T$ in all cases.

Method	Roterdam: Sub-populations			Bern: Sub-populations		
	EG (57.4%)	MG (19.4%)	AG (23.2%)	EG (70.6%)	MG (17.0%)	AG (12.5%)
`PASS+RNet` ($T = 8$)	**15.58** (**27.87**)	**45.57** (27.56)	**47.66** (24.88)	**9.14** (**24.25**)	30.31 (**21.62**)	**41.63** (22.47)
`PASS+LSTSQ` ($T = 8$)	**16.29** (**26.44**)	**46.28** (28.56)	**51.97** (23.45)	**10.28** (**23.85**)	26.82 (22.73)	45.62 (21.71)
`SORS` ($T = 8$)	16.66 (27.96)	48.37 (26.27)	53.20 (22.74)	10.54 (24.34)	24.50 (22.01)	43.86 (20.17)
`PASS+RNet` ($T = 16$)	**11.51** (55.45)	38.71 (54.71)	**34.88** (**43.43**)	**7.85** (**47.53**)	23.10 (**43.35**)	34.22 (40.80)
`PASS+LSTSQ` ($T = 16$)	12.99 (**52.83**)	**32.18** (58.15)	**36.41** (46.94)	**8.40** (**47.41**)	20.33 (43.95)	**32.46** (42.62)
`SORS` ($T = 16$)	12.78 (53.74)	32.60 (53.27)	39.21 (43.44)	8.46 (49.42)	19.10 (43.91)	32.68 (40.02)
`PASS+RNet` ($T = 36$)	**8.82** (119.61)	27.32 (**115.81**)	26.77 (92.73)	**6.30** (**108.74**)	18.23 (98.78)	29.58 (90.97)
`PASS+LSTSQ` ($T = 36$)	**8.56** (**115.42**)	**19.57** (**110.79**)	**22.95** (**91.43**)	**6.56** (**107.21**)	**14.92** (98.22)	22.12 (89.78)
`SORS` ($T = 36$)	9.44 (117.21)	21.467 (117.37)	24.43 (91.53)	6.78 (109.41)	15.17 (96.95)	21.73 (87.14)

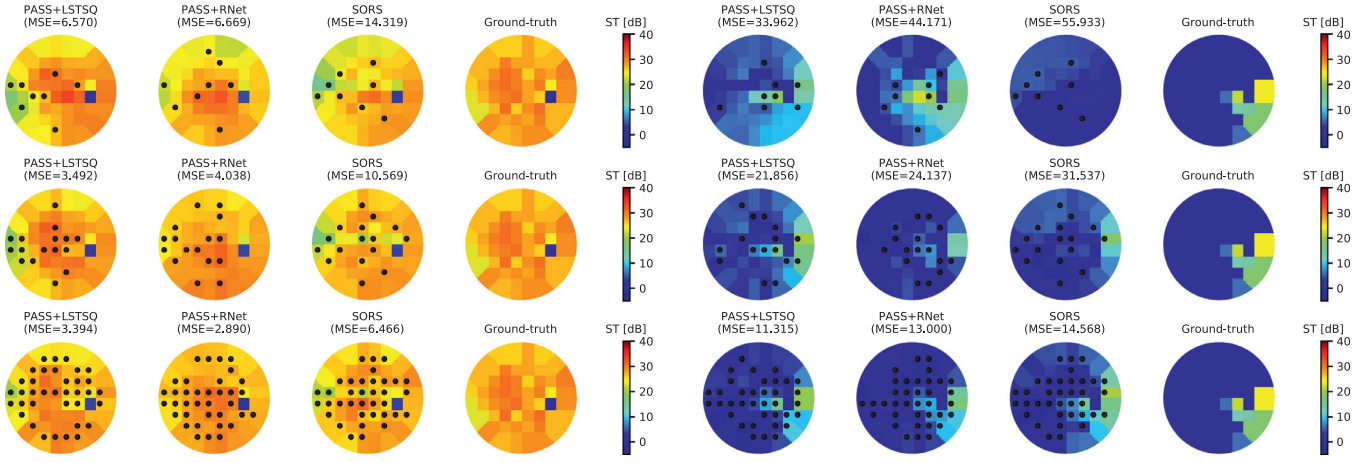
Table 6
Performance of methods on healhty and glaucomatous patients in the **Rotterdam** dataset. Mean MSEs are given, with mean nuumber of presentations in paranthesis. $T = T_{\text{test}}$ for all methods.

Method	$T_{\text{test}} = 8$		$T_{\text{test}} = 16$		$T_{\text{test}} = 36$	
	Healthy	Glaucoma	Healthy	Glaucoma	Healthy	Glaucoma
`PASS+RNet`	5.24 (29.55)	29.41 (26.80)	5.19 (59.09)	22.09 (51.43)	4.41 (124.61)	16.65 (110.61)
`PASS+LSTSQ`	8.61 (28.15)	31.04 (25.64)	5.06 (52.12)	22.82 (51.66)	3.98 (117.18)	14.49 (107.43)
`SORS`	9.12 (28.85)	31.85 (26.11)	6.88 (54.48)	23.46 (50.48)	4.60 (120.85)	15.69 (109.09)

on the three defect groups. Especially, for shorther horizon such as $T = 8$, the gap between `PASS` and `SORS` error performance is significant for almost the same number of presentations. For longer horizons such as $T = 36$, `PASS`+LSTSQ is consistently better than `SORS` as was observed in Table 3. Overall, `PASS` methods performed better than `SORS` for EG and AG in most cases (except `PASS`+LSTSQ ($T = 16$) and `PASS`+RNnet ($T + 36$). For MG, `PASS` has less evident superiority most likely due to insufficient training samples for that group.

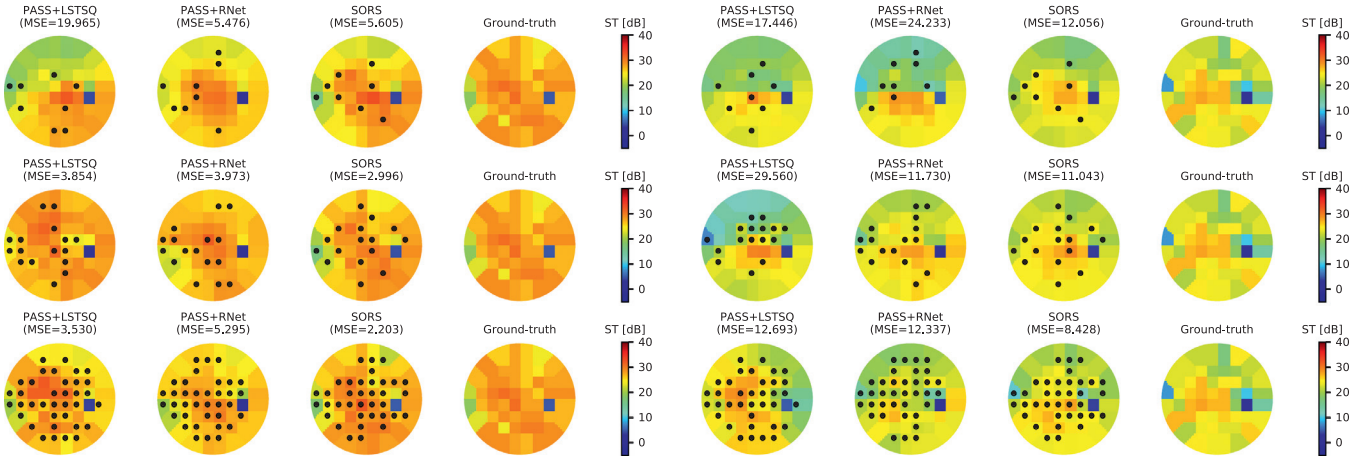
For the **Bern** dataset, similarly to the overall performance in Table 3, the advantage of `PASS` over `SORS` can be seen for EG for all horizons, whereas it is less apparent for MG and AG as was the case in Table 3. Since **Bern** dataset consists of mainly relatively healthy population, better performance, even though slight, can be expected from `PASS` as it had more examples to train within that population. The subtle difference between `PASS` and `SORS` is due to the fact that improvements on the EG group is marginal as VFs in that group are largely homogenous and easy to recover correctly by most strategies. This is also seen in the **Rotterdam** dataset where the smallest improvements were obtained on the EG group.

In addition to the performance comparison for different defect groups, we also present in Table 6 mean MSE and number of presentations results separately for healthy and glaucoma subjects in the **Rotterdam** dataset. For both healthy and glaucomatous subjects, `PASS` generally outperform `SORS` in terms of accuracy with more or less similar number of presentations. Exceptions to that fact are the cases of `PASS+RNet` for $T_{\text{test}} = 16$ and $T_{\text{test}} = 36$ where either only the number of presentations or both the MSE and the number of presentations are higher.



(a) Healthy patient (b) Glaucomatous patient

Fig. 5. (a) Comparison of VFs reconstructed with `PASS` (first and second columns) and with `SORS` (third column) for (a) a healthy patient and (b) a glaucomatous patient, both from the **Rotterdam** dataset. The last column shows the ground-truth. Results are shown for horizons $T = 8$ (top row), $T = 16$ (middle row), and $T = 36$ (bottom row). $T_{\text{test}} = T$ in all cases. Black dots indicate the queried locations for each method. `PASS` adapts queries to the underlying VF, while `SORS` uses a fix set of locations.



(a) Healthy patient (b) Glaucomatous patient

Fig. 6. Failure cases of PASS on a (a) healthy VF and (b) a glaumatous VF.

4.6. Qualitative evaluation

We now provide qualitative results of our approach, visualizing the selected locations and the estimated VF. Fig. 5a and b show representative results of our methods on healthy and glaucomatous cases, respectively.

In Fig. 5a, we present the results for a healthy VF where the overall sensitivity thresholds are high (>20, except for the blind spot region), whereas in Fig. 5b illustrates results for a glaucomatous case where the sensitivity thresholds are lower. We observe that the distribution of locations queried by our approach (PASS+RNet, PASS+LSTSQ) depends on the VF itself, whereas SORS selects the same locations for all VFs. Our approach tends to query locations within regions where the gradient of sensitivity thresholds is large. SORS locations are spread throughout the entire VF plane, as it has been trained to average over all kinds of VFs, overlooking patient-specific defects at test time. The fact that PASS reaches lower MSE than SORS quantitatively demonstrates the superiority of our strategy. Specifically, given that PASS+LSTSQ and SORS share the same reconstruction strategy, the improved performance of PASS+LSTSQ shows that the locations selected by PASS are more informative.

Fig. 6 shows two cases where PASS fail to outperform SORS. In Fig. 6a, which corresponds to a healthy VF, we see SORS performing very well for each test horizon. SORS locations are well spread throughout the VF and oriented at locations of both low and high threshold sensitivities which helps the final VF reconstruction. For this case, PASS+RNet has difficulty with longer horizons, especially when $T_{test} = 36$ whereby the MSE increases and becomes worse than cases with fewer locations tested. Conversely, PASS+LSTSQ has poor estimation for $T_{test} = 8$ due to the selected locations and overall underestimating the healthy region (i.e. region with high sensitivity threshold) within the VF. The VF estimation improves for longer horizons while still not outperforming

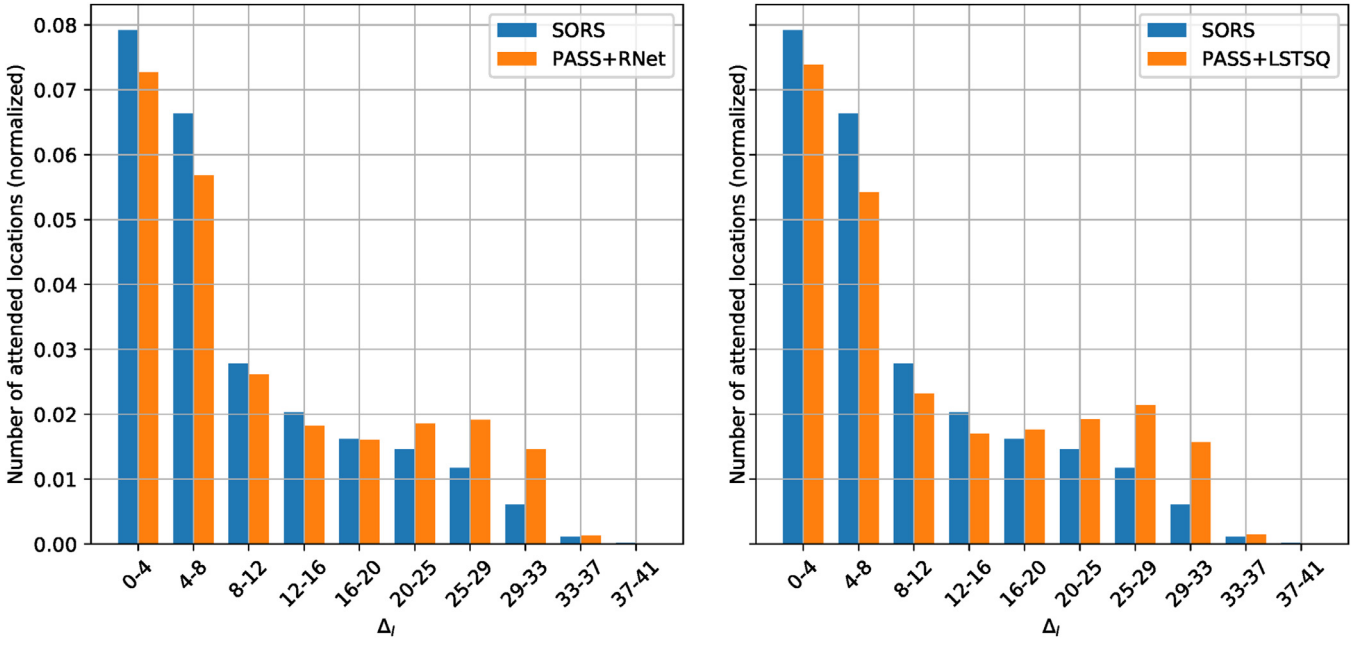


Fig. 7. Normalized bar plots of the first 8 locations selected by PASS and SORS (i.e. $T_{test} = 8$) with respect to their gradient measure Δ_l on the **Rotterdam** dataset. $T = 8$ for each model. Both plots show that PASS-selected locations are concentrated in higher gradient regions than those selected by SORS.

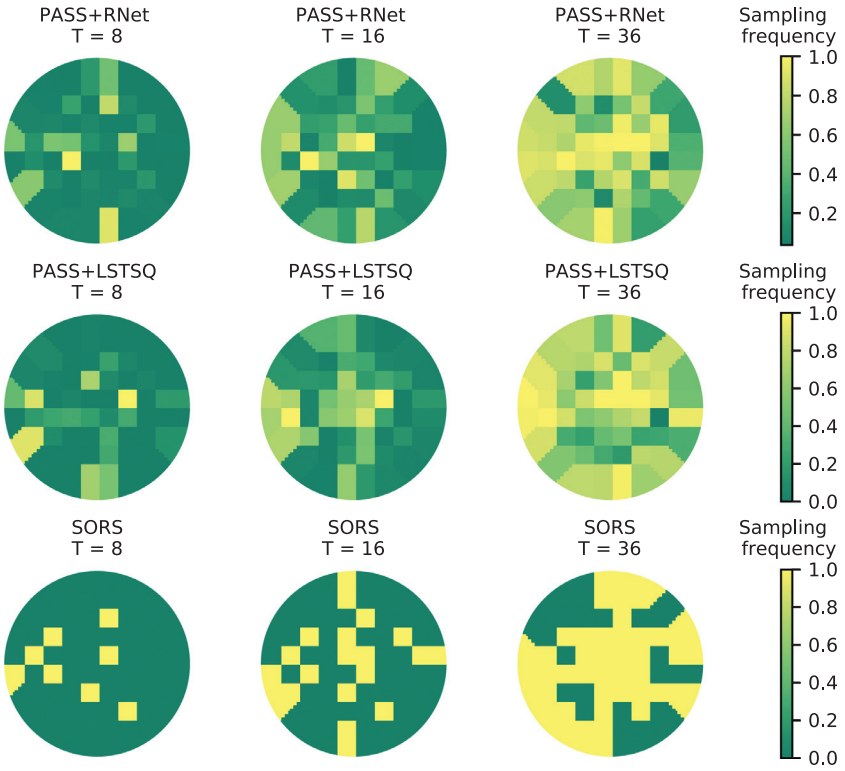


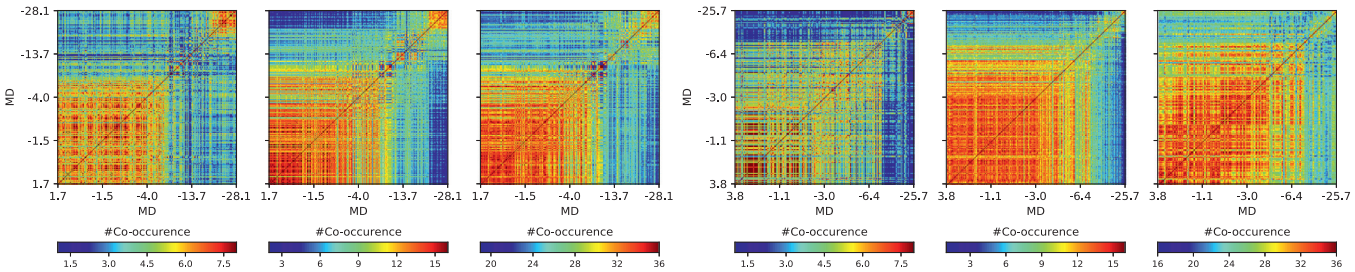
Fig. 8. Average tested locations on the **Rotterdam** test set. The proportion of times a location was selected is shown for `PASS+RNet` (top row), `PASS+LSTSQ` (middle row) and `SORS` (bottom row) for horizons $T = 8, 16, 36$.

`SORS`. In Fig. 6b, we illustrate a glaucomatous VF acqusition where there are a number of isolated defect regions. In this acquisition, we see that for $T_{test} = 8$, `PASS` methods underestimated the VF on the upper hemi-sphere. For $T_{test} = 16$, while `PASS+RNet` almost performed as well as `SORS`, `PASS+LSTSQ` concentrated its selection of locations to the upper-middle hemi-sphere making the estimation of other regions more difficult and resulting in a poor VF estimation. For $T_{test} = 36$, both `PASS` techniques reached similar MSEs but were still outperformed by `SORS`.

To support our claim that `PASS` methods generally attend regions that have high gradient, we present in Fig. 7 the normalized bar plots of the first $T_{test} = 8$ locations selected by `PASS` methods ($T = 8$) and `SORS` with respect to gradient in VFs. We quantify the gradient using $\Delta_l = \max_{l_n \in \mathcal{N}_l} |y_l - y_{l_n}|$ (Chong et al., 2014; Kucur and Sznitman, 2017) for the location l, where N_l is the set of (at most 8) neighboring locations. Δ_l is the highest difference between the sensitivity thresholds of a location and of its neighbors which quantifies the high gradient regions within a VF. Fig. 7 shows the probability of attending a location (in the first 8 time steps) within a given range of gradient measures. Accordingly, we observe that both `PASS`+RNet and `PASS`+LSTSQ first attended the locations having higher gradient measures than `SORS`.

To gain a broader view of what locations are selected by `PASS`, Fig. 8 illustrates how often a location is evaluated on the **Rotterdam** test set. Values of 1 indicate locations tested in each VF, while 0 occurs when a location is never tested. For this reason, `SORS` locations are the same regardless of the tested VF. In contrast, `PASS` adaptively selects location which is depicted by smoother averaged locations. Beyond this, Fig. 8 shows that `PASS` locations depend on the horizon T. This is visible as some locations at $T = 8$ having more importance than at $T = 16$ or $T = 32$. This indicates that `PASS` is not performing a greedy selection but adjusts based on its predefined horizon.

We also use the co-occurrence matrices of Figs. 9a and b to assess how our method adapts selected locations according to the severity of the VF defects. Each element (i, j) of a co-occurrence matrix shows the number of shared locations among those that our method selected for the i-th and the j-th VFs of the test set.



(a) **Rotterdam** (b) **Bern**

Fig. 9. Co-occurrence of selected locations using `PASS+RNet` on the (a) **Rotterdam** and (b) **Bern** test sets, with $T = 8$ (left), $T = 16$ (middle) and $T = 36$ (right). $T_{\text{test}} = T$ in all cases. Patients are sorted according to their mean defect (MD) values. Similar results are obtained for `PASS+LSTSQ`.

VFs are sorted according to their MDs. Elements to the left and bottom parts of the matrices correspond to healthier VFs. Higher values in the co-occurrence matrices indicate that the corresponding sets of queried locations are more similar.

Co-occurrence matrices show that the co-occurrence of queried locations are strongly related to the MDs. PASS queries similar locations for patients with similar VFs but adjust the selected locations in abnormal VFs, as can be deduced from the square blocks of high values that appear along the diagonals of the matrices. For example, in Fig. 9a the sets of selected locations are very similar for all VFs with MD > -5, which corresponds to relatively healthy cases and early glaucoma. This is due to the fact that healthy VFs are smooth and similar to each other. When MD ≤ -5, our approach selects sets of locations with fewer shared elements and tends to be more patient-specific, since abnormal VFs are different in each case. There are two small squared blocks of relatively high values, corresponding to the mild ($-12 <$ MD < -6) and advanced glaucoma patients (MD < -12) (Heijl et al., 2012), where VFs are mostly uniform with low ST values. A similar trend can be also observed in Fig. 9b concerning the **Bern** dataset.

5. Conclusion

In this work, we have presented a patient-specific perimetry strategy that leads to a better accuracy-speed trade-off in its ability to acquire visual fields. Our approach relies on reinforcement learning, visual attention and sparse approximation to provide a comprehensive framework for fast visual field acquisition. In practice, we decompose the problem in two by focusing on (1) a flexible model for VF reconstructions and (2) a novel method to select appropriate VF locations. By treating the selection of VF locations within a reinforcement learning context, we are able to learn a policy function that is optimized to iteratively select locations that reconstruct VF's effectively at the end of a fixed horizon. This departs from traditional greedy strategies that typically select locations based on the next best possible choice. We showed in our experiments, that even when using equivalent reconstructions methods, the locations selected with our approach outperform state-of-the-art methods. In addition, our method is patient-specific and adapts what locations are selected based on the VF itself, leading to improved accuracies at untested VF locations. In the future, we plan to determine strategies that select which of the proposed models is optimal to use depending on the patient and then extend our framework to other traditional imaging devices in medicine.

Acknowledgements

This work received partial financial support from the Haag-Streit Foundation, the Hasler Foundation Grant #18061 and the Center for Space and Habitability of the University of Bern.

References

Anderson, A.J., Johnson, C.A., 2006. Comparison of the ASA, MOBS, and ZEST threshold methods. Vis. Res. 46 (15), 2403–2411.
Anton, A., Yamagishi, N., Zangwill, L., Sample, P.A., Weinreb, R.N., 1998. Mapping structural to functional damage in glaucoma with standard automated perimetry and confocal scanning laser ophthalmoscopy.. Am. J. Ophthalmol. 125 (4), 436–446.
Bellman, R., 1957. Dynamic programming. Princeton University Press.
Bengtsson, B., Heijl, A., Olsson, J., 1998. Evaluation of a new threshold visual field strategy, SITA, in normal subjects. Acta Ophthalmol. Scand. 76 (2), 165–169.
Bryan, S.R., Vermeer, K.A., Eilers, P.H.C., Lemij, H.G., Lesaffre, E.M.E.H., 2013. Robust and censored modeling and prediction of progression in glaucomatous visual fields. Invest. Opthalmol. Vis. Sci. 54 (10), 6694.
Caicedo, J.C., Lazebnik, S., 2015. Active object localization with deep reinforcement learning. arXiv:1511.06015.
Chen, C., He, L., Li, H., Huang, J., 2018. Fast iteratively reweighted least squares algorithms for analysis-based sparse reconstruction. Med. Image Anal. 49, 141–152.
Chitsaz, M., Seng Woo, C., 2011. Software agent with reinforcement learning approach for medical image segmentation. J. Comput. Sci. Technol. 26 (2), 247–255.
Chong, L.X., McKendrick, A.M., Ganeshrao, S.B., Turpin, A., 2014. Customized, automated stimulus location choice for assessment of visual field defects. Invest. Opthalmol. Vis. Sci. 55 (5), 3265.
Chong, L.X., Turpin, A., McKendrick, A.M., 2016. Assessing the GOANNA visual field algorithm using artificial scotoma generation on human observers.. Transl. Visi. Sci. Technol. 5 (5), 1.
De Tarso Ponte Pierre-Filho, P., Schimiti, R.B., De Vasconcellos, J.P.C., Costa, V.P., 2006. Sensitivity and specificity of frequency-doubling technology, tendency-oriented perimetry, SITA standard and SITA fast perimetry in perimetrically inexperienced individuals. Acta Ophthalmol. Scand. 84 (3), 345–350.
Denniss, J., McKendrick, A.M., Turpin, A., 2013. Towards patient-Tailored perimetry: automated perimetry can be improved by seeding procedures with patient-Specific structural information.. Transl. Vis. Sci. Technol. 2 (4), 3.
Donoho, D., 2006. Compressed sensing. IEEE Trans. Inf. Theory 52 (4), 1289–1306.
Erler, N.S., Bryan, S.R., Eilers, P.H.C., Lesaffre, E.M.E.H., Lemij, H.G., Vermeer, K.A., 2014. Optimizing structure-function relationship by maximizing correspondence between glaucomatous visual fields and mathematical retinal nerve fiber models. Invest. Opthalmol. Vis. Sci. 55 (4), 2350.
Ganeshrao, S.B., McKendrick, A.M., Denniss, J., Turpin, A., 2015. A perimetric test procedure that uses structural information. Optom. Vis. Sci. 92 (1), 70–82.
Ghesu, F.C., Georgescu, B., Mansi, T., Neumann, D., Hornegger, J., Comaniciu, D., 2016. An artificial agent for anatomical landmark detection in medical images. In: Medical Image Computing and Computer-Assisted Intervention, pp. 229–237.
Gonzalez-Hernandez, M., Morales, J., Azuara-Blanco, A., Garcia Sanchez, J., Gonzalez de la Rosa, M., 2005. Comparison of diagnostic ability between a fast strategy, tendency-oriented perimetry, and the standard bracketing strategy. Ophthalmologica 219 (6), 373–378.
Haldar, J.P., Hernando, D., Liang, Z., 2011. Compressed-sensing mri with random encoding. IEEE Trans. Med. Imag. 30 (4), 893–903.
Heijl, A., Patella, V.M., Bengtsson, B., 2012. Effective Perimetry, 4 Zeiss Visual Field Primer.
Henson, D.B., Chaudry, S., Artes, P.H., Faragher, E.B., Ansons, A., 2000. Response variability in the visual field: comparison of optic neuritis, glaucoma, ocular hypertension, and normal eyes.. Invest. Ophthalmol. Vis. Sci. 41 (2), 417–421.
Hodapp, E., Parrish, R.K., Anderson, D.R., 1993. Clinical Decisions in Glaucoma. Mosby, St. Louis, Mo..
Huang, J., Zhang, S., Metaxas, D., 2011. Efficient mr image reconstruction for compressed mr imaging. Med. Image Anal. 15 (5), 670–679.
King-Smith, P.E., Grigsby, S.S., Vingrys, A.J., Benes, S.C., Supowit, A., 1994. Efficient and unbiased modifications of the QUEST threshold method: theory, simulations, experimental evaluation and practical implementation.. Vis. Res. 34 (7), 885–912.
Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv:1412.6980.
Kucur, S.S., Sznitman, R., 2017. Sequentially optimized reconstruction strategy: a meta-strategy for perimetry testing. PLoS ONE 12 (10), e0185049.
Lai, Z., Qu, X., Liu, Y., Guo, D., Ye, J., Zhan, Z., Chen, Z., 2016. Image reconstruction of compressed sensing mri using graph-based redundant wavelet transform. Med. Image Anal. 27, 93–104.
Mathe, S., Pirinen, A., Sminchisescu, C., 2016. Reinforcement learning for visual object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2894–2902.
Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K., 2014. Recurrent models of visual attention. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. MIT Press, pp. 2204–2212.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.
Morales, J., Weitzman, M.L., González de la Rosa, M., 2000. Comparison between tendency-oriented perimetry (TOP) and octopus threshold perimetry. Ophthalmology 107 (1), 134–142.
Neumann, D., Mansi, T., Itu, L., Georgescu, B., Kayvanpour, E., Sedaghat-Hamedani, F., Amr, A., Haas, J., Katus, H., Meder, B., Steidl, S., Hornegger, J., Comaniciu, D., 2016. A self-taught artificial agent for multi-physics computational model personalization. Med. Image Anal. 34, 52–64.
Neumann, D., Mansi, T., Itu, L., Georgescu, B., Kayvanpour, E., Sedaghat-Hamedani, F., Haas, J., Katus, H., Meder, B., Steidl, S., Hornegger, J., Comaniciu, D., 2015. Vito – a generic agent for multi-physics model personalization: application to heart modeling. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.
Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. NIPS-W.
Racette, L., Fischer, M., Bebie, H., Holló, G., Johnson, C., Matsumoto, C., 2016. Visual Field Digest. Haag-Streit.
Ranzato, M., 2014. On learning where to look. arXiv:1405.5488.
Ravishankar, S., Bresler, Y., 2011. Mr image reconstruction from highly undersampled k-space data by dictionary learning. IEEE Trans. Med. Imag. 30 (5), 1028.
Rubinstein, N.J., McKendrick, A.M., Turpin, A., 2016. Incorporating spatial models in visual field test procedures.. Transl. Vis. Sci. Technol. 5 (2), 7.
Sahba, F., Tizhoosh, H.R., Salama, M.M.A., 2008. Application of reinforcement learning for segmentation of transrectal ultrasound images.. BMC Med. Imag. 8, 8.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D., 2017. Mastering the game of go without human knowledge. Nature 550 (7676), 354–359.

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. MIT Press.

Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y., 2000. Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems, pp. 1057–1063.

Turpin, A., Artes, P.H., McKendrick, 2012. The open perimetry interface: an enabling tool for clinical visual psychophysics. J. Vis. 12 (11), 22.

Tyrrell, R.A., Owens, D.A., 1988. A rapid technique to assess the resting states of the eyes and other threshold phenomena: the modified binary search (MOBS). Behav. Res. Method. Instr.Comput. 20 (2), 137–141.

Wall, M., Woodward, K.R., Brito, C.F., 2004. The effect of attention on conventional automated perimetry and luminance size threshold perimetry. Invest. Opthalmol. Vis. Sci. 45 (1), 342.

Wang, L., Lekadir, K., Lee, S., Merrifield, R., Yang, G., 2013. A general framework for context-specific image segmentation using reinforcement learning. IEEE Trans. Med. Imag. 32 (5), 943–956.

Weber, J., Dannheim, F., Dannheim, D., 1990. The topographical relationship between optic disc and visual field in glaucoma.. Acta Ophthalmol. 68 (5), 568–574.

Weber, J., Klimaschka, T., 1995. Test time and efficiency of the dynamic strategy in glaucoma perimetry.. Ger. J. Ophthalmol. 4 (1), 25–31.

Wiering, M.A., van Hasselt, H., Pietersma, A.-D., Schomaker, L., 2011. Reinforcement learning algorithms for solving classification problems. In: IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, pp. 91–96.

Wild, D., Kucur, S.S., Sznitman, R., 2017. Spatial entropy pursuit for fast and accurate perimetry testing. Invest. Opthalmol. Vis. Sci. 58 (9), 3414.

Williams, R., 1988. Toward a Theory of Reinforcement-Learning Connectionist Systems. Northeastern University.

Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. 8 (3–4), 229–256.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057.