

Forecasting 3D Scanpaths in Egocentric Video

Supplementary Material

6. Visualizations

6.1. Video Visualizations

We include a demo video of our model’s outputs overlaid over egocentric video clips in `demo.mp4`. For each frame, we show the most recent observed fixation point (marked with a white dot in the middle), and our model’s predictions for the next 10 future fixation points.

6.2. 3D Visualizations

We show examples of our model’s predicted scanpaths with 3D visualizations in Fig. 7. For visualization purposes, we overlay the scanpath on the scene’s static point cloud obtained from Project Aria’s SLAM system. Note that the point cloud is static, so dynamic elements of the scene (e.g. a chair that is moved around the room) may not be visible in the point cloud or may be in a different position than in the visualized video frames.

7. Additional Implementation Details

7.1. Model & Training

We use the `facebook/dinov2-base` model weights on Huggingface for our visual encoder, and keep all encoder weights frozen during training. All transformer layers in our model’s Visual Context Encoder and Trajectory Decoder have 8 attention heads and MLP dimension 2048. In total, our model has ≈ 6.2 M learnable parameters. We train with batch size 128 and dropout=0.1. Our loss hyperparameters are $\lambda_1 = 1.5$ and $\lambda_2 = 1$. For all experiments, checkpoints are selected based on performance on the validation split. All experiments were conducted on a single NVIDIA A100 GPU.

7.2. TPP-Gaze Baselines

We adopt TPP-Gaze [20], a recent SOTA image scanpath prediction model, as a baseline in the zero-shot 2D, finetuned 2D, and finetuned 3D settings. TPP-Gaze consists of a Neural Temporal Point Process model that predicts the next fixation conditioned on a history of past fixations (observed or predicted) and image features. The architecture consists of a learned image encoder and a Transformer module for predicting fixations. The model predicts the parameters for a Log-Gaussian Mixture Model (LGMM) for the fixation duration and a 2D Gaussian Mixture Model (GMM) for the fixation position in normalized image coordinates. Given the image and previously observed or predicted fixations, the next fixation is predicted by sampling the predicted LGMM and GMM.

Table 6. TPP-Gaze zero-shot 2D performance with metric statistics computed over 10 trials with temperature 2, compared with 1 trial with temperature 1e-12 (reported in Tab. 3).

metric	10 trials, $\tau = 2$			1 trial, $\tau = 1e-12$
	mean (std)	min	max	
DTW	2.451 (0.611)	1.519	3.549	1.507
EUC	0.895 (0.211)	0.570	1.267	0.564
FRE	0.451 (0.126)	0.265	0.683	0.266
EYE	0.273 (0.081)	0.157	0.427	0.160
TDE	0.103 (0.038)	0.053	0.176	0.071
MM Sh	0.314 (0.122)	0.140	0.543	0.163
MM Dir	1.434 (0.432)	0.754	2.169	1.315
MM Len	0.285 (0.121)	0.114	0.514	0.138
MM Pos	0.221 (0.074)	0.116	0.358	0.139
MM Dur	0.606 (0.118)	0.401	0.790	0.606

Zero-shot Evaluation We evaluate TPP-Gaze zero-shot by running it on the canonical image frame for each example. We resize our images to 512×512 to match TPP-Gaze’s training resolution. As with our model, we provide the prior N_o fixations projected into canonical image coordinates to the model as observed fixations. We sample with a very small temperature, $\tau = 1e-12$ to select the most likely point in the predicted fixation distribution and report the results with these predictions in Tab. 3. We note that the TPP-Gaze codebase’s default inference code uses temperature $\tau = 2$, which introduces more randomness in sampled paths. In Tab. 6, we calculate the mean, standard deviation, minimum, and maximum values for each metric across 10 trials with $\tau = 2$. We observe that the minimum value for each metric across the trials is similar to the value obtained with a single trial with $\tau = 1e-12$. Thus, we report the results for a single trial with $\tau = 1e-12$ in Tab. 3.

Finetuned Baselines We finetune TPP-Gaze on our dataset in both the 2D and 3D setting. Following the paper, we optimize the model with negative log-likelihood loss. We train for 10 epochs on our dataset projected into 2D with learning rate 1e-3, the AdamW optimizer, batch size 128, and dropout=0.1. We use image size 512×512 during training and inference to match the pretrained TPP-Gaze model. We select the best epoch based on validation performance. In the 2D setting, we exclude trajectories with observed or future points that are outside of the canonical frame’s field of view from training and evaluation. To adapt the TPP-Gaze model to the 3D setting, we add a third dimension to the GMM for predicting fixation position as a 3D coordinate, and remove the TanH activation function in order to predict unnormalized Cartesian coordinates. We train for 10

epochs on our dataset with learning rate 1e-3, the AdamW optimizer, batch size 128, and dropout=0.1. For both the 2D and 3D finetuned models, we evaluate by sampling with temperature=1e-12.

7.3. Metrics

MultiMatch For the MultiMatch metrics [19], we use the Python implementation from [1] and adapt it for evaluation of 3D scanpaths by using 3D Euclidean distance in place of 2D Euclidean distance computations, and 3D angular distance in place of 2D angular distance. We do not perform the optional scanpath simplification process within MultiMatch because we already preprocess our input and ground truth scanpaths into fixation groups based on velocity. We report all MultiMatch metric values unnormalized.

Distance metrics We use the implementations for the DTW, EUC, EYE, FRE, and TDE metrics from [2]. For TDE calculation, we use subsequence length $k=2$. When calculating 2D pixel metrics for models that predict 3D world coordinates (Tab. 3), we clip predictions that are projected to 2D points beyond the image frame to coordinates within the frame. For descriptions and comparisons of the individual metrics, we refer to relevant surveys [6, 22].

8. Dataset Details

Recordings We use the 184 sequences in the Aria Digital Twin (ADT) dataset [55] recorded in the “Apartment” environment, which include the camera wearer performing everyday activities such as cooking, cleaning, and socializing with another person. We exclude the 52 sequences from the “Office” environment in ADT because they are exclusively recordings of the activity “object inspection”, in which the camera wearer looks at a single object throughout the full recording. As a result, these sequences are not suited for predicting dynamic scanpaths. We undistort all video frames in the original videos.

Scanpath Fixation Processing ADT provides 30Hz eye tracking estimates via the Project Aria glasses eye tracking system [21, 49], which estimates the 3D direction of gaze. ADT is recorded in an environment with a digital twin for both the static and dynamic elements of the scene (including objects and bodies), providing ground truth 3D gaze as the intersection of the estimated gaze direction with the surface of the environment. We process 3D gaze points into fixation clusters by grouping together consecutive gaze points with 3D movement between them that is below a velocity threshold τ . We note that at 30Hz, ADT eye tracking is sparser than traditional eye tracking systems, so this preprocessing primarily serves to group together gaze points on the same location rather than explicitly filter saccades.

Table 7. Performance breakdown by head movement.

Head movement	data %	DTW↓	EUC↓	FRE↓	EYE↓	TDE↓
Low (<0.25m)	23%	1.475	1.488	2.661	1.922	1.039
Med (0.25-1.5m)	61%	1.384	1.395	2.170	1.809	1.015
High (≥ 1.5 m)	16%	1.201	1.204	2.046	1.584	0.859

Table 8. Zero-shot performance on sequences from Ego-Exo4D.

Method	DTW↓	EUC↓	FRE↓	EYE↓	TDE↓
Last observed point	1.834	1.857	3.245	2.467	1.360
Ours	1.743	1.770	2.944	2.188	1.128

We use $\tau = 3m/s$ as our threshold. The 3D location of the fixation is determined as the average of the gaze points of all data points belonging to the fixation group. Our sequences of $N_o = 10 + N_f = 10$ fixation points average ≈ 2.8 seconds in length. We exclude outlier sequences that exceed 10 seconds from training and evaluation.

Dataset Statistics Scanpaths in the dataset span an average of 2.8 seconds and length and transition between an average of 10 entities (e.g. cup, table). 25% of fixations are on entities that are annotated as dynamic in the Aria Digital Twin dataset (e.g. objects that move during the recording) while 75% are on static entities (e.g. a piece of furniture that does not move). We show the breakdown of absolute head translation in the sequences in Tab. 7, along with the performance of our model. We observe that error is actually lower on cases with more head motion, where head direction is often more indicative of gaze.

9. Additional Experiments

Performance on Unseen Environment We use the ADT dataset in our experiments because it offers high quality 3D gaze ground truth via the intersection of estimated eye tracking and the 3D digital twin of the environment. However, ADT is recorded in a single multi-room environment. To demonstrate generalization to unseen scenes, camera wearers, and activities, we evaluate our ADT-trained model zero-shot on a subset of 1545 trajectories from 60 unique takes from the Ego-Exo4D [23] test set below, which includes activities that are strongly out of domain from ADT such as rockclimbing. We select only takes with calibrated personalized eye tracking. We note the 3D gaze ground truth in Ego-Exo4D is noisier than ADT due to sparser eye tracking (10Hz) and gaze depth estimated via eye vergence due to the lack of a digital twin 3D reconstruction.

Architecture Layers & Dimensions We provide additional experiments to give insight into our choice of internal model dimension in Tab. 9 and number of layers in the

Table 9. Internal dimension of model.

d	DTW↓	EUC↓	FRE↓	EYE↓	TDE↓
128	1.406	1.404	2.195	1.888	0.903
256	1.377	1.382	2.173	1.800	0.859
512	1.399	1.418	2.223	1.799	0.860
768	1.380	1.398	2.203	1.790	0.867

Table 10. Number of transformer layers in Visual Context Encoder (VCE) and Trajectory Decoder (TD).

Num. layers		DTW↓	EUC↓	FRE↓	EYE↓	TDE↓
VCE	TD					
1	1	1.401	1.418	2.216	1.818	0.865
2	1	1.383	1.398	2.180	1.805	0.858
1	2	1.387	1.404	2.209	1.808	0.858
2	2	1.377	1.382	2.173	1.800	0.859
3	3	1.379	1.395	2.190	1.804	0.847

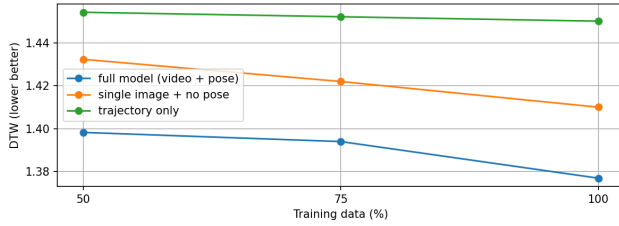


Figure 6. Performance of our model variants trained on reduced training data.

Visual Context Encoder and Trajectory Decoder in Tab. 10. Our final model uses $d = 256$ and 2 transformer layers each in the Visual Context Encoder and Trajectory Decoder.

Data Reduction The Aria Digital Twin is limited in scale, and we hypothesize that training on larger data may enable developing a model that sees greater benefits from temporal visual information and head pose. Fig. 6 shows the performance obtained by training our model with and without temporal visual information on randomized subsets of 50% and 75% of the training data. We observe that our full model with video and pose information shows a stronger performance boost trend as data scale increases. This result suggests that scaling training data for our task is a promising direction for future work.

