EyeFormer: Predicting Personalized Scanpaths with Transformer-Guided Reinforcement Learning

Yue Jiang*
Zixin Guo*
yue.jiang@aalto.fi
zixin.guo@aalto.fi
Aalto University
Finland

Hamed R. Tavakoli
hamed.rezazadegan_tavakoli@nokia.com
Nokia Technologies
Finland

Luis A. Leiva
name.surname@uni.lu
University of Luxembourg
Luxembourg

Antti Oulasvirta
antti.oulasvirta@aalto.fi
Aalto University
Finland

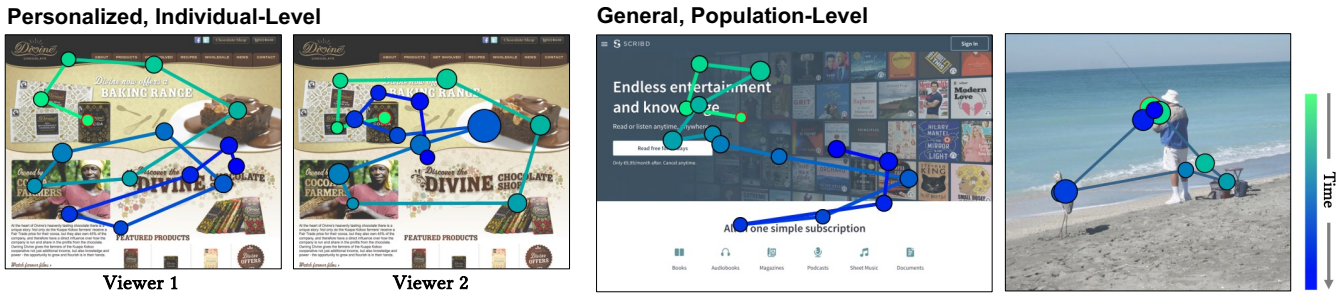

Figure 1: We develop a novel predictive model for scanpaths that represents the first approach to account for individual-level differences and diverse stimuli, from both natural scenes (e.g., landscapes and buildings) and artificial ones (e.g., user interfaces and information graphics). The predictions address a scanpath's spatial and temporal characteristics – that is, a sequence of fixation locations together with the duration of each. The model can generate an "average" scanpath to capture population-level tendencies but also scanpaths personalized for individual viewers from a few scanpath samples, thereby reflecting each viewer's unique preferences and viewing behaviors. These illustrative plots use a color gradient, from green to blue, to denote the temporal progression of each scanpath. Fixation points are denoted by circles, the radii of which represent fixation duration.

ABSTRACT

From a visual-perception perspective, modern graphical user interfaces (GUIs) comprise a complex graphics-rich two-dimensional visuospatial arrangement of text, images, and interactive objects such as buttons and menus. While existing models can accurately predict regions and objects that are likely to attract attention "on average", no scanpath model has been capable of predicting scanpaths for an individual. To close this gap, we introduce EyeFormer, which utilizes a Transformer architecture as a policy network to guide a deep reinforcement learning algorithm that predicts gaze locations. Our model offers the unique capability of producing personalized predictions when given a few user scanpath samples. It can predict full scanpath information, including fixation positions and durations, across individuals and various stimulus types. Additionally, we demonstrate applications in GUI layout optimization driven by our model.

*Both authors contributed equally to this research.

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0628-8/24/10.
https://doi.org/10.1145/3654777.3676436

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in ubiquitous and mobile computing**; • **Computing methodologies** → *Computer vision.*

ACM Reference Format:
Yue Jiang, Zixin Guo, Hamed R. Tavakoli, Luis A. Leiva, and Antti Oulasvirta. 2024. EyeFormer: Predicting Personalized Scanpaths with Transformer-Guided Reinforcement Learning. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3654777.3676436

1 INTRODUCTION

A fundamental goal in the design of graphical user interfaces (GUIs) is to guide users' attention toward discovering relevant information and possibilities for interaction [49]. However, modern GUIs' graphics-rich visuospatial arrangement of text, images, animations, and numerous interactive objects (buttons, menus, etc.) makes it increasingly difficult to predict and direct visual attention for distinct individuals and groups [19, 26–28]. Furthermore, GUI design is not the only factor in eye movements – idiosyncratic features such as expectations and user-specific attention strategies exert an influence. Therefore, predicting how the attention of a given user is going to evolve over time is technically challenging. Breakthroughs in this space would afford the design of personalized visual flows,

reduce clutter, and render user interfaces more engaging, usable, and visually appealing overall [52].

Large individual differences have been reported in viewing patterns [24]. To avoid the fallacy of predicting an "average scanpath" with no correspondence to actual viewers' behavior, models should capture individual-to-individual variability in viewing patterns. Such models would open routes to new applications: they could supply predictions for an audience segment of interest or even for individual viewers. Effective solutions to this challenge would advance applications of human-attention models in visual computing and related domains.

Prior work has focused primarily on **saliency maps**, which represent eye-movement data via density maps for the images [20]. However, as static representations, these overlook temporal information. In contrast, **scanpaths** contain a wealth of information on fixations, retaining details of the order in which objects and regions are attended to, accompanied by the respective duration [7, 24, 41]. Scanpaths are, therefore, first-order models of human vision from which second-order measurements such as saliency maps can be derived, while the converse is not true. In addition, prior research into scanpath modeling has centered predominantly on natural scenes. For making these models more generalizable, unified models that can work with multiple classes of stimuli are crucial. The problem is that visual attention hinges on the stimulus type, so viewing patterns can differ greatly between, for instance, Web sites and mobile GUIs [35]. Any improvement in scanpath-based predictive modeling will immediately carry over to practical applications. For example, the models would help designers to understand visual flows and to adjust their designs such that users are encouraged to view the GUI elements in the desired order [44].

To address this gap, we present *EyeFormer*, a scanpath model for free-viewing tasks. It can accurately predict both population- and individual-level spatiotemporal characteristics of viewing behaviors across multiple stimulus types. We formulated fixations' positioning as a reinforcement learning (RL) problem and used a Transformer architecture as a policy network guiding the selection of each sequence's subsequent fixation. Transformers have proven effective in various tasks, in fields from language to vision [6, 17, 18, 63]. Their capability of modeling long sequences [36] makes them especially suitable for scanpath prediction. EyeFormer's Transformer-guided deep RL approach was designed to address three critical shortcomings of current approaches. Firstly, predicting the order of fixations from saliency maps via their probability distribution [10, 37] is inherently hard because such representations lack temporal information. The second issue stems from post-processing steps implemented to prevent the excessive clustering of fixations that is characteristic of density-based approaches [24] (for instance, inhibition of return [20] is applied to prevent repeated fixation at a previously identified position in a saliency map). Because these steps are not learnable from data [13], one cannot formulate proper loss terms derived from them. Thirdly, though recent advances such as PathGAN [3] have brought progress toward handling fixation duration, accuracy in predicting fixation points remains limited since these techniques often generate points outside the areas of interest.

EyeFormer is the first model to predict full scanpaths at both individual and population level, including fixations with coordinates and duration both. It is unique in its ability to predict an individual's viewing behaviors when given a few sample scanpaths from such a viewer. Moreover, we find that EyeFormer compares favorably to prior scanpath models by the vast majority of metrics for both GUIs and natural scenes in its population-level scanpath prediction. It accurately predicts both spatial order ("where") and temporal (order and duration) characteristics of scanpaths with both of these scene types. Further, we develop an application of personalized scanpath prediction for creating personalized GUI layouts by considering the viewing order and fixation density of GUI elements. In addition, we can generate a single optimized GUI layout that shows minimal variability across individuals, to attract attention to desired elements. We have made our code available at https://github.com/YueJiang-nj/EyeFormer-UIST2024.

In summary, this paper makes the following contributions:

(1) We propose EyeFormer, a deep RL solution incorporating the Transformer architecture that predicts both spatial and temporal characteristics of scanpaths, thus yielding a comprehensive understanding of viewers' viewing behaviors.
(2) It shows how our model generates personalized scanpaths via only a few scanpath samples from the relevant viewer, whereby the model can capture and reflect each user's viewing behaviors and preferences.
(3) We present quantitative and qualitative evaluations demonstrating that the proposed model performs as well as or better than the state-of-the-art models at population-level scanpath prediction for GUIs and natural scenes.
(4) We demonstrate an application of personalized GUI optimization facilitated by personalized scanpath prediction.

2 RELATED WORK

Scanpath models predict sequences of fixations for a given image. This task is more challenging than predicting (dense) saliency maps because the order of the (discrete) fixations must be predicted. All previous research has concentrated on modeling scanpath patterns at population level (i.e., employing an "average-user" model), while none has focused on the prediction of personalized scanpaths. Therefore, we conducted a comprehensive review of the approaches to population-level scanpath prediction as groundwork for extending these techniques to individuals' level by using a novel Transformer-based architecture. Prior work can be classed into three main groups on the basis of how they have attempted to derive sequential information: 1) computing it *post hoc* from densities captured in saliency maps, 2) directly predicting sequences, and 3) formulating this as a sequential control problem via RL. Our evaluation compares our EyeFormer model against several models mentioned next.

2.1 Saliency-Map-Based Scanpath Prediction

Saliency maps, although they do not explicitly contain temporal information, can be used to estimate scanpaths. Itti et al. [20] introduced an Inhibition of Return (IOR) mechanism to this end. It samples a starting fixation and "discourages" future fixations from returning to it, thus producing a richer sequence. While several studies have refined the idea [2, 8, 33, 37, 48, 65, 68, 69], methods of this sort still face challenges in three respects: they (i) neglect some key temporal factors, specifically fixation duration; (ii) lack a

coherent ranking order for the fixations; and (iii) cannot serve in loss terms, since they are non-differentiable.

2.2 Predicting Fixation Sequences

Some attempts to resolve these challenges have entailed sequentially sampling fixations from pre-generated Gaussian distributions and integrating well-designed supervised loss terms. This strategic choice enforces a meaningful order among the fixation points. For instance, IOR-ROI [8, 54], ScanpathNet [10], and Visual ScanPath Transformer [45] predict fixation distributions through a parameterized Gaussian mixture for generating such distributions. GazeFormer [41] incorporates a Transformer-based architecture for goal-oriented viewing tasks, and ScanDMM [53] utilizes a Markov model to represent fixation distributions. These models suffer from accumulation of error [46], whereby errors in previously generated points affect the prediction of the following points. Other models directly predict fixation sequences. Verma and Sen [62] predicted fixations by means of a grid-based representation in which each fixation point is tied to a specific region. PathGAN [3] and ScanGAN [38], in turn, apply a GAN-based architecture to generate fixation sequences; however, GAN-based scanpath models show such limitations as clustering of fixation points toward the image's center and reduced accuracy in predicting fixation points (which sometimes get placed outside the areas of interest). Also noteworthy is NeVA [51], which addresses downstream visual tasks with unseen datasets by relying on existing pre-trained models for the task rather than simulating human scanpaths.

2.3 Reinforcement Learning for Scanpath Prediction

Studies have examined RL's potential in formulating scanpaths as a sequential control problem [5, 42]. For example, Minut and Mahadevan [40] proposed an RL model for visual search tasks wherein an agent learns to focus on relevant areas to locate a target object in a cluttered environment. Ognibene et al. [43] employed RL using an eye-centered potential action map that accumulates possible target locations over fixations, and Yang et al. [72] used inverse RL for predicting the scanpaths involved in a visual search task. In other work, Xu et al. [71] applied deep RL specifically to predict head-movement-related scanpaths for panoramic videos.

Recent work by Chen et al. [7] discretized fixation positions by representing each image as a grid and predicting the cell in the grid corresponding to a particular fixation. Inspired by policy gradient methods applied in discrete token generation for visual captioning [47], Chen et al. adopted their policy gradient to optimize for non-differentiable metrics in their discrete tokenizing. The position discretization offers the advantage of optimizing a finite set of discrete actions rather than a continuous and hence infinite space. However, the artificiality of discretization brings coarser fixation representation, which leads in turn to loss of precision/information. The challenge with continuous control is that a continuous range of control encompasses an infinitude of feasible actions [57]. Against this backdrop, we construct our fixation prediction as a continuous value generation task and turn to parametric functions for Gaussian distributions over actions, optimized by means of our designed rewards.

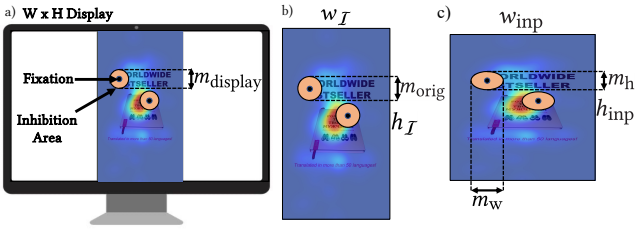

Figure 2: The mechanism of adapting a display's inhibition-of-return area to compute the salient-value reward involves modifying the radius of the inhibition area, which is determined by the disparity between the size of the saliency map and the size of the image as displayed. a) The diameter of the display's inhibition areas m_{display} is commensurate with a human's visual angle. b) Hence, we compute m_{orig}, the diameter for the corresponding inhibition areas for the input image with size $w_{\mathcal{I}} \times h_{\mathcal{I}}$. c) The image needs resizing to the dimensions $w_{\text{inp}} \times h_{\text{inp}}$, to correspond to the input image size required by the model. The inhibition areas are rendered, accordingly, as ellipses with radii m_w and m_h.

3 METHOD

EyeFormer performs scanpath predictions as sequential generation of fixation points, taking preceding fixations and the scene into account together as its **state**. The main challenges we tackle are related to 1) generating both spatial and temporal information on fixation points with parametric distributions, 2) optimizing a scanpath with non-differentiable objectives, and 3) capturing individual-specific viewing differences to predict personalized scanpaths. We propose a Transformer-guided RL approach (depicted in Figure 3) for three key reasons: 1) The Transformer architecture lets us capture long-range sequential dependencies from previous fixations with Gaussian distributions [61]. 2) We find RL preferable to directly optimizing the loss since some loss terms' non-differentiable nature precludes direct optimization. The RL framework enables optimizing scanpaths with non-differentiable reward functions [56], such as terms for computing salient values with IOR. 3) Transformer-only models suffer from the above-mentioned error-accumulation issues: prediction errors from previously generated points propagate to subsequent predictions. During training, the model is fed the previous ground truth rather than its own predictions, so a mismatch arises during inference when it must rely on those potentially inaccurate predictions. We deal with this issue by using RL to train the model to generate sequences as it will during inference, optimizing its policies through continuous feedback and adjustments in line with cumulative **rewards** over time.

3.1 Problem Formulation

Given an image $\mathcal{I}$, our technique generates a scanpath of length T: a sequence of ordered fixation points $p_{1:T} = (p_1, \ldots, p_T)$ capturing the spatial and temporal information of the human gaze. Each fixation $p_i = (x_i, y_i, t_i)$ is a three-dimensional vector representing the normalized point coordinates $x_i \in [0, 1]$ and $y_i \in [0, 1]$ alongside the third dimension, fixation duration expressed as $t_i \in (0, +\infty)$.

3.2 Environment, State, and Action

Our predictive model acts as an **agent** that interacts with the **environment**, where the latter produces the state of both input image $\mathcal{I}$ and previous fixation points. The θ parameters dictate the **policy**, π_θ, whereby the model generates an **action** as a prediction for the next fixation point $\hat{p}_i$, sampled from the distribution produced by the policy model. This process is formulated as $\pi_\theta(\hat{p}_i|\hat{p}_{1:i-1}, \mathcal{I})$.

3.3 Reward Function

After each action, the agent receives a salient-value reward r_{sal} that expresses that action's contribution to the full scanpath. Once the entire scanpath is generated, the agent is exposed to a reward r_{dtwd}, calculated by means of the Dynamic Time Warping with duration (DTWD) metric discussed below. The training's objective is to minimize the negative expected reward, which is equivalent to maximizing a positive reward:

$$\mathcal{L}(\theta) = -\mathbb{E}_{\hat{\mathbf{p}}\sim\pi_\theta} r(\hat{\mathbf{p}}), \tag{1}$$

where $\hat{\mathbf{p}} = (\hat{p}_1, ..., \hat{p}_T)$ and where $\hat{p}_i$ represents the ith fixation sampled from the model-generated distribution. The reward function combines the DTWD metric r_{dtwd}, assessing the similarity between the predicted and the ground truth (GT) scanpath, with the summed salient-value reward for each fixation point along the scanpath generated r_{sal}, thus:

$$r(\hat{\mathbf{p}}) = -r_{\text{dtwd}}(\hat{\mathbf{p}}) + \sum_{i=1}^{T} r_{\text{sal}}(\hat{p}_i). \tag{2}$$

3.3.1 Dynamic Time Warping with duration. Dynamic Time Warping (DTW) is widely used for comparing two sequences that may differ in length [4, 50]. It is useful for scanpaths because it finds an optimal alignment between the two scanpaths (ground truth and predicted ones) and computes the distance without missing any critical features. We implement DTW extended for duration to consider both spatial and temporal characteristics of scanpaths. Specifically, EyeFormer spatially aligns scanpaths by using fixation positions, then computes DTWD values as 3D vectors (x, y, t) for fixations' position and duration. By incorporating DTWD computations over the full scanpath into the reward function, we sought to generate scanpaths closer to the ground truth trajectories and duration.

3.3.2 Salient values. EyeFormer applies rewards for salient values to encourage fixations in salient areas. To avoid repeatedly fixating on the same location in the image, we implement an IOR mechanism to model the relevant tendency of the human visual system. We establish inhibition areas (as regions of the saliency map) for all the previously predicted fixation points. If the new predicted point falls within these areas, it does not elicit any additional reward; the reward corresponds to the salient value on the saliency map in all other cases. Importantly, predicted scanpaths can still return to an already-visited element, just as real-world ones may, since DTWD encourages fixations to revisit the most salient areas and our chosen IOR radius (explained next) is not so large as to preclude revisiting an element. We denote the display's dimensions as $W \times H$. It sets the diameter of that display's inhibition areas m_{display} to be consistent with a human's visual angle, the angle an object subtends at the eye (see Figure 2a). Our choices were informed by the diameter-setting suggested by Klein et al. [32] and further analyzed by Emami et al. [13]. Finally, we compute m_{orig}, the diameter for the corresponding inhibition areas for the input image with size $w_\mathcal{I} \times h_\mathcal{I}$ (see Figure 2b):

$$m_{\text{orig}} = \frac{m_{\text{display}}}{\min\left(W/w_\mathcal{I}, H/h_\mathcal{I}\right)} \tag{3}$$

. Note that preparing the image for processing necessitates resizing it to $w_{\text{inp}} \times h_{\text{inp}}$, which corresponds to the size that the policy model requires for splitting the input image into patches. Using a square input image simplifies computations of this type [12]; accordingly, we resize the inhibition areas from circles to ellipses, while accounting for potential distortions (see Figure 2c). Any point (x, y) in the image that satisfies the following condition gets inhibited (resulting in a salient-value reward of 0) and is omitted from the saliency map:

$$\frac{(x-x_i)^2}{m_{\text{w}}^2} + \frac{(y-y_i)^2}{m_{\text{h}}^2} \leq 1, \text{ where } m_{\text{w}} = \frac{w_{\text{inp}}}{w_\mathcal{I}} m_{\text{orig}},\ m_{\text{h}} = \frac{h_{\text{inp}}}{h_\mathcal{I}} m_{\text{orig}}, \tag{4}$$

where (x_i, y_i) is the coordinates for the ith predicted fixation point. Hence, salient-value reward r_{sal} at step i is defined as the salient value of predicted fixation $\hat{p}_i$ on the saliency map with IOR applied.

3.4 Policy Network

A two-stage approach characterizes the policy network for scanpath prediction. The visual representation of any image $\mathcal{I}$ is learned through the image encoder ($\mathcal{E}$), after which a scanpath gets generated by means of the fixation decoder ($\mathcal{D}$). For population-level scanpath prediction, the visual embedding $\mathcal{E}(\mathcal{I})$ is taken as input to the decoder. For individual-level prediction, feeding the decoder this input along with a viewer embedding, e_u, allows the model to generate personalized scanpaths for separate viewers.

3.4.1 Vision encoder. We use a Vision Transformer (ViT) [6] network as the vision encoder. Specifically, the image is resized to a resolution of $w_{\text{inp}} \times h_{\text{inp}}$ and split into $n_\mathcal{I}$ non-overlapping patches for the vision encoder. Splitting functions mainly to speed up the model's inference, capture local information, and obtain global information from relationships between patches. Next, a linear projection, a convolution layer, is applied to convert these patches into single-dimension embeddings $e_\mathcal{I}^k \in \mathbb{R}^{d_\mathcal{I}}$ thus:

$$e_{\tilde{\mathcal{I}}} = [e_\mathcal{I}^{CLS}, e_\mathcal{I}^1, \ldots, e_\mathcal{I}^{n_\mathcal{I}}] + e_{pos}, \tag{5}$$

where $e_\mathcal{I}^{CLS}$ is a learnable vector for the image context, $[e_\mathcal{I}^{CLS}, e_\mathcal{I}^1, \ldots, e_\mathcal{I}^{n_\mathcal{I}}]$ is a matrix from concatenating the vectors $e_\mathcal{I}^{CLS}, e_\mathcal{I}^1, \ldots, e_\mathcal{I}^{n_\mathcal{I}}$, and $e_{pos} \in \mathbb{R}^{d_\mathcal{I} \times (n_\mathcal{I}+1)}$ is the positional matrix reflecting the position context of the image patches. Finally, we apply a vision encoder $\mathcal{E}(\cdot)$ based on a 12-layer version of the ViT model [12]. By employing per-patch convolution and using a Transformer to combine patch embeddings, the ViT model expresses the relationship for each patch and lets us derive the final image embedding, denoted as $\mathcal{E}(e_{\tilde{\mathcal{I}}})$. We consider an alternative vision encoder, using a residual neural network (ResNet), also; the supplementary materials include comparison between it and the mechanism ultimately chosen.

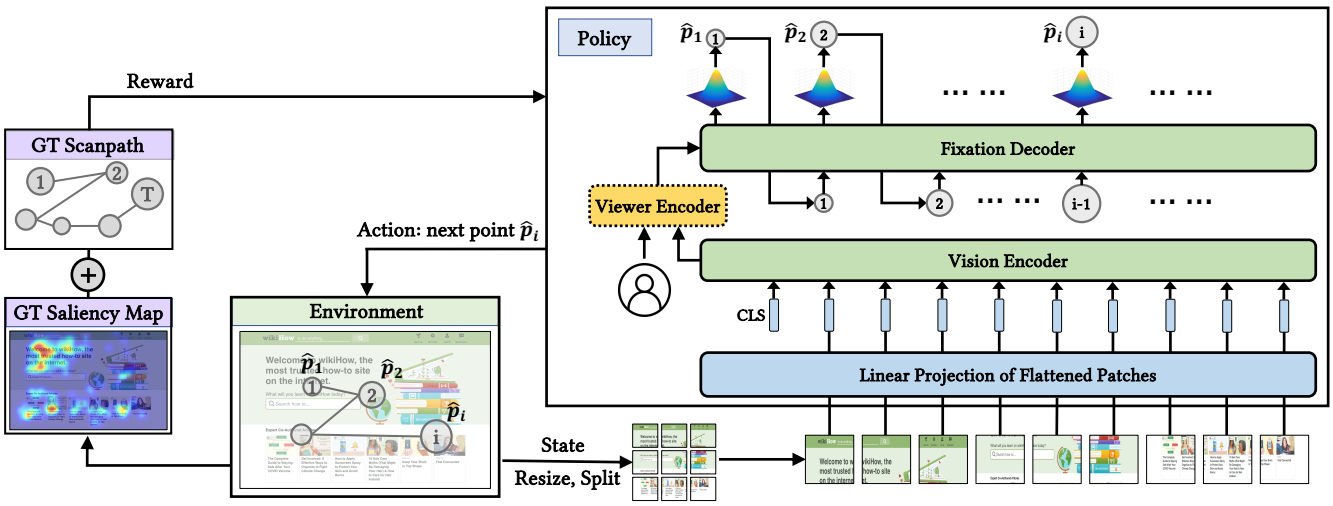

Figure 3: The overview of our Transformer-guided RL framework for scanpath prediction, comprising several components: the environment, which produces the state of the input image and previous fixation points; the Transformer model, which furnishes the policy; the policy-generated action, predicting the next point in the scanpath; and the reward function (obtained from evaluating the action against ground truth), through which the policy gets updated. Within the Transformer policy model, the image patches, resized and split from the input image, are fed to the vision encoder for generation of the image embedding; the viewer encoder generates the viewer embedding, only used in individual-level prediction to distinguish between viewers; and the fixation decoder takes the image and viewer embeddings along with previously generated fixations to generate the next points along the scanpath in sequence. During training, the model begins by sampling the next point from the distribution generated by the policy in light of the current state. This sampled point is used to update the state of the environment, and incorporating the reward indicated via ground truth serves to update the Transformer policy model. During testing, we use the policy model to generate the scanpaths directly.

3.4.2 Fixation decoder. To generate fixation points, $\hat{p}_i$, we use a multi-layer Transformer decoder, $\mathcal{D}$. It takes the image embedding $\mathcal{E}(e_{\tilde{I}})$ alongside the previously generated points denoted by $\hat{p}_{1:i-1}$ as input to generate $\mathcal{D}(\mathcal{E}(e_{\tilde{I}}), \hat{p}_{1:i-1})$. This allows previous fixation points to influence points further along the scanpath. We set the first fixation to be at the center of the screen since the conditions behind most eye-tracking datasets involve asking participants to look at the center of the display before images get presented [24, 70]. For the given state (the previously predicted fixation points and the input image), the action (the next prediction for a fixation point) is

$$\pi_\theta(\hat{p}_{1:T}|I) = \pi_\theta(\hat{p}_1|I) \prod_{i=2}^{T} \pi_\theta(\hat{p}_i|\hat{p}_{1:i-1}, I). \tag{6}$$

The policy π_θ is represented as a Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i)$. Alternatively, it could be represented as a mixed Gaussian distribution $\sum_{k=1}^{K} \lambda_{ik}\mathcal{N}(\mu_{ik}, \Sigma_{ik})$ with a total of K Gaussian components, where λ_{ik} denotes the weight of the kth Gaussian component, and Σ_{ik} denotes the covariance matrix specific to the component at step i. These variables for determining the distribution are sequentially generated by the decoder. We present more implementation details and a comparison between using a Gaussian and a mixed Gaussian distribution in supplementary materials.

3.5 Predicting Personalized Scanpaths

To distinguish between individual viewers, we select a two-layer Transformer architecture as the viewer encoder $\mathcal{E}_u$. This facilitates prediction of individual-level scanpaths considerably. The training process trains the model from the training users in the dataset. The viewer encoder is taught to allow each viewer's distinct viewing behaviors to be encoded in a separate embedding space. In the test process, when given a new viewer, the model updates the viewer encoder with a few scanpaths from that viewer by backpropagating from the scanpath samples. Once the model has updated the viewer encoder, it can predict scanpaths specific to this unique viewer, thereby customizing its predictions for this individual's viewing behaviors (note that this encoder is not applied for population-level predictions).

Specifically, the image representation, $\mathcal{E}(e_{\tilde{I}})$, serves as the input query, while viewer embedding e_u serves as the key and value in the cross-attention mechanism within the viewer encoder. The viewer embedding is a learnable matrix. For generation of fixations, the output of this encoder, $\mathcal{E}_u(e_{\tilde{I}}, e_u)$, is directed to the fixation decoder.

3.6 Policy Gradient

To compute the gradient of the objective function $\nabla_\theta \mathcal{L}(\theta)$, our method employs the REINFORCE algorithm [47, 67], which offers

a Monte Carlo variant of a policy-optimization technique commonly used in RL settings [56]. Under this algorithm, the agent accumulates samples from episodes by executing its current policy and utilizes those samples to update the policy's parameters iteratively. The REINFORCE algorithm aims to maximize the cumulative expected reward across sequential actions by approximating the gradient of the expected reward for the current policy parameters. By adjusting these parameters iteratively in accordance with the gradient estimate, the algorithm attempts to enhance the policy's performance over time. This algorithm is rooted in the insight that one can obtain the expected gradient of a non-differentiable reward function as follows:

$$\nabla_\theta \mathcal{L}(\theta) = -\mathbb{E}_{\hat{\mathbf{p}}\sim\pi_\theta}\left[r(\hat{\mathbf{p}})\nabla_\theta \log \pi_\theta(\hat{\mathbf{p}}|\mathcal{I})\right]. \tag{7}$$

To approximate the expected gradient, we use a single Monte-Carlo sample $\hat{\mathbf{p}} = (\hat{p}_1, ..., \hat{p}_T)$ from the policy π_θ for each training example in the minibatch:

$$\nabla_\theta \mathcal{L}(\theta) \approx -r(\hat{\mathbf{p}})\nabla_\theta \log \pi_\theta(\hat{\mathbf{p}}|\mathcal{I}). \tag{8}$$

REINFORCE with a baseline. Our technique uses a baseline b to assess the environment's expected reward without any actions, thus generalizing the policy gradient obtained from REINFORCE. Applying this algorithm with a baseline allows us to estimate the advantage yielded by an action – i.e., the difference between the actual reward obtained and that expected from the baseline environment. By subtracting the baseline value, we reduce the variance of the gradient estimation, thereby arriving at a stabler optimization process. The gradient of the loss with respect to the θ policy parameters is then obtained as

$$\nabla_\theta \mathcal{L}(\theta) = -\mathbb{E}_{\hat{\mathbf{p}}\sim\pi_\theta}\left[(r(\hat{\mathbf{p}}) - b)\nabla_\theta \log \pi_\theta(\hat{\mathbf{p}}|\mathcal{I})\right]. \tag{9}$$

For each step in the training, our technique approximates the expected gradient with a single sample $\hat{\mathbf{p}} \sim \pi_\theta$:

$$\nabla_\theta \mathcal{L}(\theta) \approx -(r(\hat{\mathbf{p}}) - b)\nabla_\theta \log \pi_\theta(\hat{\mathbf{p}}|\mathcal{I}). \tag{10}$$

In the discrete space, Rennie et al.'s conceptualization [47] serves as a foundational framework, wherein b is estimated by means of the reward obtained from the policy's greedy search. For operating in a continuous space, however, our approach diverges from theirs: at each step, the operation of our policy necessitates computation of b, defined as the reward associated with the mean of multiple samples drawn from the policy – in essence, the mean of the distribution generated by the policy. Consequently, the expected gradient is calculated as

$$\nabla_\theta \mathcal{L}(\theta) \approx -(r(\hat{\mathbf{p}}) - r(sg[\boldsymbol{\mu}]))\nabla_\theta \log \pi_\theta(\hat{\mathbf{p}}|\mathcal{I}), \tag{11}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_T)$ and $sg[\cdot]$ constitute a stop-gradient operator having partial derivatives of 0.

4 EXPERIMENTS

Our experiments attest to the new model's unique capability of producing personalized predictions when given a few user scanpath samples. Below, we present the experiments in connection with a comprehensive evaluation of EyeFormer against multiple recently developed models and across two very different classes of stimuli: GUIs and natural scenes. Our examination covered a large number of baselines and of evaluation metrics suited to scanpath models.

4.1 Datasets

Both datasets in our experiments – the GUI-oriented UEyes [24, 25] and OSIE [70], from natural scenes – feature multiple scanpaths for each image, from numerous viewers. The two datasets were collected by eye trackers that output fixation points and their durations, rather than saccades.

4.1.1 GUIs and information graphics. The **UEyes** dataset provided us with eye-tracking data (up to 7 s) from 62 participants who viewed 1,980 images drawn from four common types of GUI and information graphics (posters, desktop GUIs, mobile GUIs, and webpages). Collecting the data with an eye tracker in a laboratory setting guaranteed precise fixation coordinates in the X–Y plane, and the coordinate values were subject to participant-specific calibration accounting for relevant human factors such as eye–display distance [35]. We used the same training/test image split as Jiang et al. [24]: 1,872 images in the training set and 108 in the test set, with the four GUI types distributed evenly within each set. In addition, we established a training/test split for individual-level prediction, randomly assigning 53 viewers to the training set (85%) and the remaining nine to the test set (15%). Our model was trained on the data collected from when the training viewers looked at the GUIs shown in the training images. Most scanpaths in UEyes have roughly 15 fixations (the average number of fixations per image is 15.3). Further details of the dataset and implementation can be found in the supplementary materials.

4.1.2 Natural scenes. The **OSIE** dataset, from free viewing of natural scenes, comprises 700 images with associated eye-movement data from three seconds of viewing by 15 participants. With OSIE, which has been widely used in previous research [7, 54], we applied the same split used in prior work (80% training, 10% validation, and 10% testing data). We did not use datasets such as SALICON's [21], since they take mouse movements as a proxy for eye movements, whereas EyeFormer is designed for replicating actual scanpaths recorded by eye trackers.

4.2 Metrics

We assessed performance via metrics commonly employed for scanpath evaluation [1, 15]. All experiments used coordinates $x \in [0, 1]$ and $y \in [0, 1]$, normalized for image size (px/px, dimensionless), and fixation duration $t \in [0, +\infty)$ in milliseconds.

4.2.1 Dynamic Time Warping (DTW). DTW serves as a standard metric for similarity between two temporal sequences even when they differ in length [4, 50]. It identifies the optimal match and calculates the distance between two scanpaths in a manner that preserves essential features.

4.2.2 Time Delay Embedding (TDE). By focusing on assessment of similarities at sub-scanpath level [59, 64], TDE offers evaluation more nuanced than DTW's, which attends only to overall comparison of entire scanpaths.

4.2.3 Eyenalysis. Finding the closest mapping between fixation points on the two scanpaths, Eyenalysis takes each fixation point along the first scanpath and identifies the spatially closest fixation point on the second, and *vice versa* [39]. It then measures the average

Model	DTW ↓	TDE ↓	Eyenalysis ↓	DTWD ↓	MultiMatch ↑					
					Shape	Direction	Length	Position	Duration	Mean
Personalized to Other Viewers	4.152 ± 1.161	0.123 ± 0.030	**0.036 ± 0.017**	5.070 ± 1.088	**0.943**	0.733	0.935	0.821	**0.731**	0.833
Personalized to Target Viewer	**4.058 ± 1.135**	**0.121 ± 0.029**	**0.036 ± 0.017**	**4.996 ± 1.078**	**0.943**	**0.737**	**0.936**	**0.824**	**0.731**	**0.834**

Table 1: We compare the model personalized to the target test viewer against the model personalized to other test viewers to quantify the effectiveness in capturing the characteristics of individual viewers on the UEyes dataset.

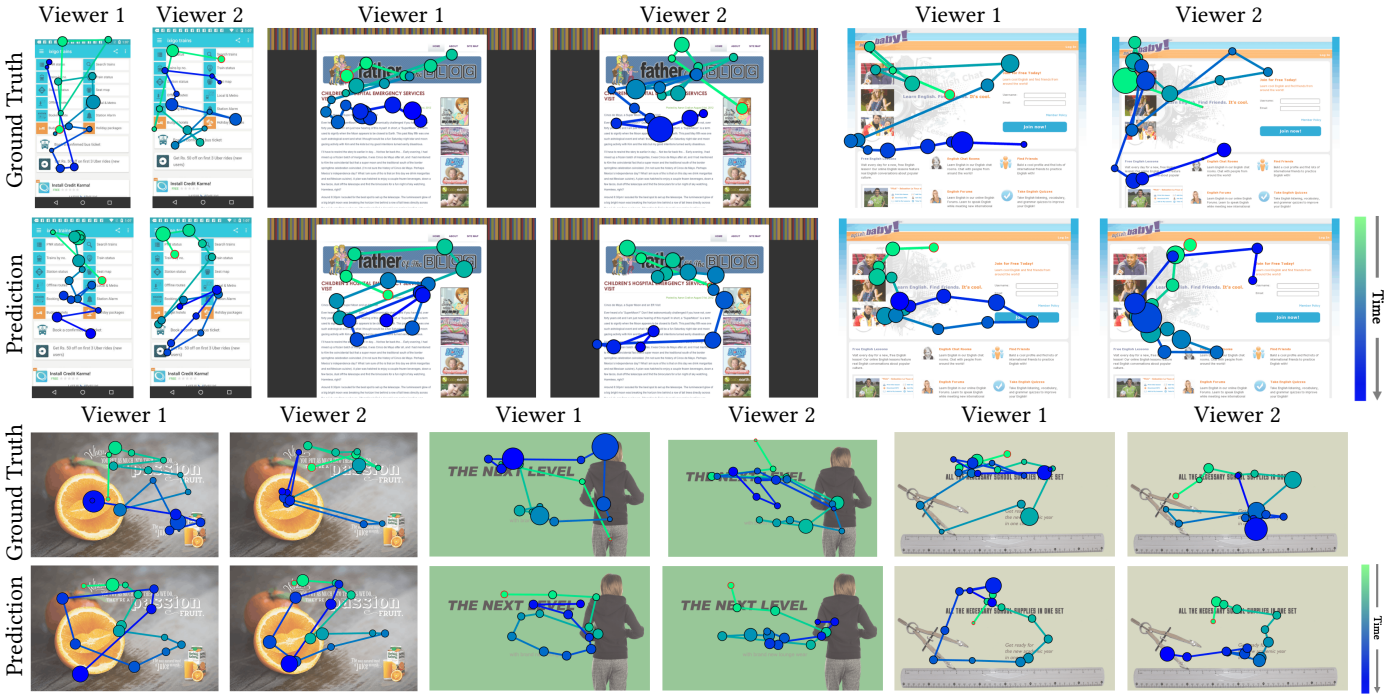

Figure 4: Scanpaths personalized for two viewers, illustrating our model's ability to generate these by means of only a few scanpath samples from each viewer (note that "Viewer 1" and "Viewer 2" are generic terms; the viewers are not the same across all examples). More examples are presented in the supplementary materials.

distances for all the closest fixation pairs, thereby emphasizing evaluation of individual fixations instead of the sequences.

4.2.4 Dynamic Time Warping with duration (DTWD). Our extension of DTW to capture duration empowered considering fixations' position and duration both. We align two scanpaths on the basis of their optimal match of (x, y) coordinates and calculate the cumulative distance by computing, for each pair of aligned points, the distance between the two three-dimensional vectors (x, y, t) representing the spatiotemporal information.

4.2.5 MultiMatch. With MultiMatch metric [11], five variants facilitate assessing important aspects of fixations along scanpaths: shape, direction, length, position, and duration. While DTWD evaluates spatial and temporal characteristics, MultiMatch excels at capturing additional features such as shape, direction, and length and gives an overall evaluation based on all these features.

5 RESULTS

The results demonstrate that our model 1) predicts individual-level scanpaths when given a few viewing samples from the user; 2) compares favorably with other models for population-level scanpath prediction; and 3) predicts both spatial and temporal characteristics of scanpaths with stimuli that include GUI images, information graphics, and natural scenes.

5.1 Individual-Level Scanpath Prediction

Prior research has not addressed the challenge of predicting personalized individual-level scanpaths, partly because full re-training for each new viewer, with more data, is impractical. Our model achieves a workable balance by generating scanpaths tailored to each person's viewing behaviors and idiosyncrasies while still permitting a single model's application for all viewers, without the burden of re-training. We verified our model's ability to generate personalized scanpaths by proceeding from a few scanpath samples from the individual, thus confirming that the model can effectively

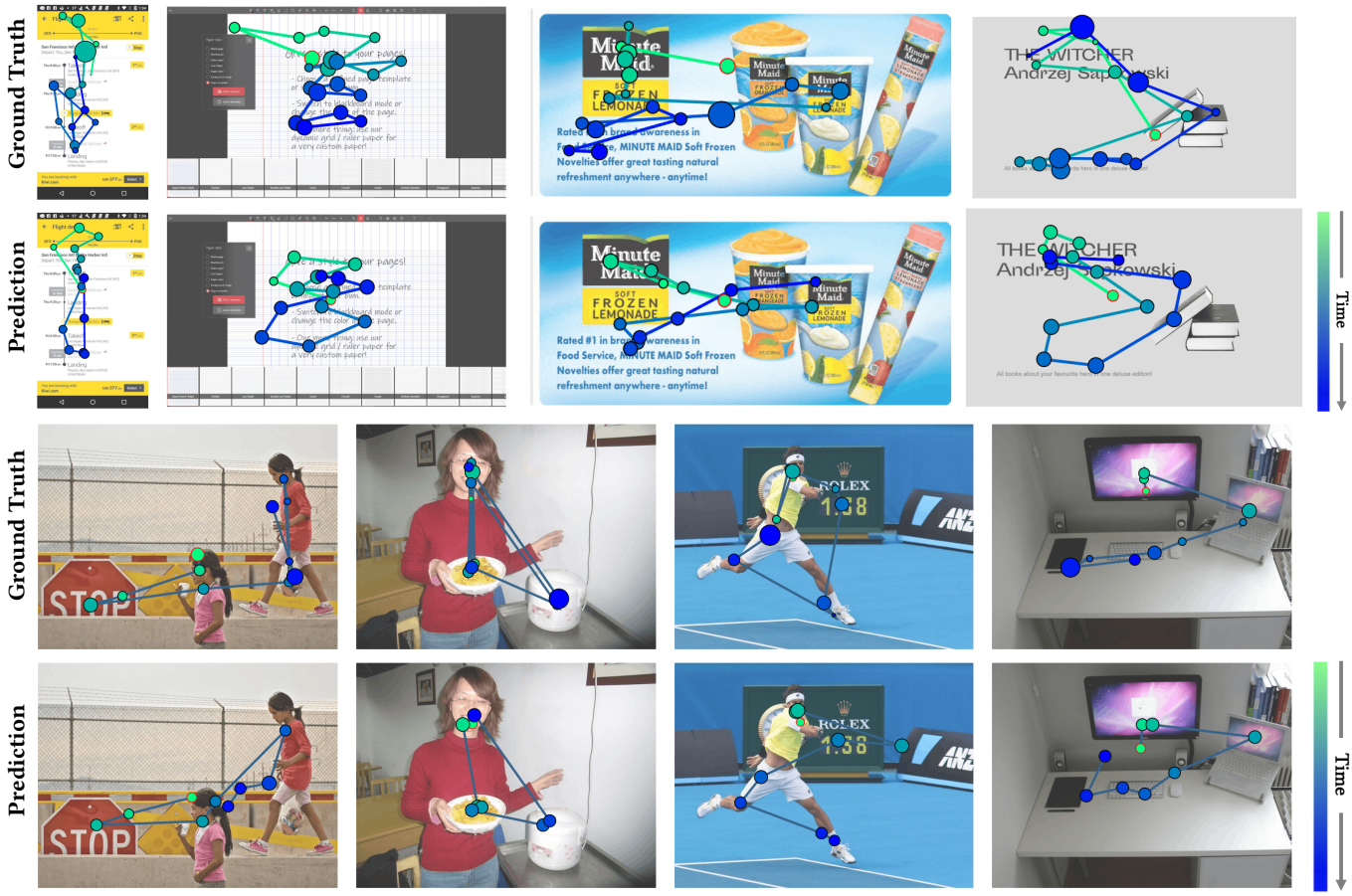

Figure 5: Our population-level scanpath prediction proves to be close to the ground truth with regard to fixation positions, ordering, and duration. The supplementary materials present further examples.

capture each viewer's viewing preferences/behaviors and reflect them in its output.

When encountering a new viewer with a few samples available, the model updates the viewer embedding with n_{path} scanpaths obtained from that viewer (in our experiments, $n_{\text{path}} = 50$). Fine-tuning the model involves backpropagating from the scanpath samples so that it can predict scanpaths specific to this unique individual's viewing behaviors.

Since no established baseline method at present can function as a point of comparison for this personalization approach, we compared the model's tailoring for the target test viewer with its tailoring for other test viewers to quantify its effectiveness in capturing the characteristics of individual viewers. The results (shown in Table 1) show that the errors in the former setting are smaller than those of personalization for other test viewers. We conclude, then, that the personalized model can better address individual-specific characteristics. Illustrative examples presented in Figure 4 capture the nature of the individual-level scanpath prediction qualitatively; in addition, the supplementary materials provide more results and explain the relationship between sample quantity and performance.

5.2 Population-Level Scanpath Prediction

To assess how well our model predicts the spatiotemporal information of scanpaths, we compared its performance with preexisting scanpath models'. We evaluated the model with both GUIs and natural scenes to check whether it can be generalized to different types of images. For GUIs, we compared to Itti–Koch [20], DeepGaze III [33], DeepGaze++ [24], SaltiNet [2], UMSS [65], PathGAN [3], PathGAN++ [24], ScanGAN [38], ScanDMM [53], and the model of Chen et al. [7]. Comparisons for natural scenes judged EyeFormer against models focused on such scenes: Itti–Koch [20], SGC [55], the model by Wang et al. [64], Le Meur et al.'s model [34], STAR-FC [68], SaltiNet [2], PathGAN [3], IOR-ROI [54], GazeFormer [41], and Chen et al. [7]. While one of the baseline models, GazeFormer, is a Transformer-based method designed for visual search, directly comparing it with other methods is not possible because GazeFormer requires a pre-specified target, which free-viewing tasks do not provide. Therefore, we adapted GazeFormer to free-viewing tasks by providing a blank target as input. For a fair comparison, we trained all these models with the same dataset split. We fed the models every individual scanpath from all the

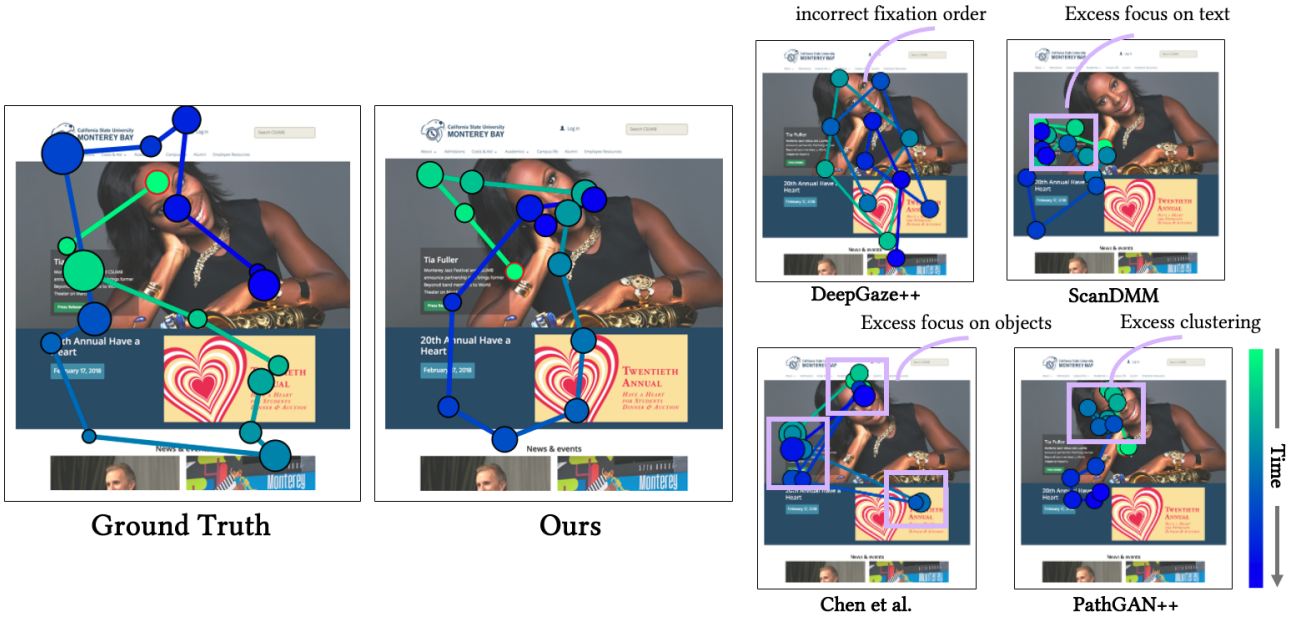

Figure 6: Annotated comparison between different models. The illustration alongside our model's result, presenting the baseline performance, is marked up to highlight particular limitations.

Model	DTW ↓	TDE ↓	Eyenalysis ↓	DTWD ↓	MultiMatch ↑					
					Shape	Direction	Length	Position	Duration	Mean
GUIs and Information Graphics (UEyes dataset)										
Itti–Koch	6.249 ± 0.986	0.150 ± 0.025	0.047 ± 0.028	–	0.861	0.721	0.819	0.746	–	–
DeepGaze III Pretrained	7.906 ± 2.466	0.274 ± 0.061	0.124 ± 0.076	–	0.937	0.567	0.886	0.746	–	–
DeepGaze++	5.454 ± 1.078	0.149 ± 0.032	0.047 ± 0.026	–	0.907	0.708	0.906	0.773	–	–
PathGAN Pretrained	4.719 ± 1.387	0.192 ± 0.049	0.072 ± 0.037	–	0.940	0.579	0.892	0.800	–	–
PathGAN	4.754 ± 1.185	0.147 ± 0.048	0.048 ± 0.025	–	0.943	0.716	0.935	0.797	–	–
PathGAN++	4.559 ± 1.182	0.146 ± 0.037	0.044 ± 0.022	–	0.943	0.706	0.933	0.807	–	–
PathGAN w/ D	5.192 ± 1.422	0.204 ± 0.045	0.092 ± 0.038	6.431 ± 1.644	0.939	0.556	0.891	0.779	0.667	0.766
PathGAN++ w/ D	5.443 ± 1.466	0.202 ± 0.044	0.096 ± 0.043	6.667 ± 1.659	0.939	0.560	0.896	0.765	0.657	0.763
SaltiNet	7.042 ± 1.622	0.187 ± 0.057	0.063 ± 0.054	8.241 ± 1.487	0.907	0.715	0.897	0.691	0.579	0.758
UMSS	5.051 ± 1.592	0.155 ± 0.048	0.050 ± 0.026	6.495 ± 1.468	0.934	0.713	0.921	0.779	0.579	0.785
ScanGAN	4.815 ± 1.238	0.136 ± 0.034	0.040 ± 0.022	–	0.931	0.734	0.929	0.796	–	–
ScanDMM	5.085 ± 1.317	0.138 ± 0.037	0.043 ± 0.027	–	0.931	0.729	0.928	0.784	–	–
Chen et al.	4.335 ± 1.299	**0.118 ± 0.034**	0.037 ± 0.019	5.533 ± 1.250	0.939	0.725	0.926	0.823	0.720	0.827
GazeFormer	4.189 ± 1.204	0.141 ± 0.038	0.046 ± 0.023	5.262 ± 1.041	**0.947**	0.734	0.931	**0.825**	0.730	0.833
EyeFormer	**4.069 ± 1.089**	0.122 ± 0.029	**0.036 ± 0.018**	**5.043 ± 1.052**	0.942	**0.748**	**0.940**	**0.825**	**0.750**	**0.841**
Natural Scenes (OSIE dataset)										
Itti–Koch	3.180 ± 0.756	0.176 ± 0.039	0.061 ± 0.027	–	0.859	0.653	0.811	0.748	–	–
SGC	2.992 ± 1.067	0.194 ± 0.071	0.073 ± 0.046	–	0.922	0.652	0.890	0.768	–	–
Wang et al.	3.798 ± 1.128	0.227 ± 0.073	0.096 ± 0.060	–	0.886	0.641	0.841	0.700	–	–
Le Meur et al.	3.027 ± 0.797	0.160 ± 0.476	0.057 ± 0.028	–	0.892	0.653	0.865	0.770	–	–
STAR-FC	3.375 ± 1.300	0.228 ± 0.091	0.090 ± 0.067	–	0.936	0.662	0.920	0.734	–	–
SaltiNet	3.439 ± 0.861	0.191 ± 0.052	0.065 ± 0.032	3.860 ± 0.814	0.895	0.641	0.872	0.719	0.573	0.740
PathGAN	5.300 ± 1.197	0.323 ± 0.073	0.142 ± 0.085	5.454 ± 1.167	0.935	0.577	0.924	0.608	0.679	0.745
IOR-ROI	2.495 ± 0.809	0.160 ± 0.055	0.060 ± 0.039	2.955 ± 0.768	0.914	**0.704**	0.889	0.812	0.629	0.790
Chen et al.	**2.183 ± 0.949**	0.125 ± 0.056	0.045 ± 0.028	2.636 ± 0.865	**0.944**	0.653	0.924	0.847	0.689	0.811
EyeFormer	2.193 ± 0.831	**0.115 ± 0.042**	**0.044 ± 0.026**	**2.562 ± 0.756**	**0.944**	0.679	**0.932**	**0.850**	**0.706**	**0.822**

Table 2: Quantitative scanpath evaluation, with the *Mean* ± *SD* reported for each metric, attesting that our model outperforms the baseline models by most metrics with both GUIs and natural scenes ("Pretrained" denotes testing via the pre-trained model, while other models were trained with the same dataset; boldface highlights the best result column-wise; arrows indicate the importance relation's direction (e.g., ↑ means "higher is better"); and dashes ("–") indicate methods unable to predict duration).

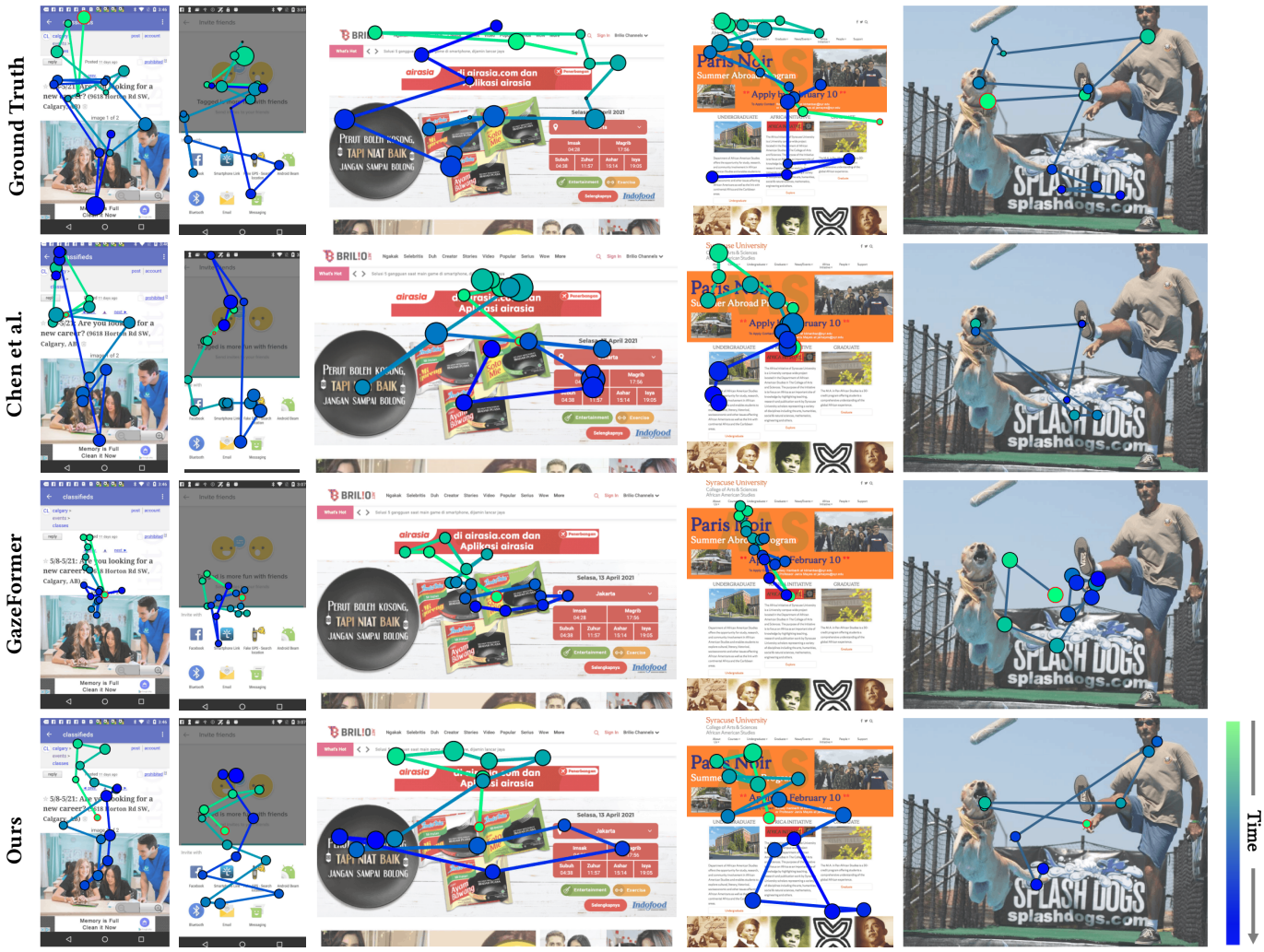

Figure 7: Qualitative comparison. The best baseline methods, from Chen et al. [7] and GazeFormer [41], are shown here. Comparison to other scanpath models is provided in the supplementary materials.

viewers for each training image, helping the models learn the underlying scanpath distribution. Note that we did not combine the two datasets: all methods were trained on each dataset separately. Training and analysis too remained separate.

5.2.1 Quantitative evaluation. To account for variations in image sizes and minimize discrepancy-related errors, we normalized the fixation points' coordinates to the $[0, 1]$ range. Specifically for training on natural scenes, we used the ResNet instead of the ViT mechanism as the vision encoder, for better comparison to other baseline models since prior work with training on the OSIE data [70] used a ResNet model as the encoder. Table 2 presents a comprehensive comparison covering all the metrics. Our model proved at least as good as the baseline models by most metrics, for GUIs and natural scenes both. The results indicate that it simulates scanpath trajectories more realistically. Of the models tested, only PathGAN, PathGAN++, SaltiNet, UMSS, IOR-ROI, GazeFormer, and Chen et al.'s technique can predict temporal information. Chen et al., which is one of the best baseline models, predicts positions and duration separately; however, fixation positions and duration are highly correlated. GazeFormer, by relying on a Transformer model to generate an entire scanpath in a single step, overlooks the local dependencies and correlations between adjacent points. The fact that our model excels by the DTWD and MultiMatch `Duration` metrics attests to its capacity to yield more accurate results and also handle prediction of temporal information.

5.2.2 Qualitative evaluation. Qualitative comparisons revealed that the predictions made by our model lie closer to the ground truth than those of the other models. Figure 5 presents population-level prediction results showcasing the performance of EyeFormer. Figure 6 and Figure 7 provide comparison between our model and the baseline ones (more results are available in the supplementary

materials). While PathGAN++ and ScanGAN generate realistic trajectories very well (thanks to their discriminative component), the points they predict often fall outside the salient areas and tend to lie in clusters. In contrast, DeepGaze++ performs well in locating fixation points, by applying post-processing to density maps. Nevertheless, it generates fixations in incorrect order; on account of the non-differentiable nature of the post-processing, the order is not optimized. The Itti–Koch, SaltiNet, and UMSS techniques generate scanpaths from saliency maps, encouraging fixations in salient areas, but they too fail to optimize for correct fixation order. Chen et al.'s technique tends to generate several clusters of closely grouped points since they improved the prediction of fixations without addressing the need to spread consecutive points out more. In additional analysis, we computed the clustering-tendency error via the Laminarity metric [1]. The Laminarity value of our model is 73.137, and that of Chen et al.'s is 178.072 (lower values are better). . Its higher score indicates that the model of Chen et al. produces fixation-clustering in locations where ground-truth fixations do not cluster. Finally, ScanDMM focuses relatively strongly on text elements. Our model assigns fixations to salient areas and attends to the points' order with greater precision. It accomplishes this by using the salient-value reward (r_{sal}) to emphasize points that lie within areas of interest and by employing the DTWD reward (r_{dtwd}) to encourage more accurate trajectories.

5.3 Ablation Study

Table 3 presents the results from an ablation study we performed on utilizing RL to produce both population- and individual-level scanpaths. The results reveal that a Transformer-only model does not yield satisfactory results and that incorporating RL greatly enhances the prediction of fixations and their duration. With its population-level prediction, our RL model brings an improvement of 14.7% and 26.5%, respectively, by the TDE and the Eyenalysis metric. This too is evidence that using RL increases the model's capacity to generate realistic fixations for scanpaths. As for fixation duration, applying RL has a positive influence on prediction accuracy, demonstrated by the 4.8% improvement shown by the DTWD metric. Similar effects are visible with the individual-specific predictions connected with training users (i.e., the trained model's prediction of scanpaths for GUI images when given the IDs of particular training users). Additionally, the results highlight that the absence of either each type of reward or of inhibition of return leads to a decline in overall accuracy. Results from further ablation studies are included in the supplementary materials.

6 APPLICATION FOR PERSONALIZED VISUAL FLOWS

EyeFormer enables handy prediction of individual-level scanpaths. Demonstrating this capability in practice, we applied it to the problem of **personalizing visual flows**. In model-assisted flow design, the designer identifies GUI elements intended to receive more attention than others [16]. Our goal was to support this by controlling the flow of attention to selected elements. While prior work has demonstrated model-assisted personalization of graphical layouts [60], its focus has been solely on visual-search time, not visual flow. In our scenario, the designer supplies a GUI layout and specifies the desired visiting order for three or more elements that should be fixated upon first (the most important ones). After this, our system outputs both population- and individual-optimized layouts. Generation of the individual-specific layouts is based on the personalized scanpath prediction results. Specifically, given a viewer with n_{path} scanpath samples (in our experiments, $n_{\text{path}} = 50$), EyeFormer generates corresponding layouts by proceeding from the predicted scanpaths at individual level for this particular viewer.

6.1 Formulation of Optimization Problem

We expressed this application as a constraint optimization problem [22, 23, 29–31] that requires ascertaining positions and sizes of elements for a GUI based on the predicted personalized scanpaths. To address this problem, we built on an integer-programming-based layout optimizer [9] that optimizes GUI layouts by considering their elements' packing, alignment, and preferred positioning. Additionally, we introduced a constraint requiring adherence to the designer-specified fixation order, along with an objective score derived from EyeFormer's predictions.

6.1.1 Fixation order constraint. We denote the order of the three most important elements, elem_1, elem_2, and elem_3, which should be fixated upon earliest, as $[\text{elem}_1, \text{elem}_2, \text{elem}_3]$. Extending the list permits handling more elements, in a similar manner. Firstly, for the predicted scanpath $[\hat{p}_1, \hat{p}_2, ..., \hat{p}_T]$, the procedure identifies the GUI element receiving fixations, per fixation point, denoted as $[\text{elem}_{\hat{p}_1}, \text{elem}_{\hat{p}_2}, ..., \text{elem}_{\hat{p}_T}]$. Secondly, $[\text{elem}_1, \text{elem}_2, \text{elem}_3]$ is restricted to being a subset from the beginning of the deduplicated sequence $[\text{elem}_{\hat{p}_1}, \text{elem}_{\hat{p}_2}, ..., \text{elem}_{\hat{p}_T}]$; that is, the sequence $[\text{elem}_{\hat{p}_1}, \text{elem}_{\hat{p}_2}, ..., \text{elem}_{\hat{p}_T}]$ begins with repeated occurrences of elem_1, followed by elem_2 and subsequently elem_3. This constraint guarantees that the required fixation order specified by the designer is honored.

6.1.2 Objective term for fixation duration. To define optimality further, we applied a fixation-duration objective term for GUI layouts that satisfy the required-order constraint. Where the fixations corresponding to the sub-sequence of repeated occurrences of elem_1 followed by elem_2 and then by elem_3, described above, are denoted as $[\hat{p}_1, \hat{p}_2, ..., \hat{p}_M]$ (with $\hat{p}_M$ being the final fixation before attention moves to other elements), the objective is to select the layout whose fixation durations for these elements sum to the maximal value: $\sum_{m=1}^{M} t_{\hat{p}_m}$.

6.2 Results

Figure 8 shows two resulting designs (more examples are provided in the supplementary materials). Given an original GUI design and an annotated sequence of the (three) most important GUI elements, we generate both 1) the population-optimized layout and 2) a layout personalized for each viewer. The population-optimized layout relies on the population-level scanpath prediction, which serves as the best compromise across viewers, while the viewer-specific layouts are based on personalized scanpath prediction such that each viewer follows the desired order and devotes maximal time to the elements deemed important. Testing for 62 individual viewers yielded the following results for the designs shown in the figure: For "Design 1", 56 viewers would follow the desired viewing order

Model	DTW ↓	TDE ↓	Eyenalysis ↓	DTWD ↓	MultiMatch ↑					
					Shape	Direction	Length	Position	Duration	Mean
Population-Level Scanpath Prediction										
Ours w/o RL	4.304 ± 1.309	0.143 ± 0.041	0.049 ± 0.024	5.299 ± 1.235	**0.946**	0.709	0.925	0.820	0.736	0.827
Ours w/o r_{sal}	4.099 ± 1.192	0.137 ± 0.036	0.045 ± 0.023	**4.981 ± 1.131**	**0.946**	0.713	0.928	0.825	**0.752**	0.833
Ours w/o r_{dtwd}	5.277 ± 1.009	0.139 ± 0.025	**0.036 ± 0.018**	6.733 ± 1.038	0.913	0.736	0.907	0.789	0.673	0.804
Ours w/o IOR	4.485 ± 1.353	0.177 ± 0.047	0.074 ± 0.034	5.327 ± 1.261	0.945	0.697	0.909	0.816	0.738	0.821
Ours	**4.069 ± 1.089**	**0.122 ± 0.029**	**0.036 ± 0.018**	5.043 ± 1.052	0.942	**0.748**	**0.940**	**0.825**	**0.750**	**0.841**
Individual-Level Scanpath Prediction for Training Viewers										
Ours w/o RL	4.362 ± 1.294	0.137 ± 0.039	0.046 ± 0.027	5.517 ± 1.200	**0.945**	0.722	0.932	0.815	0.719	0.827
Ours	**4.164 ± 1.039**	**0.120 ± 0.027**	**0.035 ± 0.016**	**5.166 ± 0.998**	0.937	**0.755**	**0.936**	**0.824**	**0.738**	**0.838**

Table 3: Results from an ablation study examining RL's impact on population-level and also individual-specific predictions for training of viewers on the UEyes dataset (the results highlight the importance of DTWD and salient-value reward terms, alongside the use of inhibition of return).

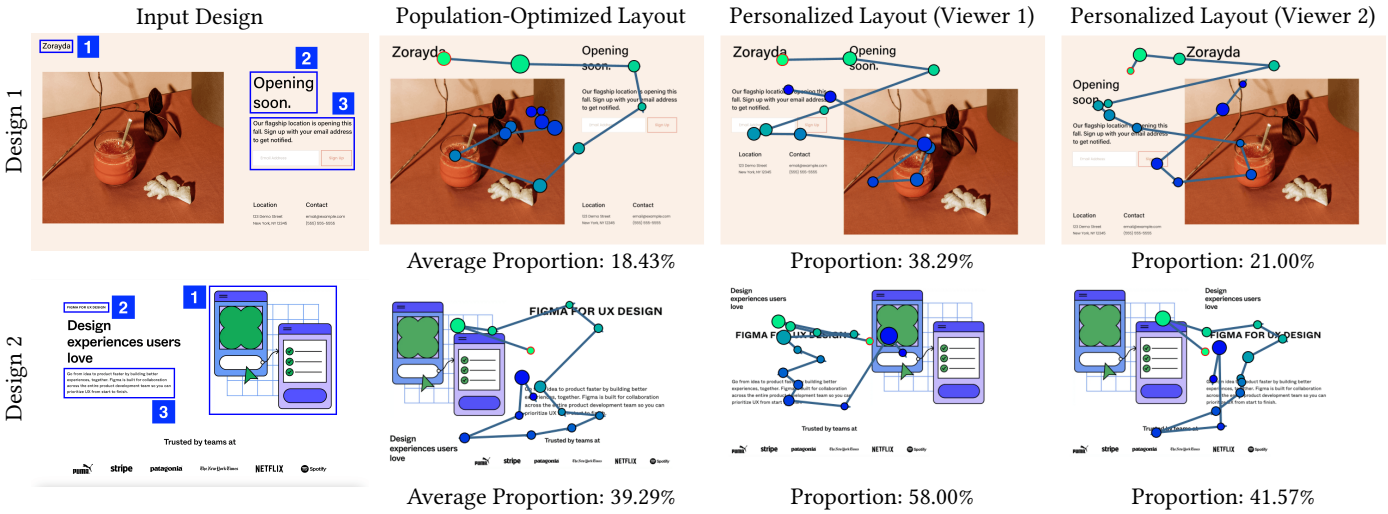

Figure 8: Given the input GUI design with the order of the three most important elements as identified by the designer, we generate both the population-optimized layout and the personalized layout for each individual viewer. The figure shows the average percentage of the total fixation duration for those elements over a seven-second viewing period for the population-optimized layout across all test viewers and the corresponding proportion for each personalized layout shown, also with seven seconds of viewing. Personalized layouts can attract more of the respective viewer's attention to the target elements than the population-optimized layout does.

for the designer-selected elements with the population-optimized layout, devoting 1.29 seconds of the seven-second viewing period to them, on average. Shown the corresponding personalized layout, all viewers would follow the desired order, with an average total duration of 1.86 seconds (44.19% more than with population-level optimizing). Given "Design 2", 46 viewers shown the population-optimized layout would follow the fixation order desired, with an average duration sum of 2.75 s. With the personalized layout for Design 2, 61 viewers would do so, and the average total duration is 3.19 seconds, a sum 16% greater than that from the population-level layout. The results attest that personalized layouts can draw more of the viewer's attention to the target elements than a population-optimized layout does.

7 DISCUSSION AND FUTURE WORK

EyeFormer is able to cover both spatial and temporal characteristics of scanpaths across various stimulus types and factor in individual-specific viewing behaviors, which are vital for understanding visual attention. It opens the door to automated personalization of visual flows, which enables GUI software to respond better to each user's behaviors and expectations. Personalized prediction is critical for practical developments. There is rather extensive variability in scanpaths across individuals; in fact, averaged scanpath prediction may not be very meaningful – after all, it might be unlikely to match any actual user. From inputting example scanpaths of a single user, we have demonstrated that personalized layouts can be generated for that user. Greater accessibility, through GUIs optimized for people with viewing difficulties, is one of many possible application domains. Further research could also use subjective comparison

studies to see whether users prefer GUIs personalized in accordance with scanpath predictions over the original interface.

7.1 Understanding Viewers

Future work could use viewer clustering to enhance the interpretability of extensive sets of scanpaths for designers. Clustering enabled by applying, for example, K-means to the viewer embeddings in our model could help reveal how viewers of various kinds interact with visual content, thereby aiding designers in cultivating aggregate-level insight beyond individual paths, for a broader perspective. Further research could also yield better tools for visualizing and comprehending diverse viewer behaviors.

7.2 Practical Applications of Personalization

By controlling the visual flow over GUIs, designers can encourage users to focus on the most important parts of the interface. This improves usability and aids in reaching specific design goals, such as effective market funneling. Personalized visual flows can support optimal ad placement and related design such that key messages catch the attention of users and drive them toward such desired actions as clicking or buying. Prior work on visual-saliency analysis has highlighted that better visual flow can enhance users' engagement and guide behaviors [14, 58, 66].

The ability to predict individuals' gaze patterns could also support creating adaptive GUIs that respond dynamically to user interactions and preferences. Moreover, associated research addressing the correlation between design trends and user-interaction behaviors could prove fruitful; for instance, being able to fine-tune scanpath prediction in light of current user data could address the fact that individuals' interactions with GUIs evolve over time.

The potential advantages extend beyond GUIs. Education tools could benefit from adjusting visual content in line with the gaze patterns of each user, thus facilitating students' improved comprehension of complex concepts. Similarly, training modules that adapt to users' learning progress "on the fly" and focus on areas ripe for improvement might promote more efficient learning. Also, predicting users' likely points of focus in augmented- and virtual-reality settings could encourage more immersive experiences through dynamic adjustment of visual content that helps users locate objects easily.

7.3 Ethics Concerns

A practical and ethics-related challenge remains, however, in how to collect eye-tracking data from individuals. We foresee two main options: 1) using Web cameras or other commodity devices, with the user's permission, and 2) inferring patterns via proxy signals such as mouse movements. Since people with privacy concerns may be reluctant to share their gaze data, the applications developed – such as GUI layouts personalized on the basis of the user's scanpaths – should be able to run locally; in the ideal case, sensitive gaze information should not be transmitted over the Internet. Another possibility is to compute "sufficient statistic" measurements[1] and send these to a server that generates personalized GUI layouts. These approaches would help maintain user privacy while still offering the benefits of personalized scanpaths.

[1]See https://en.wikipedia.org/wiki/Sufficient_statistic.

7.4 Limitations

At present, the model is limited to fixed-length scanpaths, since we considered a limited time window of free-viewing behaviors (based on the seven-second maximum span in the UEyes dataset [24], which permitted better comparison with earlier work). However, it should be possible to output variable-length scanpaths by predicting the final state. In addition, our discussion concentrated on predicting fixation sequences. We acknowledge that viewing behaviors are far more complex, encompassing many other eye dynamics (blinks, vestibulo-ocular reflexes, post-saccadic oscillations, etc.), which future studies could explore. Follow-up research could also investigate ways of reducing the number of scanpaths needed per viewer (from the current 50). Finally, the state-of-the-art scanpath-related metrics are designed primarily for natural scenes, so they may not fully capture the characteristics of scanpaths in GUI settings. Refining the metrics employed should afford deeper understanding of how models such as ours perform and thus enhance the development of more effective methods.

8 OUR CONCLUSION

EyeFormer is a Transformer-guided RL model, which predicts both population-level and individuals' scanpaths well, using the Transformer architecture as the policy model offers a novel representation for accurately capturing variability in scanning patterns across stimuli and individuals. While the Transformer-guided design effectively captures long-range sequential dependencies on the basis of previous fixations, combining it with RL enhances the generation of fixation sequences through optimization that employs non-differentiable objectives, such as maximizing the salient values of fixations. In addition to performing better than (or at least on par with) state-of-the-art models in the realm of population-level prediction, EyeFormer offers the first accurate modeling of individual-to-individual variability in scanpaths, from only a few user samples. Its application for GUIs optimized in keeping with the personalized scanpath-prediction results marks another contribution offering a way forward.

ACKNOWLEDGMENTS

This work was supported by Aalto University's Department of Information and Communications Engineering, the Research Council of Finland (flagship program: Finnish Center for Artificial Intelligence, FCAI, grants 328400, 345604, 341763; Subjective Functions, grant 357578), the Academy of Finland in project 345791, the Meta Research PhD Fellowship, the Horizon 2020 FET program of the European Union (grant CHIST-ERA-20-BCI-001), and the European Innovation Council Pathfinder program (SYMBIOTIK project, grant 101071147).

REFERENCES

[1] Nicola C Anderson, Fraser Anderson, Alan Kingstone, and Walter F Bischof. 2015. A comparison of scanpath comparison methods. *Behavior research methods* 47, 4 (2015), 1377–1392.

[2] Marc Assens, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O'Connor. 2017. SaltiNet: Scan-Path Prediction on 360 Degree Images Using Saliency Volumes. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2331–2338. https://doi.org/10.1109/ICCVW.2017.275

[3] Marc Assens, Xavier Giro i Nieto, Kevin McGuinness, and Noel E. O'Connor. 2018. PathGAN: Visual Scanpath Prediction with Generative Adversarial Networks.

ECCV Workshop on Egocentric Perception, Interaction and Computing (EPIC).

[4] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*, Vol. 10. Seattle, WA, USA:, 359–370.

[5] Kevin Brohan, Kevin Gurney, and Piotr Dudek. 2010. Using reinforcement learning to guide the development of self-organised feature maps for visual orienting. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)|Lect. Notes Comput. Sci.*, Vol. 6353. Springer Nature, United States, 180–189. https://doi.org/10.1007/978-3-642-15822-3_23 20th International Conference on Artificial Neural Networks, ICANN 2010 ; Conference date: 01-07-2010.

[6] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. 2021. When Vision Transformers Outperform ResNets without Pretraining or Strong Data Augmentations. *CoRR* abs/2106.01548 (2021). arXiv:2106.01548 https://arxiv.org/abs/2106.01548

[7] Xianyu Chen, Ming Jiang, and Qi Zhao. 2021. Predicting human scanpaths in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10876–10885.

[8] Zhenzhong Chen and Wanjie Sun. 2018. Scanpath Prediction for Visual Attention Using IOR-ROI LSTM *(IJCAI'18)*. AAAI Press, 642–648.

[9] Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. Grids: Interactive layout design with integer programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[10] Ryan Anthony Jalova de Belen, Tomasz Bednarz, and Arcot Sowmya. 2022. ScanpathNet: A Recurrent Mixture Density Network for Scanpath Prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 5006–5016.

[11] Richard Dewhurst, Marcus Nyström, Halszka Jarodzka, Tom Foulsham, Roger Johansson, and Kenneth Holmqvist. 2012. It depends on how you look at it: Scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach. *Behavior research methods* 44 (2012), 1079–1100.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR* abs/2010.11929 (2020). arXiv:2010.11929 https://arxiv.org/abs/2010.11929

[13] Parvin Emami, Yue Jiang, Zixin Guo, and Luis A. Leiva. 2024. Impact of Design Decisions in Scanpath Modeling. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*.

[14] Parvin Emami, Yue Jiang, Zixin Guo, and Luis A. Leiva. 2024. Impact of Design Decisions in Scanpath Modeling. *Proc. ACM Hum.-Comput. Interact.* 8, ETRA, Article 228 (may 2024), 16 pages. https://doi.org/10.1145/3655602

[15] Ramin Fahimi and Neil DB Bruce. 2021. On metrics for measuring scanpath similarity. *Behavior Research Methods* 53, 2 (2021), 609–628.

[16] Camilo Fosco, Vincent Casser, Amish Kumar Bedi, Peter O'Donovan, Aaron Hertzmann, and Zoya Bylinskii. 2020. Predicting visual importance across graphic design types. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST)*. 249–260.

[17] Zixin Guo, Tzu-Jui Wang, and Jorma Laaksonen. 2022. CLIP4IDC: CLIP for Image Difference Captioning. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*. 33–42.

[18] Zixin Guo, Tzu-Jui Julius Wang, Selen Pehlivan, Abduljalil Radman, and Jorma Laaksonen. 2023. PiTL: Cross-modal Retrieval with Weakly-supervised Vision-language Pre-training via Prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2261–2265.

[19] Lena Hegemann, Yue Jiang, Joon Gi Shin, Yi-Chi Liao, Markku Laine, and Antti Oulasvirta. 2023. Computational Assistance for User Interface Design: Smarter Generation and Evaluation of Design Ideas. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–5.

[20] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* 20, 11 (1998), 1254–1259.

[21] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. 2015. SALICON: Saliency in Context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1072–1080. https://doi.org/10.1109/CVPR.2015.7298710

[22] Yue Jiang. 2024. Computational Representations for Graphical User Interfaces. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*.

[23] Yue Jiang, Ruofei Du, Christof Lutteroth, and Wolfgang Stuerzlinger. 2019. ORC Layout: Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 413, 12 pages. https://doi.org/10.1145/3290605.3300643

[24] Yue Jiang, Luis A. Leiva, Paul R. B. Houssel, Hamed R. Tavakoli, Julia Kylmälä, and Antti Oulasvirta. 2023. UEyes: Understanding Visual Saliency across User Interface Types. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*.

[25] Yue Jiang, Luis A Leiva, Hamed Rezazadegan Tavakoli, Paul RB Houssel, Julia Kylmälä, and Antti Oulasvirta. 2023. UEyes: An Eye-Tracking Dataset across User Interface Types. In *Workshop Paper at the 2023 CHI Conference on Human Factors in Computing Systems*.

[26] Yue Jiang, Yuwen Lu, Clara Kliman-Silver, Christof Lutteroth, Toby Jia-Jun Li, Jeffrey Nichols, and Wolfgang Stuerzlinger. 2024. Computational Methodologies for Understanding, Automating, and Evaluating User Interfaces. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*.

[27] Yue Jiang, Yuwen Lu, Christof Lutteroth, Toby Jia-Jun Li, Jeffrey Nichols, and Wolfgang Stuerzlinger. 2023. The Future of Computational Approaches for Understanding and Adapting User Interfaces. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 367, 5 pages. https://doi.org/10.1145/3544549.3573805

[28] Yue Jiang, Yuwen Lu, Jeffrey Nichols, Wolfgang Stuerzlinger, Chun Yu, Christof Lutteroth, Yang Li, Ranjitha Kumar, and Toby Jia-Jun Li. 2022. Computational Approaches for Understanding, Generating, and Adapting User Interfaces. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 74, 6 pages. https://doi.org/10.1145/3491101.3504030

[29] Yue Jiang, Wolfgang Stuerzlinger, and Christof Lutteroth. 2021. ReverseORC: Reverse Engineering of Resizable User Interface Layouts with OR-Constraints. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 316, 18 pages. https://doi.org/10.1145/3411764.3445043

[30] Yue Jiang, Wolfgang Stuerzlinger, Matthias Zwicker, and Christof Lutteroth. 2020. ORCSolver: An Efficient Solver for Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376610

[31] Yue Jiang, Changkong Zhou, Vikas Garg, and Antti Oulasvirta. 2024. Graph4GUI: Graph Neural Networks for Representing Graphical User Interfaces. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

[32] Ronald Klein, Barbara E.K Klein, Kristine E Lee, Karen J Cruickshanks, and Richard J Chappell. 2001. Changes in visual acuity in a population over a 10-year period1 1Each author states that he/she has no proprietary interest in any aspect of this work.: The Beaver Dam eye study. *Ophthalmology* 108, 10 (2001), 1757–1766. https://doi.org/10.1016/S0161-6420(01)00769-2

[33] Matthias Kümmerer, Matthias Bethge, and Thomas SA Wallis. 2022. DeepGaze III: Modeling free-viewing human scanpaths with deep learning. *Journal of Vision* 22, 5 (2022).

[34] Olivier Le Meur and Zhi Liu. 2015. Saccadic model of eye movements for free-viewing condition. *Vision Research* 116 (2015), 152–164. https://doi.org/10.1016/j.visres.2014.12.026 Computational Models of Visual Attention.

[35] Luis A Leiva, Yunfei Xue, Avya Bansal, Hamed R Tavakoli, Tuðçe Köroðlu, Jingzhou Du, Niraj R Dayama, and Antti Oulasvirta. 2020. Understanding visual saliency in mobile user interfaces. In *Proceedings of the International conference on human-computer interaction with mobile devices and services*. 1–12.

[36] Tieyuan Liu, Meng Zhang, Chuangying Zhu, and Liang Chang. 2023. Transformer-based convolutional forgetting knowledge tracking. *Scientific Reports* (2023).

[37] Daniel Martin, Diego Gutierrez, and Belen Masia. 2022. A probabilistic time-evolving approach to scanpath prediction. *arXiv preprint arXiv:2204.09404* (2022).

[38] Daniel Martin, Ana Serrano, Alexander W Bergman, Gordon Wetzstein, and Belen Masia. 2022. Scangan360: A generative model of realistic scanpaths for 360 images. *IEEE Transactions on Visualization and Computer Graphics* 28, 5 (2022), 2003–2013.

[39] S Mathot, F Cristino, ID Gilchrist, and J Theeuwes. 2012. Eyenalysis: A similarity measure for eye movement patterns. *Journal of Eye Movement Research* 5 (2012), 1–15.

[40] Silviu Minut and Sridhar Mahadevan. 2001. A reinforcement learning model of selective visual attention. In *Proceedings of the fifth international conference on Autonomous agents*. 457–464.

[41] Sounak Mondal, Zhibo Yang, Seoyoung Ahn, Dimitris Samaras, Gregory Zelinsky, and Minh Hoai. 2023. Gazeformer: Scalable, effective and fast prediction of goal-directed human attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1441–1450.

[42] Sajad Mousavi, Michael Schukat, Enda Howley, Ali Borji, and Nasser Mozayani. 2017. Learning to predict where to look in interactive environments using deep recurrent q-learning. arXiv:1612.05753 [cs.CV]

[43] Dimitri Ognibene, Christian Balkenius, and Gianluca Baldassarre. 2008. A reinforcement-learning model of top-down attention based on a potential-action map. In *The Challenge of Anticipation: A Unifying Framework for the Analysis and Design of Artificial Cognitive Systems*. Springer, 161–184.

[44] Xufang Pang, Ying Cao, Rynson WH Lau, and Antoni B Chan. 2016. Directing user attention via visual flow on web designs. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–11.

[45] Mengyu Qiu, Quan Rong, Dong Liang, and Huawei Tu. 2023. Visual ScanPath Transformer: Guiding Computers to See the World. In *2023 IEEE International*

Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 223–232.

[46] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* (2015).

[47] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7008–7024.

[48] Hamed Rezazadegan Tavakoli, Esa Rahtu, and Janne Heikkilä. 2013. Stochastic bottom–up fixation prediction and saccade generation. *Image and Vision Computing* 31, 9 (2013), 686–693. https://doi.org/10.1016/j.imavis.2013.06.006

[49] Ruth Rosenholtz, Amal Dorai, and Rosalind Freeman. 2011. Do Predictions of Visual Perception Aid Design? *ACM Trans. Appl. Percept.* 8, 2 (2011).

[50] Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[51] Leo Schwinn, Doina Precup, Björn Eskofier, and Dario Zanca. 2022. Behind the Machine's Gaze: Neural Networks with Biologically-inspired Constraints Exhibit Human-like Visual Attention. *arXiv preprint arXiv:2204.09093* (2022).

[52] Jeremiah D. Still and Christopher M. Masciocchi. 2010. A Saliency Model Predicts Fixations in Web Interfaces. In *Proc. MDDAUI Workshop*.

[53] Xiangjie Sui, Yuming Fang, Hanwei Zhu, Shiqi Wang, and Zhou Wang. 2023. ScanDMM: A Deep Markov Model of Scanpath Prediction for 360deg Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6989–6999.

[54] Wanjie Sun, Zhenzhong Chen, and Feng Wu. 2019. Visual scanpath prediction using IOR-ROI recurrent mixture density network. *IEEE transactions on pattern analysis and machine intelligence* 43, 6 (2019), 2101–2118.

[55] Xiaoshuai Sun, Hongxun Yao, and Rongrong Ji. 2012. What are we looking for: Towards statistical modeling of saccadic eye movements and visual saliency. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 1552–1559. https://doi.org/10.1109/CVPR.2012.6247846

[56] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[57] Yunhao Tang and Shipra Agrawal. 2020. Discretizing continuous action space for on-policy optimization. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 34. 5981–5988.

[58] Nađa Terzimehić, Renate Häuslschmid, Heinrich Hussmann, and m.c. schraefel. 2019. A Review & Analysis of Mindfulness Research in HCI: Framing Current Lines of Research and Future Opportunities. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300687

[59] Sauer Tim, A Yorke James, and Casdagli Martin. 1991. Embedology. *Journal of statistical Physics* 65, 3-4 (1991), 579–616.

[60] Kashyap Todi, Jussi Jokinen, Kris Luyten, and Antti Oulasvirta. 2019. Individualising graphical layouts with predictive visual search models. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 10, 1 (2019), 1–24.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[62] Ashish Verma and Debashis Sen. 2019. HMM-based Convolutional LSTM for Visual Scanpath Prediction. In *2019 27th European Signal Processing Conference (EUSIPCO)*. 1–5. https://doi.org/10.23919/EUSIPCO.2019.8902643

[63] Chenguang Wang, Mu Li, and Alexander J. Smola. 2019. Language Models with Transformers. *CoRR* abs/1904.09408 (2019). arXiv:1904.09408 http://arxiv.org/abs/1904.09408

[64] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao. 2011. Simulating human saccadic scanpaths on natural images. In *CVPR 2011*. 441–448. https://doi.org/10.1109/CVPR.2011.5995423

[65] Yao Wang, Andreas Bulling, et al. 2023. Scanpath prediction on information visualisations. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[66] Yao Wang, Yue Jiang, Zhiming Hu, Constantin Ruhdorfer, Mihai Bâce, and Andreas Bulling. 2024. VisRecall++: Analysing and Predicting Visualisation Recallability from Gaze Behaviour. *Proceedings of the ACM on Human-Computer Interaction* 8, ETRA (2024), 1–18.

[67] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning* (1992), 5–32.

[68] Calden Wloka, Iuliia Kotseruba, and John K. Tsotsos. 2018. Active Fixation Control to Predict Saccade Sequences. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3184–3193. https://doi.org/10.1109/CVPR.2018.00336

[69] Chen Xia, Junwei Han, Fei Qi, and Guangming Shi. 2019. Predicting Human Saccadic Scanpaths Based on Iterative Representation Learning. *IEEE Transactions on Image Processing* 28, 7 (2019), 3502–3515. https://doi.org/10.1109/TIP.2019.2897966

[70] Juan Xu, Ming Jiang, Shuo Wang, Mohan S Kankanhalli, and Qi Zhao. 2014. Predicting human gaze beyond pixels. *Journal of vision* 14, 1 (2014), 28–28.

[71] Mai Xu, Yuhang Song, Jianyi Wang, MingLang Qiao, Liangyu Huo, and Zulin Wang. 2018. Predicting head movement in panoramic video: A deep reinforcement learning approach. *IEEE transactions on pattern analysis and machine intelligence* 41, 11 (2018), 2693–2708.

[72] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. 2020. Predicting Goal-Directed Human Attention Using Inverse Reinforcement Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.