Learning Cooperative Personalized Policies from Gaze Data

Christoph Gebhardt[1,2], **Brian Hecox**[1], **Bas van Opheusden**[1,3], **Daniel Wigdor**[1,4], **James Hillis**[1], **Otmar Hilliges**[2], **Hrvoje Benko**[1]
Facebook Reality Labs[1], ETH Zürich[2], Princeton University[3], University of Toronto[4]

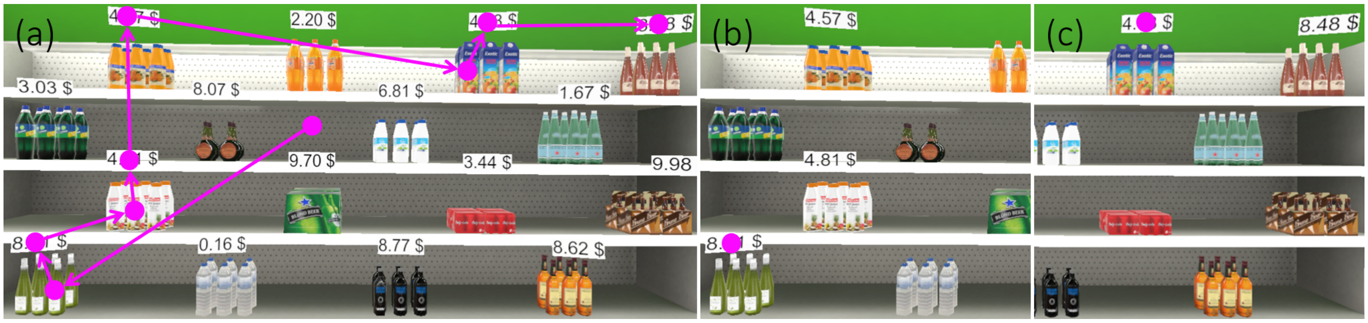

Figure 1. We propose a novel approach to learning personalized policies with applications to mixed reality. By observing (a) user gaze patterns when performing visual search tasks, such as browsing a supermarket shelf, our approach can learn a cooperative policy that implicitly identifies items of interest to the user (here: wine and juice). At runtime the policy only displays labels belonging to these categories and when the user looks at them (b, c). The approach does not require explicit supervision or explicit user interaction but relies only on gaze data to recover user preferences.

ABSTRACT

An ideal Mixed Reality (MR) system would only present virtual information (e.g., a label) when it is useful to the person. However, deciding when a label is useful is challenging: it depends on a variety of factors, including the current task, previous knowledge, context, etc. In this paper, we propose a Reinforcement Learning (RL) method to learn when to show or hide an object's label given eye movement data. We demonstrate the capabilities of this approach by showing that an intelligent agent can learn cooperative policies that better support users in a visual search task than manually designed heuristics. Furthermore, we show the applicability of our approach to more realistic environments and use cases (e.g., grocery shopping). By posing MR object labeling as a model-free RL problem, we can learn policies implicitly by observing users' behavior without requiring a visual search model or data annotation.

Author Keywords

reinforcement learning, eye tracking, mixed reality

CCS Concepts

•**Computing methodologies** → **Semi-supervised learning settings; Mixed / augmented reality;**

UIST'19 October 20–23, 2019, New Orleans, LA, USA

ACM ISBN .
DOI:

INTRODUCTION

Enhancing the real world via augmentation of objects is the central promise of Mixed Reality (MR). For instance, such a system could identify products of interest on a supermarket shelf and display additional information such as their price (see Figure 1, b and c). However, a naive solution of labelling every real-world object runs the danger of visually overloading the user (see Figure 1, a). In the worst case, such visually cluttered scenes with overlapping and occluding augmentations can hinder rather than support users. It has been shown that task performance declines simply due to the presence of non-task relevant visual features [32].

To reduce the number of visual features in a scene, the ideal MR system would only show labels if they are *spatially* and *semantically* relevant for users. Spatial relevance depends on individual visual search strategies determined by, for instance, users' average saccade length. Semantic relevance depends on a variety of factors, including the current task, previous knowledge and context, and user preferences. Determining the spatial and semantic relevance of a label is extremely challenging. We refer to this problem as the *MR object labeling* problem. Importantly, these factors are user-specific, requiring a labeling policy to be personalized. Many of these factors - for example preference - are also dynamic and can not necessarily be known a priori. Thus, the traditional approach of designing a static UI once and then applying it to all users does not scale well in the MR setting.

In this paper, we propose a method that learns to label objects according to user intentions by simply observing their gaze interactions with objects and labels. The resulting labeling policies support users by filtering information based on semantic and spatial relevance. We cast this as a Reinforcement Learning problem. Note that in standard RL, and its applications to graphics and robotics, agents are usually trained in a simulated physical environment to *imitate human behavior*, for example walking or playing games (e.g. [23, 19]). In contrast, we train an agent that *behaves cooperatively* in an RL-environment which *simulates a human*. More specifically, we propose a model-free RL-method that can learn cooperative personalized labeling policies to minimize the displayed labels in an MR-environment, without hiding relevant information.

To identify solutions for learning cooperative policies from human gaze behavior, we propose three main contributions. First, we formalize *MR object labeling* as a control problem using a Semi-Markov Decision Process (SMDP) where states are updated at each fixation and actions can span multiple time steps to account for the different duration of saccades. Second, we introduce an environment to train an RL-agent in a model-free setting by simulating gaze-object interactions of a particular user. To this end, we collected eye tracking data from experiments where all object labels are present. The RL-environment then plays back gaze trajectories at random allowing the agent to learn labeling policies that are coherent with the behavior of the simulated user. Third, we propose a reward function that allows the evaluation of the agent's actions only on the basis of human gaze behavior without the need for explicit user feedback. We learn labeling policies using a continuous state action value function represented with a RBF-parameterized function approximator [30].

We demonstrate the effectiveness of our method in various environments and for different tasks. For instance, by observing a user browsing a supermarket shelf (see Figure 1, a), the agent can learn that the user was interested in the prices of juices and wines and, in consequence, only display the labels of these objects if the user gaze is in proximity (see Figure 1, b). In addition, we also show that our approach works in visually more complex environments such as realistic indoor environments with object and label occlusions (see Figure 11). Even in such settings, our approach can still produce policies that distinguish between target and distractor objects.

We also demonstrate the benefit of our method via a user study in which participants solved a visual search task where they identified a target object among different 3D primitives. The study results indicate that our method has a higher perceived support than the baselines while reducing the displayed information by 87% compared to showing all labels at all time. This suggests that our approach performs well in filtering irrelevant labels while showing all required labels.

In summary, we contribute: i) the formalization of MR object labeling as a control problem using a SMDP, ii) a model-free approach that leverages recorded gaze-object interactions as environment for Reinforcement Learning, iii) an RL reward function that allows learning only on implicit user interaction (gaze) without explicit labels, and iv) the empirical evidence (through a lab-based user study) that policies found via our RL approach are helpful in visual search tasks.

RELATED WORK

Optimizing Properties of MR Labels

The optimization of label placement and properties, to ensure readability or to avoid occlusion of other objects is a long standing goal in MR. In their seminal work, Bell et al. [6] propose a method that modifies the properties of virtual labels (position, size, etc.) to maintain visual constraints, such as labels being located near their related objects, labels not occluding each other, etc. The method uses rectangular extents to project the visible parts of a 3D scene on the view plane where constraints are enforced. Extending this work, Azuma et al. [4] propose a gradient-descent-, a cluster- and a simulated-annealing-based algorithm to optimize label placement.

Other works propose methods that identify areas of less visual interest based on feature density [27] or visual saliency and detected edges [11]. This information is then used to optimize label layout or to adapt label rendering. Tatzgern et al. [31] propose a method that manages label placement in 3D object space instead of 2D image space to improve temporal consistency of the position of virtual labels on the view plane. This has proven to improve task performance compared to approaches that manage label placement in image space [20]. Leykin et al. [16] present a supervised learning method that automatically identifies if areas of interest have a textured background that affects the readability of text labels. In contrast to these works, we do not optimize label placement but *label visibility timing*, i.e., seek to control whether or not to show a label given users' gaze information and task.

User-Intention-Based Optimization of MR Labels

Another stream of research aims to optimize MR label assignment according to users' intention. Sharing the goal of our work, these approaches try to only label objects if users need the labels. In [13], Julier et al. present a mobile Augmented Reality system that based on users' state and properties of real-world objects filters augmented information. The importance of virtual elements is calculated by multiplying a user and an object state vector. These vectors contain information such as location, user intention or object importance. Note that subjective information such as user intention are not inferred by the system but assumed to be given. In contrast, our system learns by observing gaze behavior if users intend to look at the label of a particular object and adjusts the labeling accordingly.

In more recent work, Tatzgern et al. [32] propose an adaptive MR display which clusters virtual augmentations based on user-defined preferences in order to avoid visual clutter. User-preferences are set through an interface where sliders are used to specify item-preferences, or are estimated based on an item's click history. This approach only has limited applicability to MR as it can only be employed in applications where users interact with objects via clicking. It is easily conceivable that future MR glasses have a broader scope of use cases. Also, explicitly setting preferences will reach its limits in MR applications due to the vast number of real-world objects that can be encountered. In contrast, our method is capable of inferring users' label preferences only by observing their gaze-object interactions and does not require click histories or explicit preference setting.

Reinforcement Learning from Human Behavior

In recent years, RL has shown great promise in a variety of different domains, for instance, robotic control. In this section, we will focus on works that, like ours, use RL in combination with human data. While RL learns a policy given a reward function and an environment, Inverse Reinforcement Learning attempts to learn the reward function from a behavioral policy of a human expert [22] or to directly learn a policy from the behavior of a human expert [2]. This idea was successfully applied in the robotics domain [1, 8], to model human routine behavior [5], but also for character control and gaming [14]. Our work, in contrast, does not try to reproduce expert behavior, but learns a cooperative policy that, given users' behavior, chooses actions to support them in their task.

A stream of research applied Reinforcement Learning in combination with human motion capture data to improve policies for character animation and control [21, 33, 15, 19]. These works usually use an RL-agent to learn how to stitch captured motion fragments such that natural character motion as a sequence of clips is attained. A more recent work in character animation rewards the learned controller for producing motions that resemble human reference data [23] or directly learns full-body RL-controllers from monocular video [24]. In a similar fashion, Aytar et al. [3] use YouTube-videos of humans playing video games to specify guidance for learning in cases where the normal reward function only provides sparse rewards. These works employ human behavioral sequences either as the agent's actions or as reference motion to provide additional rewards for training. In contrast, in our work, human behavioral sequences form the RL-environment and the agent learns a policy that is complementary to users' behavior.

Most related to our work are approaches that treat users as the environment of an RL-agent to learn policies from explicit user feedback. For example, in the domain of dialog manager systems, RL-agents were learned based on users' responses to automated speech segments selected by a policy [10, 29]. Hu et al. [12] use Reinforcement Learning to learn an incentive mechanism which maximizes the quality and throughput of crowdsourcing workers. In this case, the agent's actions are different payment scaling factors and its reward is based on the estimated accuracy of labels (for supervised learning tasks) provided by crowdsourcing workers. Reinforcement Learning is also used in recommender systems for domains with sequential recommendations (e.g., online video platforms, music streaming services) [7, 18, 17]. Here, the agent's actions are videos or songs available for recommendation and the environment is users reacting on the recommended item. These works require explicit user interaction with the agent's action (e.g., clicking recommended song, working with higher accuracy due to higher incentive) to calculate the reward. In contrast, we learn policies entirely on implicit user behavior, i.e., no explicit user feedback. More specifically, we propose a reward function that allows the evaluation of the agent's actions only on human gaze behavior.

METHOD OVERVIEW

We propose a reinforcement learning based method to implicitly learn personalized cooperative labeling policies from gaze behavior. The goal of these policies is to support users in their task while avoiding visual overload. Our method is inspired by observations from a formative study, which we detail first. Leveraging gaze data and insights on user behavior during visual search tasks, we then detail our method including the exact task, the RL-environment and reward function that are necessary to train MR-labeling policies. Importantly, the proposed approach does not require any explicit labels and does not rely on explicit user interaction. Furthermore, the resulting policies are not attempting to perform the same task as a human would independently but rather they're trained to automatically adjust the UI such that it supports the user in the current task as well as is possible.

DATA COLLECTION STUDY

To train our RL agent, we require gaze trajectories. We collected this data via eye tracking in two well-defined visual search tasks. Participants were asked to identify targets among a set of objects based on their labels. For example, to find objects of a certain kind, displaying the highest value on their label. Objects are 3D primitives positioned on a shelf-like virtual environment (see Figure 2) and all the labels are present.

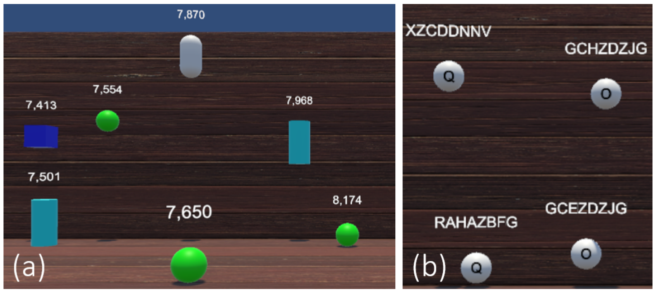

Figure 2. Virtual environment of the visual search task showing (a) the highest-number task with pre-attentive features and (b) the matching-strings task with attentive features.

Experimental Design

Object features:

Objects possess pre-attentive or attentive features [34, 35]. Attentive features require humans to focus on the object itself in order to distinguish two different kind of features. In contrast, pre-attentive features can be recognized in peripheral vision, allowing feature distinction without needing to fixate the object. In our data collection, we use color and shape as pre-attentive features (see Figure 2, a) and the letters "O" and "Q" as attentive features (see Figure 2, b). In one trial, all objects either have attentive or pre-attentive features.

Tasks:

The data collection comprises of two tasks: 1) Participants have to find the target object with the highest number on its label (see Figure 2, a). 2) Participants have to find two target objects with matching strings on their labels (see Figure 2, b). Target objects are green spheres in the case of pre-attentive features and spheres marked with a "Q" when using objects with attentive features. In both tasks, participants had to search for a target object until they found it.

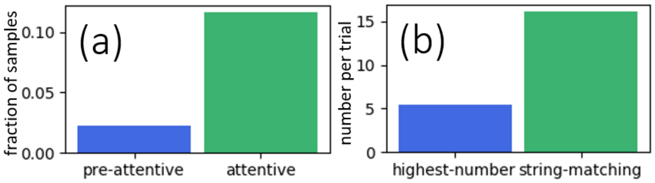

Figure 3. (a) Mean fraction of samples in which participants looked at objects with pre-attentive versus attentive features. (b) Mean number of times per trial in which participants looked at labels of target objects in the highest-number versus the string-matching task.

Conditions:
This 2x2 design results in four conditions: 1) pre-attentive features in highest-number task, 2) pre-attentive features in matching-strings task, 3) attentive features in in highest-number task, 4) and attentive features in matching-strings task (listed in the order of presentation).

Apparatus:
The visual search environment is implemented in the Unity game engine and rendered in Virtual Reality. Participants could see the scene through an HTC Vive headset with integrated Tobii Pro eye tracking. We logged participants' gaze data relative to object and label positions, and their pupil dilation. All data were logged at 120Hz (operating frequency of eye tracker). In post-processing, we ran the eye tracking event detection algorithm of [9] to estimate fixations and saccades.

Procedure:
The eye tracker was calibrated for each participant. After calibration, participants solved the visual search task of each of the four conditions for ten minutes. Object labels were shown at all time in all conditions.

Participants:
We conducted our study with 14 participants (5 female, 9 male). They were recruited via email from the participant pool of our institution. All participants reported normal vision.

Formative Analysis

Our formative analysis of a total of 1300 visual search trials revealed that gaze patterns depend on the *object features* and the *task*. First, we calculate the fraction of eye tracker samples in which participants' gaze rays intersect with an object in the scene. Figure 3, a) shows that this fraction is higher for objects with attentive compared to objects with pre-attentive features. This observation provides further evidence that attentive features require foveal processing while pre-attentive features do not. Second, we compute the number of times per trial in which participants looked at a label of an object, i.e. their gaze rays intersect with a label. Figure 3, b) indicates that the labels of target objects are focused fewer times in the highest-number than in the string-matching task. This confirms existence of differences in gaze movements when solving both tasks. In the highest-number task, participants can memorize the current highest number and compare it against the number of an unseen label. In the matching-strings task users have to go back and forth between the characters displayed on labels to assess string similarity.

METHOD

Our goal is to use a Reinforcement Learning approach to learn personalized cooperative labeling policies from gaze behavior. A common goal of standard RL is to learn policies that can mimic human behavior. Therefore, an agent selects actions (e.g., move a paddle) to achieve a certain behavior (e.g., playing the video game Pong as well or better than a human). These actions are passed to an environment that has a finite set of possible responses to those actions, a game environment or a physical simulator. The agent is then rewarded or penalized according to a goodness function that is designed to lead the agent to a goal, for instance bouncing the ball past the other player in the game of Pong. This is not directly applicable for our agent that learns to cooperate with humans because it would require a complete and realistic simulation of users.

In contrast, we pose our problem such that users' behavioral data function as the RL-environment. In other words, the agent makes observations about the user behavior (i.e., data from our data collection experiment) to learn to *cooperate* with the user on a certain task. More specifically, the agent observes gaze trajectories and learns to label or not to label objects based on a function that rewards its actions given only the user gaze behavior and no form of further explicit supervision. Figure 4 visualizes the differences between standard RL (blue) and our setting (green). In the following subsections, we will explain the individual components of our RL method: the underlying decision process, the state action space of the agent, the RL-environment, the reward function, and the learning procedure.

Decision Process

Saccades are considered as ballistic movements since humans cannot respond to changes in the position of a target, while undergoing rapid eye movements [25]. Therefore, the decision where to fixate next has to be made *before* the start of a saccade. To better support users in our setting, it is important that the agent decides to show or hide a label at the time the planning decision is made. Therefore, we design our RL-environment to update the state of the user gaze only at fixations (subject to the sampling frequency of the eye tracker) and to ignore state changes during saccades. The resulting setting can be seen as a dynamical system which does not update its state at a fixed sampling rate, but which provides snapshots of its state at decision points where the time between samples can vary.

This control problem can be formalized as a Semi-Markov Decision Process (SMDP). An SMDP is a generalization of a

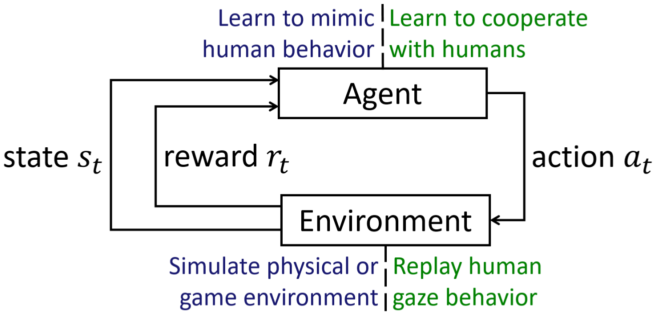
Learn to mimic human behavior
Learn to cooperate with humans
Agent
state s_t
reward r_t
action a_t
Environment
Simulate physical or game environment
Replay human gaze behavior

Figure 4. Comparing standard RL with our setting.

Markov Decision Process (MDP) that can handle actions of different temporal length. Specifically, it is defined as a five-tuple (S, A, P, R, F), where S is a set of states of the world and A is a set of actions. P is the state transition probability function specifying the probability of going from a state s to state s' after performing action a (i.e. $P(s'|s,a)$). R is the reward function determining the reward obtained by performing action a in state s (i.e. $R : S \times A \rightarrow \mathbb{R}$) and F is a function giving the probability of transition times for each state-action pair (i.e. $F(t|s,a)$, the probability that the next decision epoch occurs within t time units). The expected discounted reward for taking action a in state s and then following policy π is known as the Q value and is defined as $Q^{\pi}(s,a) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)]$, where γ is a discount factor. Q values are related to another through the Bellman equation:

$$Q^{\pi}(s,a) = \sum_{s',t} F(t|s,a)P(s'|s,a)[R(s,a) + \gamma^t Q^{\pi}(s',\pi(s'))]. \tag{1}$$

The optimal policy can be computed as $\pi^* = argmax_a Q^{\pi}(s,a)$. $Q^{\pi}(s,a)$ is called the optimal state-action-value function. In our case $Q^{\pi}(s,a)$ tells us how much reward can be collected by showing the label of an object given the current gaze position (as attained from the reward function Eq. 4). The state-action value $Q(s,a)$ is formed by summing the reward of the current gaze position with the reward of all possible gaze positions that can follow normalized by their probability. Thereby, the reward of future gaze positions gets discounted. This depends on the difference between the current time and the time they are normally encountered (after visiting the current state). Note that in model-free RL (our setting), the distributions $P(s'|s,a)$ and $F(t|s,a)$ are not explicitly available but are implicitly approximated through experiences during learning.

State and Action Space

The agent will need to decide whether to show a label for each object in the scene. A naïve way to represent the agent's state would be to take the geometric relations of all objects with respect to user's gaze in the world coordinate frame of the virtual scene. However, this would result in large state- and action-spaces, rendering generalization to unseen scenes difficult. A more compact state space representation is given by the geometric relation of the gaze point with respect to the center of an individual object. The agent then decides label visibility for all objects in the scene. The state space is defined in the local coordinate frame of an object with its center as origin (see Figure 5, a). More concretely, state and action space are given by:

$$s = [b_o, \alpha_{go}, \alpha'_{go}] \tag{2}$$

$$a = \begin{cases} show \\ hide \end{cases} \tag{3}$$

where α_{go} is the angle between gaze unit vector n_g and gaze to object center vector r_{go} (see Figure 5, b) and α'_{go} is the angular velocity calculated by taking finite differences between two consecutive values of α_{go}. b_o is a one-hot vector encoding for object properties which in the particular case of our visual search task is a binary feature to distinguish between Os and Qs or spheres and other primitives. The actions are to show or to hide the label of an object. Euclidean distance is not included in the state space as it caused results to deteriorate.

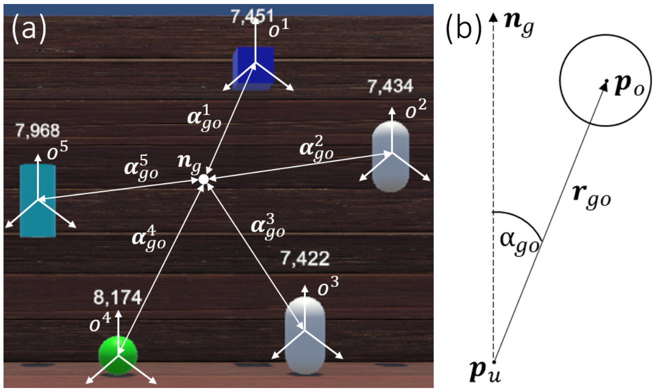

Figure 5. (a) Angles α_{go}^{1-5} between objects o^{1-5} and gaze ray $\mathbf{n}_g$ in the local coordinate frames of the objects from the perspective of the user. (b) Geometric relations between the gaze unit vector $\mathbf{n}_g$, position of object $\mathbf{p}_o$ and user $\mathbf{p}_u$, gaze object vector $\mathbf{r}_{go}$, and the gaze object angle α_{co}.

RL-Environment

Due to the high stochasticity of eye movements, it is impossible to analytically model the state transition dynamics between two consecutive samples of the eye tracker. However, if it is possible to draw a large number of samples from an RL-environment, in which state transitions follow the true transition dynamics probability distribution, model-free RL approaches have been shown to be able to learn useful policies [30]. By training on a large corpus of human gaze traces (ca. 90 visual search trials per participant) and given the small state space ($s \in \mathbb{R}^3$), we assume this assumption to hold in our case. Hence, we propose an RL-environment that enables model-free learning of policies on human gaze data.

For each object in the scene, we generate a trajectory from gaze recordings. This is constructed by transforming the gaze point from the global into an object-centered coordinate frame. For each such trajectory we calculate the state as defined in (2) for each detected fixation. Figure 6 shows the resulting trajectory consisting of a sequence of states that depict the movement of user gaze with respect to a particular object (green sphere) in one trial. Note that state-to-state transitions of our RL-environment are independent from the chosen action, i.e., for one trajectory the transition from s_t to s_{t+1} is independent of a_t. This allows the exploration of different action sequences for the same state trajectory by simulating it multiple times. Nevertheless, we stay within the RL-framework as model-free RL learns value estimates for particular state-action pairs.

The design of this environment assumes that participants behave compliantly. That is, we assume that they only take actions that are necessary for the visual search task during data collection, looking only at the labels of target objects and ignoring distractors. In the case of complete random gaze patterns, it would not be possible to extract meaningful cooperative policies. We assume and show experimentally that the correct search behavior can be recovered by the agent when exposing it to a sufficient number of trials.

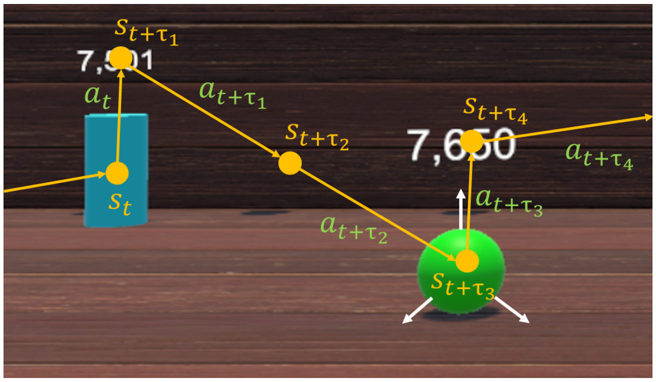

Figure 6. A gaze trajectory with respect to a particular object (the green sphere) specifying the progression of states and the decision sequence of actions the agent is learning.

Reward Function

In Reinforcement Learning it is necessary to provide a reward function that the agent can query in order to evaluate the goodness of a chosen action. In our case, we model the reward function to depict our goals of supporting users in the visual search tasks while reducing the amount of displayed information. This is broken down into two factors. First, we want to always show a label when it is needed, represented by the reward r_l. Second, we want to minimize the number of shown labels in total, specified with the reward r_c. The full reward function is then defined as

$$r(s,a,s') = \begin{cases} r_l & \text{if } a \text{ is } \textit{show} \text{ and label is fixated in } s' \\ -r_c & \text{if } a \text{ is } \textit{show} \text{ and label is not fixated in } s' \\ -r_l & \text{if } a \text{ is } \textit{hide} \text{ and label is fixated in } s' \\ r_c & \text{if } a \text{ is } \textit{hide} \text{ and label is not fixated in } s'. \end{cases} \tag{4}$$

We consider a label to be fixated if α_{go} is zero and if the algorithm of [9] detects a fixation. All four if statements are necessary to avoid convergence to cases where either all or no labels are shown. Empirically, we derived that reasonable policies are attained with the reward values $r_l = 10$ and $r_c = 1$.

Learning Procedure

With the RL-environment and the reward function in place we can now run standard algorithms like Q-learning and SARSA [30] to learn an approximation of the state action value function. Due to the small state space it is sufficient to represent the continuous state action value function $\hat{q}(s_t, a_t, \mathbf{w}_t)$ with a RBF-parameterized function approximator (cf. [26]). In our experiments more powerful function approximators, such as deep neural networks, did not yield performance improvements. For SARSA, the function's update rule is as follows:

$$\begin{aligned} \mathbf{w}_{t+1} = \mathbf{w}_t + \alpha[r_{t+1} + \gamma^{\tau_t}\hat{q}(s_{t+1}, a_{t+1}, \mathbf{w}_t) - \\ \hat{q}(s_t, a_t, \mathbf{w}_t)] \bigtriangledown \hat{q}(s_t, a_t, \mathbf{w}_t) \end{aligned} \tag{5}$$

where $\mathbf{w}_t$ is the parameter vector of the state action value function and $\bigtriangledown$ denotes the gradient of function $\hat{q}$. In accordance with the underlying SMDP and to account for the varying duration of saccades, an action a_t can be of different temporal length, modeled by τ_t. Eq. 5 corresponds to performing standard stochastic gradient descent on the state action value function. Using epsilon-greedy exploration, the agent then learns for a particular state s_t to show or to hide the label a_t in the next state s_{t+1} in correspondence to the reward provided by Eq. 4 (see Figure 6).

TECHNICAL EVALUATION

In this section, we want to quantitatively and qualitatively investigate the nature of the policies learned with our method. Thus, we run an experiment in which we analyze if policies learn to behave cooperatively given our RL-environment and reward function. Therefore, we train the agent on all gaze trajectories of one trial. After each hundred samples, we test the current policy by applying it on the trajectories of an unseen trial. For each of these tests, we save the attained accumulated reward of the policy on the particular test trial. We ran the experiment on five random trials of all four conditions for all participants. The results can be seen in Figure 7.

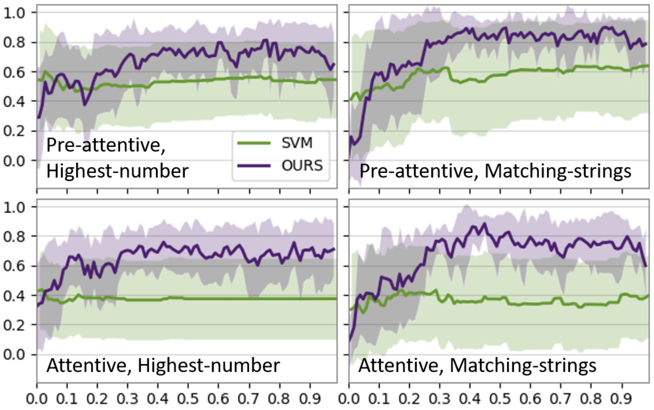

Figure 7. Performance comparison between *ours* (in purple) versus an *SVM*-based baseline (in green). Solid line denotes average normalized reward on an unseen test trial (y-axis) over percentage of experienced training samples (x-axis). The shaded area represents the standard deviation. Ours attains higher rewards and continues to learn from experiencing more samples, whereas the baseline converges to a low reward, displays high variance and does not improve with additional samples.

In all conditions, our policies improve with increasing number of training samples and converge to a high normalized reward towards the end. The relatively high variance, indicated by the shaded area in the plots, can be explained by the non-deterministic RL-environment. The agent attains samples from a highly stochastic transition probability distribution, but still manages to converge.

To provide a comparison with a sensible baseline we conduct the same experiment with a supervised machine learning approach. To this end, we choose a Support Vector Machine (SVM). The SVM has comparable model complexity and discriminative power to our RBF-based function approximator. Employing the SVM as a binary classifier, we use the same state representation as in the RL-setting (only considering fixations). We then assign respective labels (show/hide) to individual states, based on whether the object label was fixated or not in the following time step. Results show that policies which have been learned with explicit supervision attain a lower reward than policies learned with our method. That is they more often make decisions that are in conflict with the

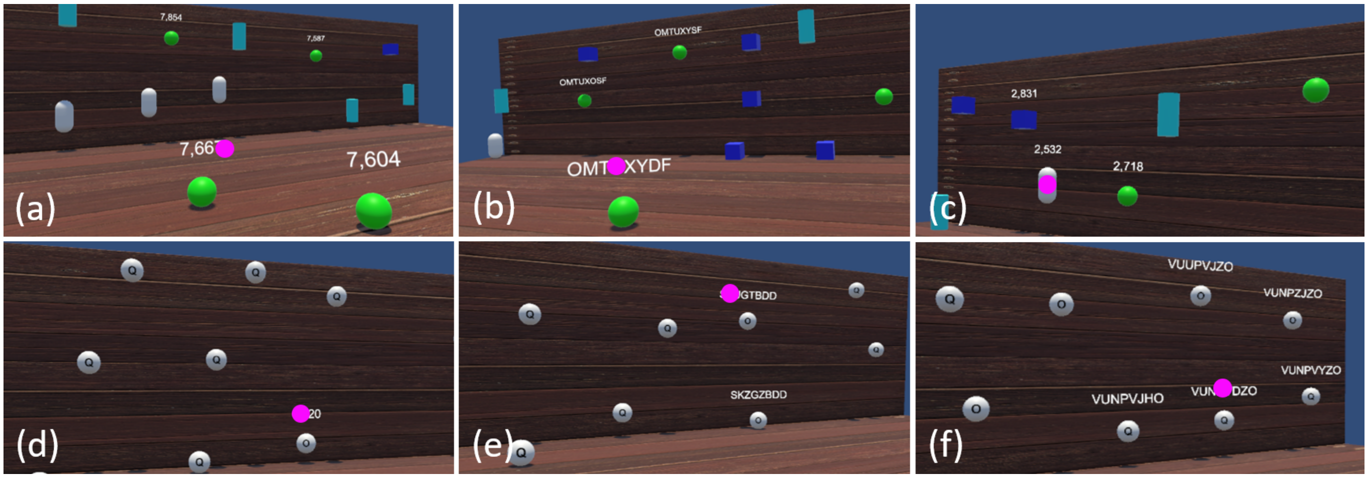

Figure 8. The labeling output of policies learned in our experiment (pink dot is user's focus) for highest-number task (a) pre-attentive and (b) attentive object features as well as matching-strings task (d) pre-attentive and (e) attentive object features. Also, the output of supervised policies learned for (c) pre-attentive and (f) attentive object features is shown.

true user behavior. Furthermore, the shaded area in Figure 7, indicates that the supervised policies produce a higher variance in rewards than RL-policies and thus are less stable. Finally, we highlight that supervised policies show zero improvement with increasing number of training samples, indicating that the SVM mostly learns an angular threshold around users' gaze (see next paragraph). This threshold can be determined with a small number of samples but it does not suffice to fully capture the underlying decision process of the user. Recall that the next fixation point is determined ahead of the saccade, which requires more involved reasoning.

To assess if learned policies are useful we investigated their output with new users on unseen trials in the VR environment. We perform this test with policies of the SVM and our RL-method. Policies were trained on gaze trajectories of all available trials per condition and participant. Interestingly, policies of the supervised setting mostly converged to showing labels within a certain angle around the current fixation point (see Figure 8, c). In contrast, the RL-agent learns to distinguish between target and distractor objects and only displays the labels of targets (see Figure 8, a, b, d, and e). An intuition behind the different labeling strategies of supervised- and RL-policies is that the shape of the reward function helps the agent to distinguish between target and distractor objects. By giving a higher reward to correctly displaying a label as opposed to correctly hiding it, the agent can more easily identify the type of objects for which a label is needed.

Finally we note that when comparing policies trained on different tasks there is no perceivable difference in their behavior. However, the agent learns qualitatively different policies when trained on attentive versus pre-attentive features. In the case of pre-attentive features, the agent either shows the labels of all target objects (see Figure 8, a) or labels of target objects within a certain angle of the user's gaze (see Figure 8, b). For attentive features, this angle is smaller, resulting in cases where only the label of one (see Figure 8, d) or two objects (see Figure 8, e) in the scene are shown. These labeling strategies reflect the difference in human search behavior for attentive and pre-attentive features (see Section 4).

Interestingly, our approach is also capable of distinguishing fixations and saccades. Including angular velocity in the state space allows agents to distinguish whether users are saccading over or fixating on a given label. (see video 2:30 - 2:45).

USER STUDY

The goal of our method is to learn cooperative policies that support users while minimizing information overload. To evaluate the success of our approach, we conducted a user study in which a new set of participants solved the visual search tasks of the data collection study with the help of our RL-method labeling policies. We compare participants' task performance using our RL-method with three other baselines. In this experiment, the tasks, object features, and apparatus are identical to those used during data collection.

Experimental Design

Conditions:

In addition to the two object features and two tasks of the data collection, four different policies are introduced as conditions:

1. Showing labels of all objects at all time (SA = "Show All").
2. Showing one label at a time corresponding to the object with the closest angular distance (according to α_{go}, see Figure 5)to the user's gaze ray (CO = "Closest Object").
3. Showing labels of objects according to predictions of an SVM[1] (SL = "Supervised Learning").
4. Showing the labels of objects according to labeling policies learned by our method (RL = "Reinforcement Learning").

This results in a 2x2x4 design with a total of 16 conditions. For the SL- and RL-conditon, we qualitatively evaluated policies and picked the ones we assessed to behave the best given task and object features (Figure 8 a), b), d), e) show their output).

[1] Supervised training is explained in Sec. "Technical Evaluation".

Procedure:

At the start, the Tobii eye tracker in the HTC Vive headset was calibrated for each participant. After that, participants completed the 16 conditions of the study. The order of the first 2x2 condition was fixed and the same as in the data collection study (see Section 2). Participants solved each 2x2 condition with the four labeling policies (SA, CO, SL, RL). The order of labeling policies was counterbalanced according to Latin Square. In each of the 16 conditions, the participants solved the respective visual search task five times. After finishing one condition, participants completed a questionnaire with two Likert items asking for perceived support and disruption of a policy. A session took on average approximately 72 minutes (without briefing and debriefing).

Participants:

12 people participated in our study (5 female, 1 non-binary, 6 male). They were recruited via email from the participant pool of our institution. Everyone reported normal vision.

Results

We analyze the effect of conditions on supporting users in accomplishing the task and on reduction of unnecessary information. It is important to consider both goals to avoid degenerate solutions, such as hiding all labels, which would negatively impact the task performance. For significance testing, we use Friedman's test since the normality assumption of the data is violated. Tasks were treated as repeated measures. Pairwise comparisons were performed using Connover's post-hoc tests with Holm-Bonferroni-adjusted p-values.

Task Performance Analysis:

To analyze task performance, we compare the task execution times across conditions (see Figure 9, a). Friedman's test did not reveal any significant differences ($H(3) = 1.52, p = 0.68$).

We also analyze perceived support and disruption of policies. Perceived support across conditions differs significantly ($H(3) = 10.18, p < 0.001$). Pairwise comparison shows that participants perceived RL to be significantly more supportive than CO and SL (both $p = 0.02$). All other pairwise differences are not significant. No significant effect of conditions on perceived disruption has been found ($H(3) = 6.80, p = 0.08$).

We do not consider falsely reported objects in this analysis since only 56 false reports were made in 960 trials (no significant differences between conditions).

Visual Clutter Reduction Analysis:

To analyze the four conditions against our second goal of minimizing the amount of displayed information, we calculate the percentage of labels which are shown over all objects and frames per condition. This measure is defined as

$$lp = \frac{1}{N_f N_o} \sum_{f,o}^{N_f,N_o} \mathbb{1}_L(f,o). \tag{6}$$

$$\mathbb{1}_L(f,o) = \begin{cases} 1 & \text{if label of } o \text{ is shown in } f \\ 0 & \text{if label of } o \text{ is not shown in } f, \end{cases}$$

where o is the object, f is the frame, N_f is the number of frames of a particular trial and N_o is the number of objects shown in that trial. The values for the conditions SA and CO can be analytically computed. To attain the percentage of shown labels for the conditions SL and RL, we average the logging data over all trials and participants for a particular condition. According to this measure, CO shows the lowest fraction of labels (4%), followed by SL (9%), and RL (13%). SA shows all labels at all time (100%, see Figure 9 d).

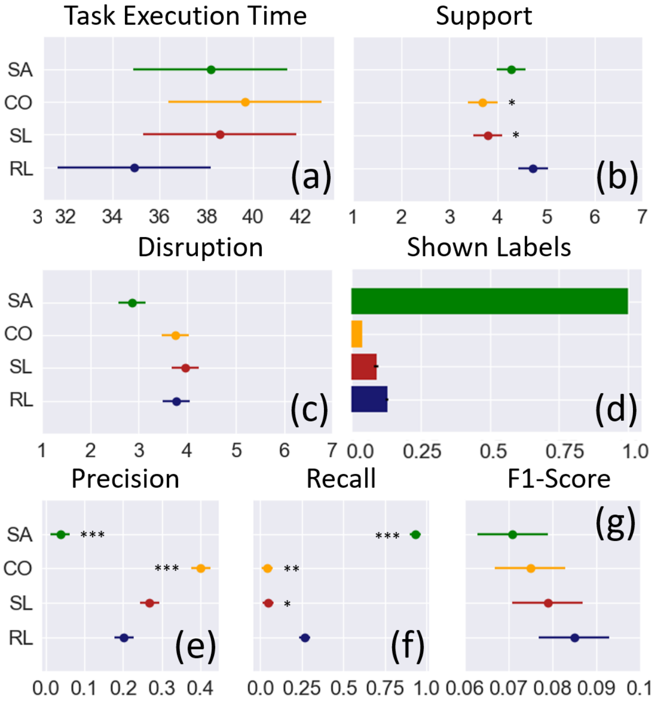

Figure 9. Mean and 95%-confidence interval of (a) task execution time (in seconds), (b) perceived support and (c) disruption (Likert-item range was one to seven, higher number standing for more support / disruption), (d) fraction of shown labels, (e) precision, (f) recall, and (g) F1-score. Significance notation is with respect to the condition RL.

Label Relevance Analysis:

We also examine if the shown labels of objects were relevant to participants during the visual search tasks. Therefore, we parse the gaze traces of all target objects of all conditions. For each sample, we intersect participants' gaze rays, given by the eye tracker, with all labels in the scene to determine if a label was focused. Samples are then categorized as follows:

i) the label of a target object is shown and user looks at it (true positives, TP),

ii) the label of a target object is shown and user does not look at it (false positives, FP),

iii) the label of a target object is not shown and user does not looks at it (true negatives, TN),

iv) and the label of a target object is not shown and user looks for it (false negatives, FN).

We then calculate precision ($TP/(TP+TN)$) and recall ($TP/(TP+FN)$) of the four conditions.

Precision is defined by the fraction of shown labels of target objects that were looked at by participants. Results show that CO has the highest precision followed by SL and RL. By showing all labels all the time, SA consequently ranks last (see Figure 9, e). The statistical analysis revealed that the differences are significant ($H(3) = 62.07, p < 0.001$). Significance holds for all pairwise comparisons ($p < 0.001$) but RL to SL.

Recall is defined by the fraction of samples where an object label was shown when it was needed. Thereby, we assume that users express the need for a label by looking at its position, no matter if the label is shown or not. By showing all labels at all time, SA naturally has the highest recall, followed by RL, SL, and CO (see Figure 9, f). Again, there is a significant effect of conditions on recall ($H(3) = 105.84, p < 0.001$), holding for all pairwise comparisons ($p < 0.02$) but CO and RL.

To summarize precision and recall in a single relevance measure, we calculate their harmonic mean, the F1-score (see Figure 9, g). Statistical testing revealed no significant effect of conditions on relevance ($H(3) = 2.78, p = 0.43$).

Discussion

The results of the study provide evidence that our method (RL) can learn policies that support users in their task while reducing the amount of unnecessarily shown labels. Statistical testing did not find significant differences in task execution time, disruption, support and F1-score between our method and the baseline of showing all labels at all times (SA). Nonetheless, RL reduces the amount of shown labels compared to SA by 87%. Likewise, the conditions CO and SL only show a fraction of the labels of SA. However, participants perceived they were significantly better supported by our policies compared to CO and SL. We attribute this to the fact, that our method decides to show the label of an object not only based on spatial information (e.g., closest distance to gaze ray) but also learns and considers the semantic relevance of the object for the task.

ADDITIONAL USE CASES

While our user study has shown that our method works for visually reduced virtual environment and for relatively simple visual search tasks, we are also interested in the applicability of our approach to more realistic environments and tasks. Therefore, we apply our method in two additional virtual scenarios: a supermarket and an apartment. With these tests we want to investigate if our method can still learn useful policies in more complex task scenarios (supermarket) and in scenes with almost photo-realistic graphics with salient distractions and occlusions (apartment).

Supermarket Scenario: More Realistic Task

We implemented the supermarket environment to employ our method in a more realistic task scenario. In this scenario, a participant was asked to search for the cheapest drink of a particular class of drinks (water, juice, soda, etc.) in a virtual supermarket shelf. In our previous experiments, we distinguished between target and distractor objects by representing them as a single binary feature in the state space of the agent. In total there are seven different classes of drinks (water, juice, soda, milk, beer, wine, liquor), requiring to represent them as a one-hot vector of length seven in the state space of the agent (b in Eq. 2). With this state space in place, we conducted experiments where a participant had to find the cheapest item of one, three and five drink categories in each trial (i.e. "find the cheapest water, soda, juice and beer on the shelf"). The policies learned on this data show that the agent can identify the drinks of interest in all three cases (see Figure 1, b), Figure 10, a-b), and video 3:03 - 3:40 min). However, the setting is prone to flicker. This can be explained by the fact that items of different classes of drinks have a high visual similarity (see juice- and milk cartons in Figure 10, a-b) which causes participants to confuse items and to accidentally check the prices of distractors. Since the quality of labeling policies depends on the compliance of participant behavior with the specified task, this can be seen as a drawback of our approach which we discuss further in the limitations section.

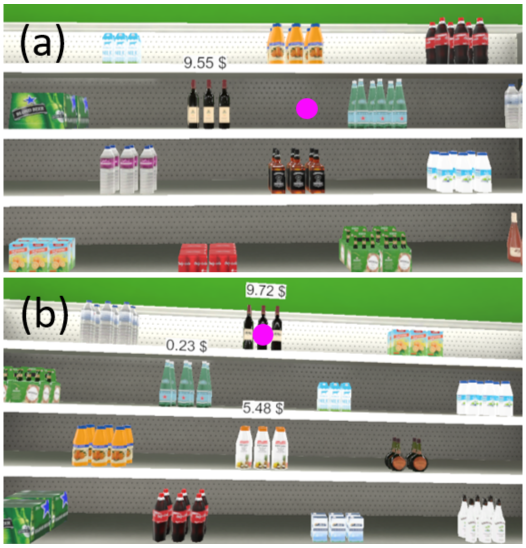

Figure 10. Comparison of task-aware policies: Behavior of two different policies trained and tested on the Supermarket scenario, with different number of target categories each. *Setting:* (a) policy trained on data where the user was instructed to look for wine. (b) policy trained on trials where users are looking for wine, water, and juice. *Results:* (a) The policy correctly displays only the label of a single item of interest (pink dot is user's gaze). (b) Here, the policy displays the labels of multiple items of the target categories, while hiding those of other drinks.

Apartment Scenario: Higher Visual Fidelity

To investigate how well our method works in a realistic looking environment in which objects and labels can be occluded (see Figure 11, a), we ask participants to find the object with the highest number on its label in the rooms of a virtual apartment (bathroom, kitchen, living room, etc.). Objects are items such as a soccer ball, a rubber duck, a toy airplane, etc. Between trials we change the room and the layout of objects. We conduct this experiment to see if the additional randomness in participant's gaze behavior, introduced by salient features as well as object and label occlusions, prevents the agent from learning meaningful policies. Such realistic scenes are challenging to our RL-agent. Policies tend to show the labels of all objects close to participants' point of gaze since occlusions make it difficult for the agent to identify behavioral differences for target and distractor objects (see Figure 11, d). However, even in these environments, learning can converge to policies

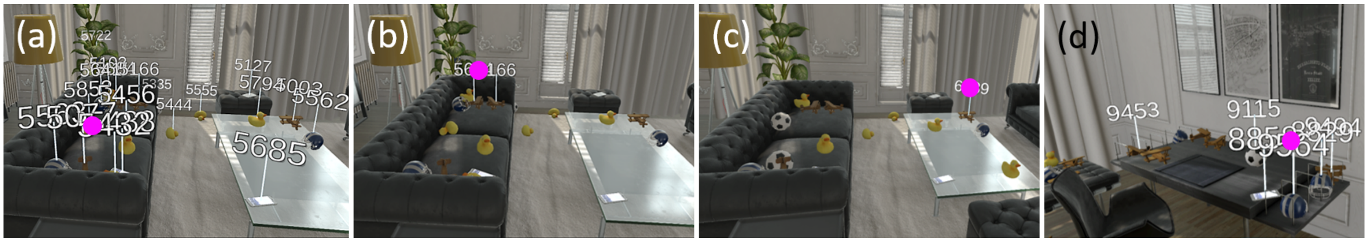

Figure 11. Apartment scenario: (a) a realistic apartment environment with label and object occlusions makes for a difficult visual search task; (b, c) policy that identifies target objects and only shows their labels; (d) policy that shows labels of objects which are close to user's gaze.

that identify target objects and only show the desired labels (see Figure 11, b, c) and video 4:05 - 4:53 min).

LIMITATIONS AND FUTURE WORK

The results of our technical evaluation and user study have shown that our RL-method is capable of learning cooperative labeling policies from gaze behavior that support users in their task while filtering unnecessary information. Nevertheless, this work is not without limitations. In our setting, the agent observes users' behavior and learns cooperative policies based on it. The problem of this approach is that the extent to which a policy can support a task depends on the compliance of users' gaze behavior with this task in training data. If users, during data collection, regularly confused target and distractor objects and checked the labels of distractors the cooperative policy will learn to label objects accordingly. This limitation became apparent in the more realistic use cases where policies could not identify a distinct behavior for certain object types, resulting in flickering labels or cases where labels of all objects around users' gaze were displayed.

A further limitation is the task-dependency of our approach. We have shown that our method can learn effective labeling policies for tasks where the pattern of eye movements around objects are predictive of whether users will look at a task-relevant label. However, during data collection participants always only solved the specified visual search task and did not engage in secondary activities. This will not hold in a real-world setting where multitasking is a common theme. One solution would be to learn specific policies for certain locations (e.g., supermarket) or events (e.g., networking event) where labels have a distinct meaning to users (e.g., price or name tags). Still, the policy will be exposed to non-task-driven behavior. For instance, a user talking to other customers rather than browsing supermarket shelves. To accommodate such cases, one could try to over-sample events of interest (i.e., label fixations) during learning [28].

We have shown that our method can learn task- and preference-sensitive policies from noisy eye tracking data in close-to-realistic tasks (supermarket) and environments (apartment). Potential applications scenarios of our approach are grocery shopping situations, where policies display the price tags for regularly bought items, or cocktail parties, where policies show the name tags of people that users are unfamiliar with. In future work, we would like to investigate how phenomena of user behavior in such real-world applications, like sidetracking, multitasking or changing preferences, affect our method and if it is possible to find suitable extensions to recover from them. Currently, we learn our policies offline. Another interesting direction of future work is to test our method in an online setting to see how quickly personalized cooperative policies can be learned. Therefore, users could solve the visual search task with an arbitrary behavioral policy (e.g., show the labels of all objects). The agent could then learn online and off-policy to only display the labels for objects of interest.

CONCLUSION

In this paper, we demonstrate a RL-method that implicitly learns personalized cooperative labeling policies from gaze behavior that support users in a visual search task while filtering unnecessary information. We cast MR object labeling as a RL control problem using a Semi-Markov Decision Process. In addition, we introduce an RL-environment that simulates the gaze-behavior of a particular user such that our agent can leverage recorded gaze-object interactions to learn cooperative policies. Finally, we propose a reward function that allows the evaluation of the agent's actions only on implicit user interaction (gaze) without needing explicit user-action feedback.

Our evaluation shows that the RL-agent can learn policies which quantitatively perform better in an unseen environment than supervised baselines. Furthermore, we provide empirical evidence that our cooperative polices are helpful in visual search tasks. Our user study results demonstrate that our approach has higher perceived supported than baseline methods while reducing the amount of displayed information by 87% compared to showing all labels at all time. We attribute this to the fact that our method can learn the relevance of an object for users' tasks. This allows our agent to display the label of an object based on spatial and semantic information.

We also demonstrate the applicability of our method in realistic environments and tasks. Applying our method to a supermarket scenario, we were able to learn users' preferred products by observing them browsing a supermarket shelf. Furthermore, we provide proof that our approach can learn meaningful policies even in visually rich high fidelity MR environments with object and label occlusions. We hope our work inspires others to apply Reinforcement Learning to cooperative assistance tasks to achieve personalized user interfaces.

ACKNOWLEDGEMENTS

We thank Chip Connor and Casey Brown for conducting and organizing the two studies as well as Seonwook Park for providing the voice-over for the video. We are also grateful for the valuable feedback from the area chairs and external reviewers.

REFERENCES

[1] Pieter Abbeel, Dmitri Dolgov, Andrew Y Ng, and Sebastian Thrun. 2008. Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE International Conference on Intelligent Robots and Systems 2008. (IROS '08)*. IEEE, 1083–1090. DOI: http://dx.doi.org/10.1109/IROS.2008.4651222

[2] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*. ACM, 1. DOI: http://dx.doi.org/10.1145/1015330.1015430

[3] Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. 2018. Playing hard exploration games by watching YouTube. In *Advances in Neural Information Processing Systems (NIPS '18)*. 2930–2941. https://arxiv.org/abs/1805.11592

[4] Ronald Azuma and Chris Furmanski. 2003. Evaluating label placement for augmented reality view management. In *Proceedings of the 2nd IEEE/ACM international Symposium on Mixed and Augmented Reality (ISMAR '03)*. IEEE, 66.

[5] Nikola Banovic, Tofi Buzali, Fanny Chevalier, Jennifer Mankoff, and Anind K Dey. 2016. Modeling and understanding human routine behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 248–260. DOI: http://dx.doi.org/10.1145/2858036.2858557

[6] Blaine Bell, Steven Feiner, and Tobias Hoellerer. 2001. View Management for Virtual and Augmented Reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. 101–110. DOI: http://dx.doi.org/10.1145/502348.502363

[7] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, 456–464. DOI: http://dx.doi.org/10.1145/3289600.3290999

[8] Adam Coates, Pieter Abbeel, and Andrew Y Ng. 2009. Apprenticeship learning for helicopter control. *Commun. ACM* 52, 7 (2009), 97–105. DOI: http://dx.doi.org/10.1145/1538788.1538812

[9] Ralf Engbert and Reinhold Kliegl. 2003. Microsaccades uncover the orientation of covert attention. *Vision Research* 43, 9 (2003), 1035–1045. DOI: http://dx.doi.org/10.1016/S0042-6989(03)00084-1

[10] Milica Gašić and Steve Young. 2014. Gaussian processes for POMDP-based dialogue manager optimization. *IEEE Transactions on Audio, Speech and Language Processing* 22, 1 (2014), 28–40. DOI: http://dx.doi.org/10.1109/TASL.2013.2282190

[11] Raphael Grasset, Tobias Langlotz, Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. 2012. Image-driven view management for augmented reality browsers. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR '12)*. IEEE, 177–186. DOI: http://dx.doi.org/10.1109/ISMAR.2012.6402555

[12] Zehong Hu, Yitao Liang, Jie Zhang, Zhao Li, and Yang Liu. 2018. Inference aided reinforcement learning for incentive mechanism design in crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS '18)*. 5508–5518. https://arxiv.org/abs/1806.00206

[13] Simon Julier, Marco Lanzagorta, Yohan Baillot, Lawrence Rosenblum, Steven Feiner, Tobias Hollerer, and Sabrina Sestito. 2000. Information filtering for mobile augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR '00)*. IEEE, 3–11. DOI: http://dx.doi.org/10.1109/MCG.2002.1028721

[14] Seong Jae Lee and Zoran Popović. 2010. Learning behavior styles with inverse reinforcement learning. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 122. DOI:http://dx.doi.org/10.1145/1778765.1778859

[15] Yongjoon Lee, Kevin Wampler, Gilbert Bernstein, Jovan Popović, and Zoran Popović. 2010. Motion fields for interactive character locomotion. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 138. DOI: http://dx.doi.org/10.1145/1882261.1866160

[16] Alex Leykin and Mihran Tuceryan. 2004. Automatic determination of text readability over textured backgrounds for augmented reality systems. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality. (ISMAR '04)*. IEEE, 224–230. DOI: http://dx.doi.org/10.1109/ISMAR.2004.22

[17] Elad Liebman, Maytal Saar-Tsechansky, and Peter Stone. 2015. DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*. 591–599. https://arxiv.org/abs/1401.1880

[18] Feng Liu, Ruiming Tang, Xutao Li, Weinan Zhang, Yunming Ye, Haokun Chen, Huifeng Guo, and Yuzhou Zhang. 2018. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *arXiv preprint arXiv:1810.12027* (2018). https://arxiv.org/abs/1810.12027

[19] Wan-Yen Lo and Matthias Zwicker. 2008. Real-time planning for parameterized human motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08)*. 29–38.

[20] Jacob Boesen Madsen, Markus Tatzqern, Claus B Madsen, Dieter Schmalstieg, and Denis Kalkofen. 2016. Temporal coherence strategies for augmented reality labeling. *IEEE transactions on visualization and computer graphics* 22, 4 (2016), 1415–1423. DOI: http://dx.doi.org/10.1109/TVCG.2016.2518318

[21] James McCann and Nancy Pollard. 2007. Responsive characters from motion fragments. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 6. DOI: http://dx.doi.org/10.1145/1276377.1276385

[22] Andrew Y Ng and Stuart J Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. 663–670.

[23] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018a. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Transactions on Graphics* 37, 4 (2018). DOI: http://dx.doi.org/10.1145/3197517.3201311

[24] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. 2018b. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics* 37 (Nov. 2018). DOI: http://dx.doi.org/10.1145/3272127.3275014

[25] D. Purves, D. Fitzpatrick, L.C. Katz, A.S. Lamantia, J.O. McNamara, S.M. Williams, and G.J. Augustine. 2000. *Neuroscience*. Sinauer Associates. https://books.google.ch/books?id=F4pTPwAACAAJ

[26] Aravind Rajeswaran, Kendall Lowrey, Emanuel V. Todorov, and Sham M Kakade. 2017. Towards Generalization and Simplicity in Continuous Control. In *Advances in Neural Information Processing Systems (NIPS '17)*. 6550–6561. https://arxiv.org/abs/1703.02660

[27] Edward Rosten, Gerhard Reitmayr, and Tom Drummond. 2005. Real-time video annotations for augmented reality. In *International Symposium on Visual Computing (ISVC '05)*. Springer, 294–302. DOI: http://dx.doi.org/10.1007/11595755_36

[28] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (2015). https://arxiv.org/abs/1511.05952

[29] Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *arXiv preprint arXiv:1707.00130* (2017). https://arxiv.org/abs/1707.00130

[30] Richard S Sutton and Andrew G Barto. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

[31] Markus Tatzgern, Denis Kalkofen, Raphael Grasset, and Dieter Schmalstieg. 2014. Hedgehog labeling: View management techniques for external labels in 3D space. In *2014 IEEE Virtual Reality (VR)*. IEEE, 27–32. DOI: http://dx.doi.org/10.1109/VR.2014.6802046

[32] Markus Tatzgern, Valeria Orso, Denis Kalkofen, Giulio Jacucci, Luciano Gamberini, and Dieter Schmalstieg. 2016. Adaptive information density for augmented reality displays. In *2016 IEEE Virtual Reality (VR)*. IEEE, 83–92. DOI: http://dx.doi.org/10.1109/VR.2016.7504691

[33] Adrien Treuille, Yongjoon Lee, and Zoran Popović. 2007. Near-optimal character animation with continuous control. *ACM Transactions on Graphics* 26, 3 (2007), 7. DOI:http://dx.doi.org/10.1145/1276377.1276386

[34] J. M. Wolfe. 1994. Guided Search 2 . 0 A revised model of visual search. *Psychnomic Bulletin & Review* 1, 2 (1994), 202–238. DOI: http://dx.doi.org/10.3758/BF03200774

[35] Jeremy M. Wolfe. 2005. Guidance of visual search by preattentive information. *Neurobiology of Attention* (2005), 101–104. DOI: http://dx.doi.org/10.1016/B978-012375731-9/50021-5