Predicting visual attention using the hidden structure in eye-gaze dynamics

GABOR LENGYEL, Central European University, Hungary
KEVIN CARLBERG, Facebook Reality Labs, United States
MAJED SAMAD, Facebook Reality Labs, United States
TANYA JONKER, Facebook Reality Labs, United States

To enhance human-computer interaction in naturalistic environments, a computing system could benefit from predicting where a user will direct their visual attention, which would allow it to adapt its behaviour accordingly. We investigated whether future visual attention could be predicted from past eye-gaze dynamics in a simulated meeting in virtual reality. From recorded eye movements, we extracted gaze samples across objects and people, which significantly reduced the dimensionality of the input and output space of the model compared to a coordinate-based approach and enabled us to train predictive time-series models on long (16min) videos with low computational costs. Compared to baseline and classical autoregressive models, a recurrent neural network model improved performance in future gaze prediction by 64%. Using a self-supervised approach, these initial results suggest that there is structure in users' gaze dynamics and that predictive models could be used to enable human-centric adaptive interfaces.

CCS Concepts: • **Human-centered computing → User models**; **Virtual reality**.

Additional Key Words and Phrases: visual attention, eye tracking, eye-gaze dynamics, virtual reality, predicting user intent, time-series modelling, recurrent neural network, self-supervised training

ACM Reference Format:
Gabor Lengyel, Kevin Carlberg, Majed Samad, and Tanya Jonker. 2021. Predicting visual attention using the hidden structure in eye-gaze dynamics. In *EMICS '21: ACM CHI '21 Workshop on Eye Movements as an Interface to Cognitive State, May 14, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

People move their eyes across visual scenes to extract information relevant for their goals [11]. As they gather information relevant to their task, they drive future eye movements towards areas of interest. In dynamic, information-rich environments, this active, goal-directed sensing becomes increasingly complex [8], and yet adaptive human-computer interaction systems would benefit greatly from the ability to predict future targets of eye gaze in such complex environments. A computing system that can predict future targets of gaze could reduce its hypothesis space of possible future interactions and prepare its capabilities to the new targets of attention, which would decrease the friction in the interaction (for a discussion of the problem space, see [9]).

However, recent generative and predictive models of visual attention are still not performant in complex, dynamic environments [2, 8, 17]. In a real-world interaction task, visual scenes contain a large number of possible states for the model to consider, and the dynamical interaction between these states increases the state space exponentially with time. Therefore, due to the computational complexity of these models, the majority of prior research has focused on predicting gaze in static images [2–4, 13]. Computational models in the Bayesian generative framework [4, 8, 12, 19] become

EMICS '21, May 14, 2021, Yokohama, Japan

computationally intractable even for static images and require time-costly approximations for high-dimensional state spaces [1, 10]. However, predictive deep neural network (DNN) models have extended visual saliency mapping to short video clips and are capable of predicting regions in frames that might draw attention in the future with high precision [7, 15, 16, 18]. The drawback of these DNNs is that they use enormous number of parameters [2, 17], which limits consideration to short video clips (e.g., 5-30 seconds).

We hypothesized that the long-term dynamics of gaze contains rich statistical structure that can be used to predict future eye gaze. However, to date, this hypothesis has rarely been explored due to computational costs. Here, we developed a modeling framework with low computational and data requirements, which allowed us to investigate the statistical information in long-term gaze dynamics for predicting future targets of gaze. To this end, we collected eye-tracking data from participants passively observing a 16-minute simulated meeting in VR. Our model predicted the target of future gaze (e.g., an object) instead of predicting the 3-dimensional coordinates of the environment (see Fig. 1D). We call the relevant objects in the VR environment of our experiment *targets of interest* and filter the eye-tracking data by only considering eye gaze directed to these targets while disregarding eye gaze on other areas of the environment. Given that prior work has demonstrated that people tend to look at objects regardless of whether objects are defined by low-level visual [3] or high-level semantic [5] cues, our approach preserves the most informative part of eye gaze while providing the benefit of reduced computational costs. Our framework is highly flexible to any set of objects, allowing a designer to specify potential objects of interaction and then model visual attention across these, substantially reducing the computational burden while simultaneously increasing the interpretability of the model outputs for human-computer interaction applications.

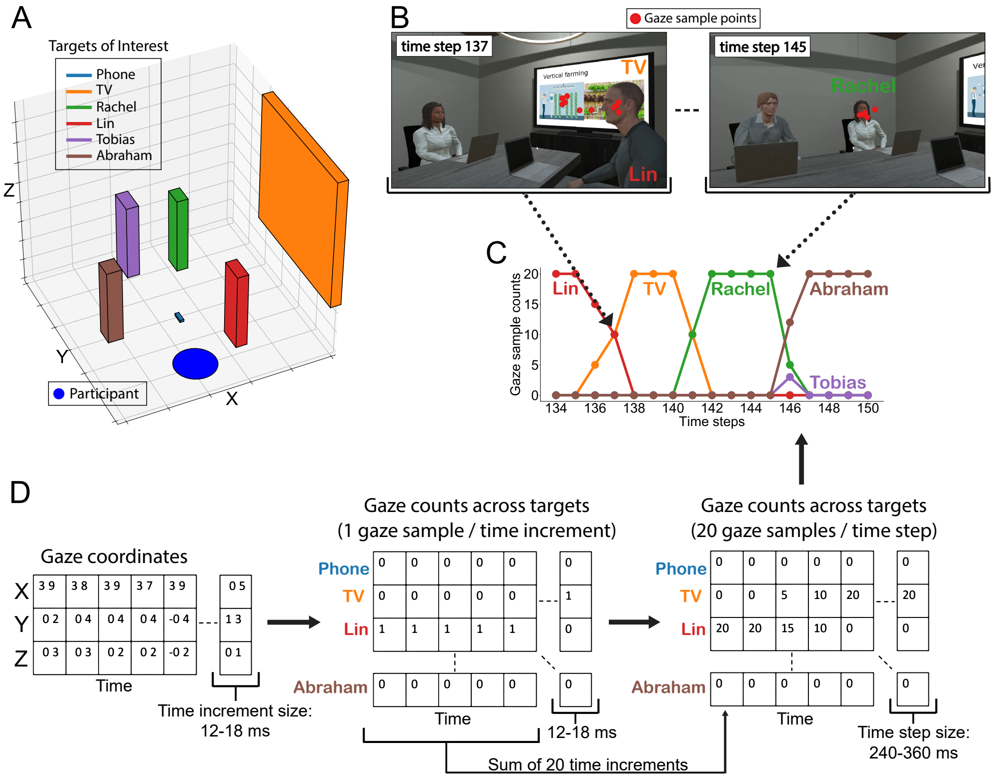
2 METHODS

2.1 Participants, Stimuli, Equipment, and Procedure

In the VR experiment, participants (n=6, 2 female, age range=35-45) each observed the same fictional discussion between four scientists who were planning an extension of a base on the moon. Participants' eye movements were recorded during the 16 minutes long VR meeting using a prototype VR headset with built-in eye tracking with sampling rate between 57-85hz (see A.1-3).

2.2 Analysis and models

First, we filtered the eye-gaze data to extract gaze samples directed to the targets of interest. We defined targets of interest as all dynamic objects and people during the meeting (see Fig. 1D and A.2 & A.4). Then, the gaze samples were summed across a "time step" and we considered three different sizes each of which contained 10, 20, and 30 gaze samples to ensure that our results were not dependent on an arbitrarily selected time step size (these corresponded to 118-175, 236-350, and 354-525 ms intervals, respectively, depending on sampling rate of the eye tracker). We had between 1829 to 8136 time steps in a session for each participant. Second, we trained a Vanilla RNN model [14] for each participant, which predicted the distribution of the observer's eye gaze across the targets of interest in the VR environment. At each time step, the input to our computational models corresponded to the gaze counts over each targets of interest and the non-target category (see Fig.1D). We compared the RNN model to a baseline model to test whether the RNN model could learn structure in long-term gaze dynamics. The baseline model (PREV) weighted the current gaze counts to predict the gaze counts in the next time step. We also compared the RNN model to a classical linear autoregressive model to investigate whether the structure in the long-term gaze dynamics is more complex than a linear combination of past gaze counts (see the details of



Gaze coordinates

X	39	38	39	37	39	05
Y	02	04	04	04	-04	13
Z	03	03	02	02	-02	01

Gaze counts across targets (1 gaze sample / time increment)

Phone	0	0	0	0	0	0
TV	0	0	0	0	0	1
Lin	1	1	1	1	1	0
Abraham	0	0	0	0	0	0

Gaze counts across targets (20 gaze samples / time step)

Phone	0	0	0	0	0	0
TV	0	0	5	10	20	20
Lin	20	20	15	10	0	0
Abraham	0	0	0	0	0	0

Fig. 1. Panel A. The map of the environment in the VR experiment. The blue circle on the map shows the place where the participants was located. The colored rectangles show the bounding boxes of the relevant objects and the people (i.e., targets of interest). Panel B. Screenshots of the VR meeting with overlaid eye gaze samples. Red dots show the locations of the gaze during the given time step. Note that the red dots were artificially generated for this figure. Panel C. Hypothetical data showing gaze sampling by time step. The colored lines represent the gaze counts over each target of interest. The figure shows how the two example scenes with the gaze sample points in panel B corresponds to two data points in the figure in panel C. Panel D. Data preprocessing details. For each gaze sample, we first identified the target based on the gaze coordinate (each gaze sample corresponded to a time increment of 12-18 ms depending on the sampling rate of the eye tracker); then, we summed 10, 20, and 30 gaze samples together to create larger time steps (here, we show 20 gaze samples per time step). All gaze samples that did not intersect a target of interest were assigned to a "non-target" category.

the models in A.5). The autoregressive and RNN models were trained and tested within-subject with hyperparameter tuning. During model training, we split our data into training (first 60% of the data), validation (next 20%), and test (last 20%) sets and used negative log likelihood for loss function (see A.5). The training, being self-supervised, did not require any special labels, as the model was predicting gaze counts at the next time step using gaze counts from the current time step. To evaluate the models, we computed fraction of variance unexplained (FVU) on the test set, contrasting the predicted gaze samples with the observed gaze samples for that time step. FVU is a standard error metric for regression tasks; lower values indicate a better fitting model reflecting stronger correlation between the predicted and the observed values on the test set (see A.5).

3 RESULTS

First, the contrast with the baseline model allowed us to test whether using the longer history of eye movements would improve the prediction of future eye gaze. If there is hidden structure in the long-term gaze dynamics, the RNNs should outperform the baseline (PREV). We found that RNNs outperformed the baseline by 71% on average and reduced the FVU by 79% (SD=9%) in the small 10 $\frac{\text{samples}}{\text{time step}}$, by 69% (SD=12%) in the medium 20 $\frac{\text{samples}}{\text{time step}}$, and by 67% (SD=12%) in the large 30 $\frac{\text{samples}}{\text{time step}}$ conditions compare to the baseline (Fig. 2). Second, the contrast with the autoregressive models allowed us to test whether the structure the RNNs learned in the long-term gaze dynamics is more complex than a linear combination of past gaze targets. The RNNs attained smaller errors on the test set than autoregressive models (AR) and reduced the FVU by 64% (SD=10%) in the small 10 $\frac{\text{samples}}{\text{time step}}$, by 55% (SD=12%) in the medium 20 $\frac{\text{samples}}{\text{time step}}$, and by 49% (SD=10%) in the large 30 $\frac{\text{samples}}{\text{time step}}$ conditions compare to the autoregressive models (Fig. 2). All of these results were consistent across all participants (see the lines in Fig. 2). Thus, despite our small sample (due to difficulties running eye-tracking studies during COVID-19), the remarkable consistency across participants provides evidence that gaze dynamics contain hidden structure that can be used to predict future eye gaze. Regarding the hyperparameter tuning, we found that for each participant the best performing RNNs were with 100 latent states (100 was the maximum value in our hyperparameter search) and models with one and two layers performed equally well.

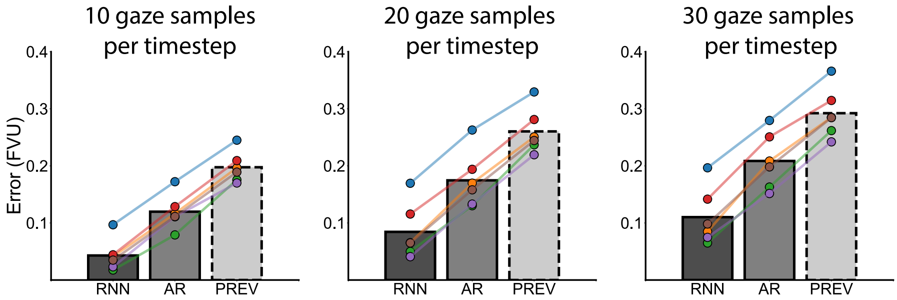


Fig. 2. Fraction of variance unexplained (FVU) computed on the test set in the three training conditions, each with different time step sizes. Lower FVU scores indicate better performance on the test set. The bars reflect the average FVU values across participants. The dots reflect each participant’s individual FVU value, with lines connecting the FVU values for each participant across the two conditions. PREV = weighted previous step baseline model. AR = linear autoregressive model. RNN = vanilla RNN time-series model.

4 DISCUSSION AND CONCLUSION

We trained within-subject, self-supervised Vanilla RNN models to predict the future targets of attention using eye-tracking data from a VR experiment. The results provide evidence for hidden structure in gaze for the considered scenarios, which can be used to predict future gaze. In particular, RNNs with one and two layers and 100 hidden states were able to improve upon the baseline and autoregressive models, on average, by 64%. Due to the effect of COVID-19, we were unable to collect a large sample in our laboratories. However, the observed pattern of results was highly consistent across all of our participants, suggesting robustness of these results.

The main advantage of our approach compared to previous studies of gaze prediction is that we filtered the environment based on relevant objects and people. This method moves away from the pixel/coordinate based description of the environment and focuses the predictive modeling to the targets of interest, which drastically reduces the dimensionality of the input and output of the models and allows for lighter weight systems that could be capable of predicting in real-time. For example, future experiments could show that our approach can handle motion of the observer and the targets of interest, a problem that is far more complex with a pixel-based approach. Furthermore, this approach provides interpretable outputs, which can be used by a designer to establish particular interaction rules. For example, when the output indicates attention to the augmented reality calendar app, the system could map controls to X actions, whereas when it indicates attention to the augmented reality messaging app, it could map controls to Y actions. Additionally, the modeling approach we demonstrate here is very flexible and can be applied to different tasks, environments, and research goals by simply redefining the targets of interest.

In this work, the models used gaze data alone to predict future eye gaze. In future studies, the modelling framework can incorporate additional input features (e.g., onset of the speech, head and hand movements, semantics of the conversation, history of interactions), which might further improve the performance of the considered models. For example, future experiments could show that our approach can handle motion of targets of interest (unlike a pixel-based approach), which happens in real world usage. The current model assumes a fixed set of objects in the environment and future work can extend the current framework to incorporate objects appearing, disappearing, and reappearing in a changing real world environment. Furthermore, we used within-subject modeling only and future work might explore group-level modeling to develop methods that could work for a new participant immediately upon donning a headset.

The modeling framework we present here provides a lightweight, computationally tractable method for real-time prediction of visual attention for human-computer interaction applications and adaptive interface design. Indeed, the adaptive interactions that can be enabled by accurate long-term attention prediction would greatly improve the user experience through reduced latency, reduced friction, reduced explicit interaction, and enhanced capabilities.

REFERENCES

[1] Jonathan Baxter and Peter L. Bartlett. 2001. Infinite-Horizon Policy-Gradient Estimation. *J. Artif. Int. Res.* 15, 1 (Nov. 2001), 319–350.

[2] Ali Borji. 2019. Saliency Prediction in the Deep Learning Era: Successes and Limitations. *IEEE transactions on pattern analysis and machine intelligence* (2019). https://doi.org/10.1109/TPAMI.2019.2935715

[3] Ali Borji and Laurent Itti. 2013. State-of-the-Art in Visual Attention Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 185–207. https://doi.org/10.1109/TPAMI.2012.89

[4] N. J. Butko and J. R. Movellan. 2008. I-POMDP: An infomax model of eye movement. In *2008 7th IEEE International Conference on Development and Learning*. 139–144. https://doi.org/10.1109/DEVLRN.2008.4640819

[5] Falk Huettig Floor de Groot and Christian N. L. Olivers. 2005. When meaning matters: The temporal dynamics of semantic influences on visual attention. *Journal of experimental psychology. Human perception and performance* 42, 2 (2005), 180–196. https://doi.org/10.1037/xhp0000102

[6] Stanton A. Glantz, Bryan K. Slinker, and Torsten B. Neilands. 2016. *Primer of Applied Regression Analysis of Variance, 3rd Edition.* McGraw-Hill Education / Medical.

[7] S. Gorji and J. J. Clark. 2018. Going from Image to Video Saliency: Augmenting Image Salience with Dynamic Attentional Push. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7501–7511. https://doi.org/10.1109/CVPR.2018.00783

[8] M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao. 2016. Learning to Predict Sequences of Human Visual Fixations. *IEEE Transactions on Neural Networks and Learning Systems* 27, 6 (2016), 1241–1252. https://doi.org/10.1109/TNNLS.2015.2496306

[9] Tanya R. Jonker, Ruta Desai, Kevin Carlberg, James Hillis, Sean Keller, and Hrvoje Benko. 2020. The Role of AI in Mixed and Augmented Reality Interactions. In *CHI ’20 extended abstracts Hum.-Comput. Interact.* (Honolulu, HI, USA). ACM Press, New York, NY. https://doi.org/"https://doi.org/10.1145/3334480.XXXXXXX"

[10] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101, 1 (1998), 99 – 134. https://doi.org/10.1016/S0004-3702(98)00023-X

[11] S. P. Liversedge and J. M. Findley. 2000. Saccadic eye movements and cognition. *Trends Cogn. Sci.* 4 (2000), 6–14. https://doi.org/10.7554/eLife.12215

[12] Jiri Najemnik and Wilson S. Geisler. 2005. Optimal eye movement strategies in visual search. *Nature* 434 (2005), 387–391. https://doi.org/10.1038/nature03390

[13] Tam V. Nguyen, Mengdi Xu, Guangyu Gao, Mohan Kankanhalli, Qi Tian, and Shuicheng Yan. 2013. Static Saliency vs. Dynamic Saliency: A Comparative Study. In *Proceedings of the 21st ACM International Conference on Multimedia* (Barcelona, Spain) *(MM '13)*. Association for Computing Machinery, New York, NY, USA, 987–996. https://doi.org/10.1145/2502081.2502128

[14] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536. https://doi.org/10.1038/323533a0

[15] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang. 2020. Inferring Salient Objects from Human Fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 8 (2020), 1913–1927. https://doi.org/10.1109/TPAMI.2019.2905607

[16] W. Wang, J. Shen, F. Guo, M. Cheng, and A. Borji. 2018. Revisiting Video Saliency: A Large-Scale Benchmark and a New Model. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4894–4903. https://doi.org/10.1109/CVPR.2018.00514

[17] W. Wang, J. Shen, and L. Shao. 2018. Video Salient Object Detection via Fully Convolutional Networks. *IEEE Transactions on Image Processing* 27, 1 (2018), 38–49. https://doi.org/10.1109/TIP.2017.2754941

[18] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao. 2018. Gaze Prediction in Dynamic 360° Immersive Videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5333–5342. https://doi.org/10.1109/CVPR.2018.00559

[19] Scott Cheng-Hsin Yang, Mate Lengyel, and Daniel M Wolpert. 2016. Active sensing in the categorization of visual patterns. *eLife* 5 (2016). https://doi.org/10.7554/eLife.12215

A SUPPLEMENTARY METHODS

A.1 Participants

Six participants (2 female, mean age = 40, age range = 35-45) completed our experiment. All participants were employees at Facebook Reality Labs. Our sample size was small due to substantial challenges with participant recruitment during the COVID-19 pandemic.

A.2 Stimuli and Equipment

In the experiment, participants observed a meeting in virtual reality (VR), which was designed and created for the experiment. The meeting involved a fictional discussion between four scientists who were planning an extension of a base on the moon. The dialogue was paired with realistic avatars and mouth movements were rendered from audio using the Oculus Lipsync package for Unity. Screenshots and the map of the environment can be seen in Fig. 1, panels A & B.

The VR experience was 16 minutes long and participants' eye movements were recorded using a prototype VR headset with built-in eye tracking. The sampling rate of the eye tracking varied between 57-85hz across participants due to slight variation in the headset prototypes of the participants. The eye trackers were calibrated to an angular accuracy of less than 1.1 degrees. The VR headset included 2K x 2K displays with 100° field-of-view for VR visuals.

Instead of predicting the coordinates of the eye gaze, we predicted the distribution of eye gaze across relevant objects (i.e., targets of interest) in the VR environment (see Fig. 1D). In this environment, we defined targets of interest as all dynamic objects and people during the meeting. These targets of interest comprised the following: the four animated characters, the projection screen (on which the important information appeared during the meeting), and the animated cell phone of the participant that buzzed and displayed text messages several times during the experiment. The rest of the objects (others' laptops, chairs, lamp, and desk) in the environment were static and did not convey any information to the participant during the experiment; thus, prior to analyses, they were excluded.

A.3 Study Procedure

Prior to launching the VR meeting, participants seated themselves in a comfortable position and were asked to passively observe the meeting and behave naturally. Participants' head motion was not restricted, and they were encouraged to look around freely. They were also informed that they would not have to interact with the characters during the meeting.

A.4 Data Preprocessing

The eye-gaze data preprocessing was crucial in our computational framework to reduce drastically the number of possible spatial states that the gaze could take. Instead of using each pixel or grid location as possible states of eye movements in the 3-dimensional environment, we only considered a subset of relevant objects. This reduction in the state-space allowed us to use data from long dynamic scenes as input to computational models; if we were instead to use pixel/coordinate-based state-spaces, it would be intractable or very computationally costly for such long scenes.

The eye-gaze data preprocessing steps are shown in Fig. 1, panel D. First, we identified the target of each gaze sample based on the spatial coordinates that intersected the gaze ray. All gaze samples that did not intersect a target of interest were assigned to a "non-target" category. Next, we binned the gaze samples into time steps. For each time step, we extracted the number of gaze samples for each targets of interest and the non-target category. On average, the non-target category contained 32% percent of the gaze samples

To ensure that our results were not dependent on an arbitrarily selected time step size, we considered three different sizes each of which contained 10, 20, and 30 gaze samples. We used the number of gaze samples instead of time duration in ms to determine the time step sizes for the participants because we wanted to have the same number of gaze samples in a time step for all participants. Had we used fixed time duration in ms for the time steps across participants we would have had different number of eye gaze samples in a time step for our participants because the sampling rates of their eye tracking devices were different (it varied between 57-85hz). The 10, 20, and 30 time step sizes corresponded to 118-175, 236-350, and 354-525 ms intervals, respectively, depending on the sampling rate of each participant's eye-tracker. Given that the length of the experimental sessions was 16 min, we had between 1829 to 8136 time steps in a session for each participant.

The input to the time-series models at each time step corresponded to the gaze counts over all 7 categories (targets of interest and the non-target category) at the current time step. To train and test the time-series models we used separate training, validation, and test windows of multiple time steps; each window contained the time-series of gaze counts as input.

A.5 Analysis and Models

Previous research has not yet explored whether the long-term dynamics of gaze can be used to predict future gaze targets. Therefore, as a first step, we implemented classical RNN time-series models that can learn the hidden long-term dynamics of gaze. The preprocessing step described above resulted in a time-series of gaze counts over the targets of interest. We modeled these data as a time-series:

$$y_t^i \in \{0, \ldots, n_t\}, \quad i = 1, \ldots, C, \quad t = 1, \ldots, T, \text{ with } \sum_i y_t^i = n_t \,, \tag{1}$$

where $y_t^i \in \mathbb{N}$ denotes the number of eye gaze samples landing on target of interest i at time step t, C denotes the total number of targets of interest (including the non-target category), T denotes the total number of time steps, and n_t denotes the total number of gaze samples in each time step t.

Thus, for time step t, the vector $\mathrm{Y}_t \equiv [y_t^1, \ldots, y_t^C] \in \mathbb{N}^C$ represents gaze-sample counts for each target of interest from the total number of eye gaze samples n_t.

The Vanilla RNN assumes a latent-state time-series regression model where the time-series Y_t, $t = 1, \ldots, T$ is driven by latent dynamics; i.e., the time-series depends on a hidden process $\mathrm{Z}_t \in \mathbb{R}^p$, $t = 1, \ldots, T$, which itself depends on the history of the latent process and some external input. For input, we only employed the gaze counts from the current time step Y_{t-1}. In particular, we employed the following formulation for latent-dynamics models:

$$\mathrm{Z}_t = f(\mathrm{Z}_{t-1}, \mathrm{Y}_{t-1}) \tag{2}$$

$$\boldsymbol{\lambda}_t = h(\mathrm{Z}_t) \tag{3}$$

$$\pi_t^i = \frac{\exp\left(\lambda_t^i\right)}{\sum_{j=1}^{C} \exp\left(\lambda_j^t\right)} \tag{4}$$

$$p(\mathrm{Y}_{t+M} \mid n_t, \boldsymbol{\pi}_t) = \frac{n_t!}{\prod_{i=1}^{C} y_{t+M}^i!} \prod_{i=1}^{C} \pi_t^{i\, y_{t+M}^i} \ , \tag{5}$$

where the functions f and h model the latent dynamics and the mapping from the latent states to the multinomial logits, respectively. Here, M denotes the number of time steps in the future for prediction. We used $M = 0$ for training the models to predict one time step ahead. The hidden process was initialized with zeros. We computed the predicted values of the future gaze counts over the targets of interest by taking the expected value of Y_{t+M}. The hyperparameters of the Vanilla RNN comprised the number of layers and the number of hidden units. Note that this is a *self-supervised* framework, as the model does not require any special labels; the inputs and the outputs correspond to the same visual attention data. This method allows us to investigate the extent to which the targets of eye gaze can be predicted solely from the structure hidden in the history of eye gaze. As such, future work will consider additional input features (e.g., the onset time of speech, head movements, or semantics of the conversation) in the model, as their inclusion can only improve the models' performance beyond what is reported in the current study.

During model training, we split our data into training, validation, and test sets. We kept the ratio of the training, validation, and test sets fixed at 60%, 20% and 20%, respectively. Focusing on real world applications, we explored whether we could predict time steps from the latter portion of the VR session by training the model only on the first 60% of the session. Thus, we used the first 60% of the time steps for the training, the next 20% for the validation, and the last 20% for the test sets. We split the validation set into two equal sets. One split was used to determine when to stop the training algorithm, and the other was used to tune hyperparameters. We applied the Adam optimizer [20] with early stopping on the first validation set. Early stopping evaluates the model's prediction performance after each optimization iteration using a validation set. If the change in the prediction performance on the validation set becomes negligible (as defined by a threshold value), then optimization iterations are terminated. We used the second validation set to tune hyperparameters and choose the best performing model architecture for each model type. We defined a hyperparameter grid for the number of layers (1, 2) and the hidden units (5, 10, 50, 100) per layer.

We used negative log likelihood computed from the multinomial distribution (4) for loss function in the training:

$$-\log \mathrm{L}(\boldsymbol{\theta}) = -\sum_{t=1}^{T} \left[\log \frac{n_t!}{\prod_{i=1}^{C} y_{t+M}^i!} + \sum_{i=1}^{C} y_{t+M}^i \log \pi_t^i(\boldsymbol{\theta}) \right] \ , \tag{6}$$

where $\boldsymbol{\theta}$ denotes the parameters of the model. We evaluated each model's performance by computing the fraction of variance unexplained (FVU) on the test set:

$$FVU = \frac{\sum_{n_{\text{test}}}(\mathrm{Y}_t - \mathrm{Y}_t^{\dagger})^2}{\sum_{n_{\text{test}}}(\mathrm{Y}_t - \overline{\mathrm{Y}})^2} \quad \text{where} \quad \overline{\mathrm{Y}} = \frac{1}{n_{\text{test}}}\sum_{n_{\text{test}}}\mathrm{Y}_t \quad \text{and} \quad \mathrm{Y}_t^{\dagger} = n_t\boldsymbol{\pi}_t \ , \tag{7}$$

where $\mathrm{Y}_t^{\dagger}$ are the predicted values of the models for the eye gaze counts at time step t, which were computed as the expected value of Y_t. FVU is a standard error metric for regression tasks and is equal to $FVU = 1 - R^2$ where R^2 is the coefficient of determination [6]. FVU is closely related to the linear correlation coefficient (CC) metric through the coefficient of determination, which measures the proportion of variance in the actual gaze that is predictable from the gaze predicted by the model. CC measures how correlated the predicted and actual gaze in image/video saliency prediction task. Thus, both metrics (FVU and CC) represent the degree of correlation between the predicted and actual gaze, however FVU is an error metric (the lower the FVU the better) while CC is a performance metric (the larger the CC the better). We choose to use FVU instead of CC because it is a more frequently used metric for time-series regression models (that we used here), however both metrics would convey the same information about a model's performance in our modelling framework.

CC is one of the three most frequently used evaluation metrics in eye gaze prediction tasks, along with Normalized Scanpath Saliency (NSS), and the area under the receiver operating curve (AUC). AUC and NSS metrics are useful evaluation metrics for classification tasks. However, in our modeling framework, both NSS and AUC would give misleading evaluation for the models' performance because our framework involves predicting counts as outcomes i.e., the distribution of gaze samples across relevant objects in the environment; see the loss function in (6), which was computed from the multinomial distribution in (4).

We compared the Vanilla RNN models to a baseline models. The baseline model meant to capture a simple computation that do not use the long-term dynamics of gaze and only weighted the values in the previous time step by a parameter for each targets of interest and the non-target category to predict future gaze counts, wherein $\boldsymbol{\lambda}_t = \omega\mathrm{Y}_{t-1}$ where $\omega \in \mathbb{R}^{c\times c}$, c is the number of targets of interest and the non-target category, and (4)–(5) above still hold.

Finally, we also considered a simple linear autoregressive model (AR) that does not model any latent dynamics, wherein $\boldsymbol{\lambda}_t = \omega_1\mathrm{Y}_{t-1} + \omega_2\mathrm{Y}_{t-2} + \ldots + \omega_m\mathrm{Y}_{t-m}$ where $\omega_i \in \mathbb{R}^{c\times c}$ and (3)–(4) above still hold. The hyperparameter of this model was the number of previous time steps m to include in the linear combination (the values 5, 10, 50, and 100 were considered during the hyperparameter tuning). Note that the PREV baseline is a special case of the AR model with $m = 1$. The training methods for the autoregressive models were the same as for the RNNs.