# GAZEFLOW: From Human Gaze Behavior to Generative Egocentric Gaze Prediction

**Sheng Zhao**
Department of Computer Science
University of Rochester
szhao32@ur.rochester.edu

**Weikai Lin**
Department of Computer Science
University of Rochester
wlin33@ur.rochester.edu

**Yuhao Zhu**
Department of Computer Science
University of Rochester
yzhu@rochester.edu

## Abstract

Egocentric gaze prediction enables many downstream applications but remains challenging, as human gaze is inherently stochastic. This stochasticity is constrained by structured temporal dynamics alternating between fixations and saccades, top-down influences from tasks, and bottom-up visual saliency. Based on this observation, we introduce GAZEFLOW, a framework that directly models gaze as a joint distribution of temporal gaze positions conditioned upon both top-down and bottom-up information. In particular, GAZEFLOW uses conditional flow matching (CFM): a learned velocity field iteratively transports a Gaussian noise sample into a plausible gaze trajectory drawn from this joint distribution. The velocity field is conditioned on bottom-up visual features extracted by a video encoder and on top-down task information obtained by globally querying these features. On standard datasets, GAZEFLOW achieves state-of-the-art performance on per-frame metrics, and the generated trajectories align better with human gaze temporal dynamics. The project page is at: https://horizon-lab.org/gazeflow/.

## 1 Introduction

The eye provides a direct window into the brain. Egocentric video captures visual experience from the wearer's perspective, offering a natural way to probe this link—provided that gaze is known. However, gaze is not directly observable in such videos without dedicated eye-tracking hardware, which requires calibration and introduces additional cost and complexity in real-world deployment. At the same time, access to gaze would enable a range of applications, including foveated video compression [18, 22, 40, 10, 30], hazard anticipation in autonomous driving [42], power-efficient rendering [35, 36], and imitation learning in robotics [24]. A natural alternative is to infer where the wearer looks directly from the video itself, a problem known as egocentric gaze prediction [17, 26, 31].

Gaze patterns are notoriously difficult to predict because of the stochasticity, as widely recognized in prior work [26, 17, 31, 55]: even for the same observer viewing the same scene, trajectories are not exactly reproducible. This stochasticity, however, is not purely random; rather, it is jointly constrained by three factors. First, human gaze trajectories exhibit structured dynamics under the oculomotor control, characterized by alternating cycles of saccades and fixational eye movements superimposed with occasional microsaccades [47]; this imposes strong temporal constraints on how gaze can evolve over time. Second, gaze is strongly task-dependent [49, 13, 46, 14]; information about the task, often revealed in future frames, can therefore inform where the observer is likely to look at the current

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

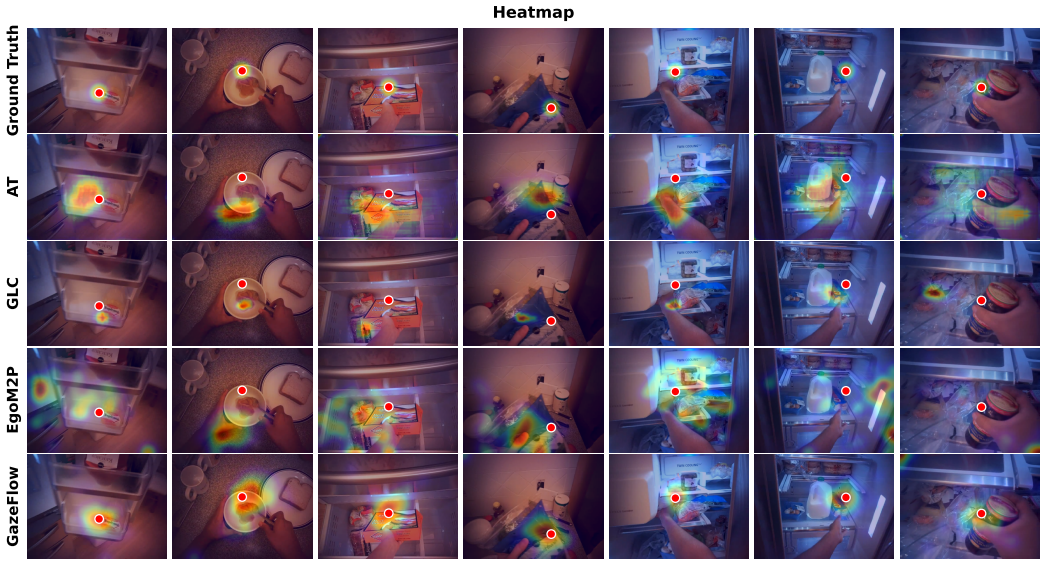



Figure. 1: **Qualitative comparison on EGTEA Gaze+.** Example frames where GAZEFLOW predicts a heatmap tightly localized on the ground-truth gaze positions (red dots) compared to existing methods (AT [17], GLC [26], EgoM2P [31]).

moment. Third, gaze is also influenced by stimulus-driven factors such as conspicuity and visual saliency [19, 51], which requires analyzing the spatio-temporal signal properties in the scene.

Taken together, gaze pattern is best modeled as a *joint* distribution of temporal gaze positions conditioned upon the *entire* video sequence. Any gaze trajectory arising from interacting with the scene captured in the video can then be viewed as a sample from this distribution. Modeling the joint distribution enables us to capture the global temporal dynamics of gaze, while conditioning on the full video allows us to account for both top-down influences, which arise from longer-term contextual and task-related information (on the order of seconds), and bottom-up saliency (e.g., local contrast or flicker), which operate at shorter timescales (tens of milliseconds).

We formalize these observations into a computational framework, GAZEFLOW. In GAZEFLOW, egocentric gaze prediction is formulated as a generative process, where each generated trajectory is a sample from the underlying human gaze distribution. In particular, we use conditional flow matching (CFM), a generative modeling method. We use a video encoder to extract bottom-up spatio-temporal features from video frames. To incorporate top-down influences, the encoded video embeddings are allowed to be globally queried by gaze at any time, yielding task-relevant information. The bottom-up and top-down embeddings are then used to condition the generation process.

We evaluate GAZEFLOW on two common egocentric video datasets, EGTEA Gaze+ [33] and Ego4D [12]. Not only does GAZEFLOW achieve state-of-the-art performance on per-frame metrics, the predicted trajectories also exhibit closer alignment with the temporal dynamics of human gaze. Figure 1 shows the qualitative results. In summary, this paper makes the following contributions:

- Starting from human gaze mechanisms, we first formulate egocentric gaze prediction as a conditional joint-distribution modeling problem.
- We propose a flow matching-based generative algorithm, where gaze trajectory generation is conditioned on both bottom-up (saliency) and top-down (task-specific) information from a video.
- We demonstrate that GAZEFLOW achieves state-of-the-art per-frame prediction accuracy on EGTEA Gaze+ and Ego4D, and is better aligned with the temporal dynamics of human gaze.

Table 1: **Comparison of egocentric gaze prediction paradigms.** $\mathcal{V}$ is the input video and $\mathcal{V}_{\leq t}$ its causal prefix, $\mathbf{g}_t \in \mathbb{R}^2$ is the gaze position at frame $t$, $\mathbf{g}_{1:T}$ the full trajectory, and $q_\theta(\cdot \mid \cdot)$ denotes a model's parametric conditional distribution. MSE: mean squared error, KL: Kullback–Leibler divergence, and BCE: binary cross-entropy. ✓ if satisfied, × if not satisfied, and ∼ if architecturally implied but not enforced.

| | | | | Human gaze mechanism | | | |
|---|---|---|---|---|---|---|---|
| Method | Paradigm | Prediction | Training objective | Bottom-up | Top-down | Joint | Stochastic |
| Li et al. [32] | Regression | $\mathbf{g}_t \mid \mathcal{V}$ | per-frame MSE | × | ∼ | × | × |
| GLC [26] | Marginal density estimation | $q_\theta(\mathbf{g}_t \mid \mathcal{V})$ | per-frame KL | ✓ | × | × | ✓ |
| DFG [54] | | $q_\theta(\mathbf{g}_t \mid \mathcal{V})$ | per-frame KL | ✓ | × | ∼ | ✓ |
| AT [17] | Autoregressive | $q_\theta(\mathbf{g}_t \mid \mathbf{g}_{t-1}, \mathcal{V}_{\leq t})$ | per-frame BCE | ✓ | ∼ | ∼ | ✓ |
| EgoM2P [31] | | $q_\theta(\mathbf{g}_t \mid \mathbf{g}_{<t}, \mathcal{V})$ | masked $L^2$ | ∼ | ∼ | ∼ | ✓ |
| **GAZEFLOW (Ours)** | Distribution mapping | $q_\theta(\mathbf{g}_{1:T} \mid \mathcal{V})$ | $L^2$ velocity field | ✓ | ✓ | ✓ | ✓ |

## 2 Related Work

**Mechanisms of Human Gaze.** Human gaze is shaped by three complementary mechanisms. Bottom-up saliency is driven by low-level visual features such as contrast, orientation, and motion [20, 19]. Top-down task control guides the eyes to anticipate upcoming actions and select objects relevant to the current goal [53, 28, 13, 49, 46, 56]. Oculomotor control further imposes structured temporal dynamics on gaze. These dynamics alternate between saccades, fixations, and microsaccades, constraining how gaze evolves over time [47]. GAZEFLOW addresses these factors jointly: a visual encoder supplies bottom-up evidence, a task-token bank supplies top-down context, and conditional flow matching captures temporal structure.

**Joint Distribution Modeling.** Modeling a high-dimensional joint distribution generally falls into two families. Autoregressive factorization decomposes the joint into per-step conditionals [52, 4], while distribution mapping transports a simple prior to the data via a learned map, as in GANs [11], VAEs [25], diffusion [15], and flow matching [37, 50]. GAZEFLOW adopts the latter, sampling a full trajectory through one video-conditioned Ordinary Differential Equation (ODE) transport (Section 4).

**Egocentric Gaze Prediction.** Existing methods fall into four paradigms summarized in Table 1.

*Regression* directly regresses one gaze point per frame from the video, trained with per-frame Mean Squared Error (MSE). Early egocentric methods regress over hand-crafted cues [32]. This collapses gaze to a single mode and cannot capture its multimodality. *Marginal density estimation* conditions on the video and independently predicts a per-frame heatmap per frame. GLC [26] produces this heatmap by correlating each frame with a global clip summary through a transformer. CSTS [27] additionally exploits sound, and DFG [54] learns a GAN. Per-frame factorization misses temporal coupling across frames.

*Autoregression* factors the joint distribution into a chain of per-frame conditionals on previously generated gaze. Attention Transition [17] predicts each fixation by recursively conditioning on the previous-frame fixation through an LSTM-based attention-transition module. EgoM2P [31] decodes VQ-VAE-quantized gaze tokens within a multimodal pretraining stack. This captures temporal dependency but accumulates errors under self-rollout. *Distribution mapping* models the full trajectory directly, sampling each trajectory by transporting a simple prior through a learned map. GAZEFLOW adopts conditional flow matching [37, 50]: a learned velocity network defines a video-conditioned velocity field that transports each Gaussian noise sample into a gaze trajectory. This jointly captures temporal coupling, stochasticity, and the bottom-up/top-down factors of natural gaze (Section 3).

## 3 GAZEFLOW

In egocentric gaze prediction task, a head-mounted video $\mathcal{V}$ of $T$ frames and its ground-truth gaze trajectory $\mathbf{g} \in \mathbb{R}^{2T}$ are recorded jointly while human performs a task. Gaze is inherently stochastic and depends on both what the human is doing and the visual saliency in the scene. So $\mathbf{g}$ is one realization of a distribution conditioned on $\mathcal{V}$. We write $p^\star(\mathbf{g} \mid \mathcal{V})$ for the population distribution

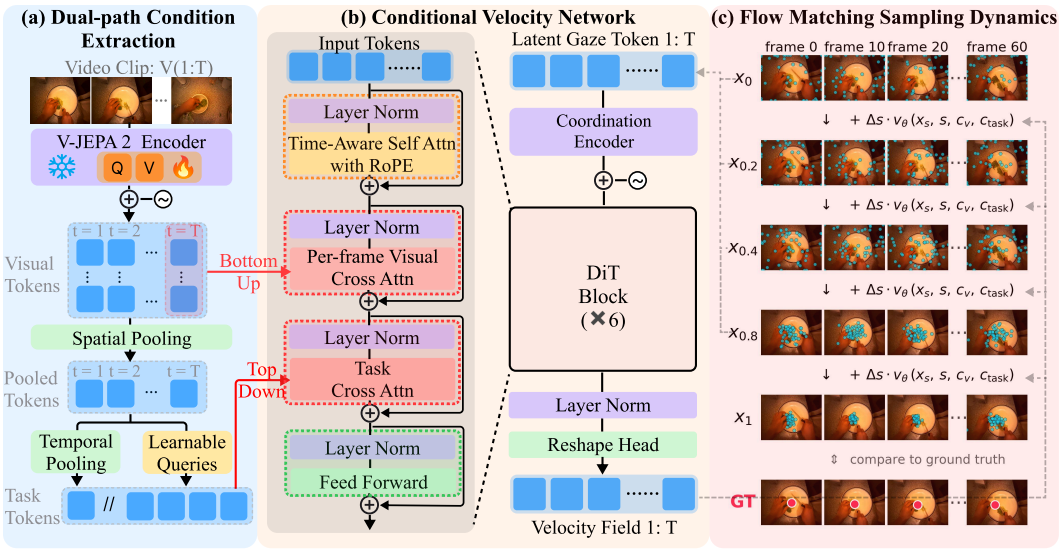



Figure. 2: **GAZEFLOW architecture.** (a) Dual-path condition extraction (Section 3.1). A LoRA-adapted V-JEPA 2 backbone produces bottom-up visual tokens $\mathbf{c}_v$; spatial–temporal pooling and learnable queries form the top-down task-token bank $\mathbf{c}_{\text{task}}$. (b) Conditional velocity network (Section 3.2). Latent gaze tokens encoding $\mathbf{x}_s$ and $s$ pass through six DiT blocks, each containing four sub-layers; a linear head outputs the velocity field $\mathbf{v}_\theta$. (c) Flow-matching sampling dynamics. Euler integration of $\mathbf{v}_\theta$ transports Gaussian noise $\mathbf{x}_0$ to a clean trajectory $\mathbf{x}_1$; the cyan sample cloud at $s \in \{0, 0.2, 0.4, 0.8, 1\}$ contracts toward the ground truth.

over all gaze trajectories given $\mathcal{V}$, and each annotated trajectory in the dataset is one sample drawn from $p^\star$. Our goal is to learn a model distribution $q_\theta(\mathbf{g} \mid \mathcal{V})$ that approximates $p^\star$, the underlying distribution we want to recover. Since we have only samples from $p^\star$, not a closed-form density, we adopt conditional flow matching (CFM) [37, 50], whose training objective is a sample-based regression that fits a velocity field transporting Gaussian noise to $q_\theta$ (see details in Appendix B).

Figure 2 shows the architecture of GAZEFLOW, which contains two main components: First, video feature extraction computes bottom-up visual tokens and top-down task tokens (Section 3.1). Second, the conditional velocity network uses these tokens for calculating a velocity transport that maps a Gaussian noise to target trajectory distribution (Section 3.2). We discuss how our framework can deal with training and inference on long videos (Section 3.3).

### 3.1 Dual-Pathway Video Condition Extraction

The video-conditioning module $E_\psi(\cdot)$ maps an input clip to two conditions. The first condition is bottom-up and frame-local. It comes from a pretrained video encoder. The second condition is top-down and clip-level. It comes from a learned task-token bank. This dual-pathway design follows the gaze mechanisms reviewed in Section 2: Human gazing depends on both low-level visual saliency and high-level task goals. Formally,

$$\mathbf{c} := \big(\mathbf{c}_v, \mathbf{c}_{\text{task}}\big) = E_\psi(\mathcal{V}), \tag{1}$$

where $\mathbf{c}_v$ contains frame-level visual tokens and $\mathbf{c}_{\text{task}}$ contains task-level tokens.

**Bottom-Up Feature Extraction.** The bottom-up path provides visual saliency for gaze prediction. We use V-JEPA 2 ViT-L [1] as the pretrained video encoder. We resize each frame ($T$ in total) to size of $256\times256$. With a patch size of $N = 16\times16$, each frame produces a $16\times16$ grid of spatial tokens. A trainable linear layer projects each token to a width $d = 256$:

$$\mathbf{c}_v = \mathbf{V} \in \mathbb{R}^{T\times N\times d}, \qquad N = 16 \times 16. \tag{2}$$

Each gaze token $\mathbf{h}_t$ (see Section 3.2) is allowed to cross-attend to the spatial tokens from the same frame $t$. This biases $\mathbf{c}_v$ toward frame-specific evidence. Modeling the temporal gaze correlation is handled by the velocity network (Section 3.2).

**Top-Down Task Token Bank.** Frame-level visual tokens alone do not encode task-level intent over the whole prediction horizon. We therefore aggregate the video tokens into a small bank of task tokens. This module provides the top-down task-level signal in GAZEFLOW.

We first average the video tokens over the spatial axis to obtain one summary $\bar{\mathbf{V}}_t$ per frame. We then average the frame summaries over time to obtain one global video token $\mathbf{p}$. This single token is stable, but it has limited representational power and could miss local task information. We therefore add a learnable-query mechanism based on multi-head cross-attention (MHCA) [52]. Specifically, we initialize $M$ learnable task queries $\mathbf{Q}_{\text{task}}$ attend to the $T$ frame summaries:

$$\bar{\mathbf{V}}_t = \frac{1}{N}\sum_{n=1}^{N} \mathbf{V}_{t,n,:}, \qquad \mathbf{p} = \frac{1}{T}\sum_{t=1}^{T} \bar{\mathbf{V}}_t, \qquad \mathbf{q}_{1:M} = \mathrm{MHCA}\big(\mathbf{Q}_{\text{task}}, \bar{\mathbf{V}}\big). \tag{3}$$

Here, $n$ indexes the $N$ spatial tokens within frame $t$. $\bar{\mathbf{V}} = [\bar{\mathbf{V}}_1; \ldots; \bar{\mathbf{V}}_T]$ stacks all the frame summaries. The global token $\mathbf{p}$ captures coarse task context while the queries $\mathbf{q}_{1:M}$ allow for more fine-grained, selective extraction of the task information from the video. The final task-token bank is

$$\mathbf{c}_{\text{task}} = [\mathbf{p}; \mathbf{q}_{1:M}] \in \mathbb{R}^{(M+1)\times d}. \tag{4}$$

We verify this design in three ways. Section 4.3 shows that the task tokens $\mathbf{c}_{\text{task}}$ encode task-level semantics. Section 4.5 shows that the learnable queries $\mathbf{q}_{1:M}$ strengthen task encoding and improve prediction compared with a single global summary. Section 4.4 shows that task information improves gaze prediction. Appendix C gives all the details.

### 3.2 Conditional Velocity Network

We now describe how the bottom-up embeddings $\mathbf{c}_{\text{v}}$ and top-down embeddings $\mathbf{c}_{\text{task}}$ are used to help generate gaze trajectories. In particular, we use conditional flow matching [37, 50], where generation is a transport process. A sample starts as a Gaussian noise $\mathbf{x}_0 \in \mathbb{R}^{2T}$, where $T$ is the number of discrete points (frames) on the gaze trajectory we predict (each of the discrete point has two coordinates, hence $2T$ dimensions). It then passes through noisy trajectory states $\mathbf{x}_s$, and eventually reaches a clean trajectory sample $\mathbf{x}_1$. This transport is realized by the following differential equation:

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \frac{d\mathbf{x}_s}{ds} = \mathbf{v}_\theta(\mathbf{x}_s, s, \mathbf{c}), \quad \hat{\mathbf{g}} = \mathbf{x}_1 = \int_0^1 \mathbf{v}_\theta(\mathbf{x}_s, s, \mathbf{c})\mathrm{d}s + \mathbf{x}_0 \sim q_\theta(\mathbf{g} \mid \mathcal{V}). \tag{5}$$

Conditioned on $\mathbf{c}_v$ and $\mathbf{c}_{\text{task}}$ (collectively referred as $\mathbf{c}$), the velocity network $\mathbf{v}_\theta$ decides how the current state should move at flow time $s$. $\mathbf{v}_\theta(\cdot)$ takes the current state $\mathbf{x}_s$, the flow time $s$, and the conditions $\mathbf{c}$ as input. $\mathbf{x}_s$ and $s$ are first embedded into $T$ gaze tokens, each of which corresponds to the gaze at frame $t$. The gaze tokens are then updated in each denoising step by four separate blocks, which are visualized in Figure 2(b).

First, **time-aware self-attention** lets gaze tokens exchange information over time. This is where temporal correlation of gaze positions in the trajectory can be captured. We use 1D RoPE to encode relative frame offsets [48]. Second, **per-frame visual cross-attention** lets each gaze token read the spatial visual tokens of the same frame from $\mathbf{c}_v$. This injects bottom-up visual saliency. Third, **task cross-attention** lets all gaze tokens read the task-token bank $\mathbf{c}_{\text{task}}$. This injects top-down, task-specific information. Fourth, a **feed-forward** layer transform each token independently. We implement the velocity network with DiT-style blocks [43]. Appendix C gives the full design details. Finally, a linear output head maps the latent gaze tokens to velocity in the trajectory space $\mathbb{R}^{2T}$.

At inference time, the integral in Eq. 5 is evaluated numerically with a set of Euler steps, each of which evaluates $\mathbf{v}_\theta(\cdot)$ once. Each integral evaluation uses a noise sample $\mathbf{x}_0$ and gives a $\mathbf{x}_1$, representing one sample of the underlying distribution of the gaze trajectory $q_\theta(\mathbf{g} \mid \mathcal{V})$. Repeating this integration with different noise samples gives a sample distribution of the gaze trajectory.

### 3.3 Training & Inference

**Training.** We train GAZEFLOW on 64-frame clips paired with per-frame gaze. For each clip, we sample $s \sim \mathcal{U}[0,1]$ and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{2T})$, and form $\mathbf{x}_s = (1-s)\mathbf{x}_0 + s\mathbf{g}$ with a sample $\mathbf{g} \sim p^\star(\cdot \mid \mathcal{V})$ given by the dataset. The CFM loss regresses $\mathbf{v}_\theta$ on the per-sample velocity $\mathbf{g} - \mathbf{x}_0$:

$$\mathcal{L}_{\text{CFM}}(\theta) \;=\; \mathbb{E}_{\mathcal{V},\, s,\, \mathbf{x}_0,\, \mathbf{g}}\Big[\, \big\| \mathbf{v}_\theta(\mathbf{x}_s,\, s,\, \mathbf{c}) - (\mathbf{g} - \mathbf{x}_0) \big\|_2^2 \Big]. \tag{6}$$

See Appendix D.1 and Appendix D.2 for the full training algorithm and engineering details.

**Long-Video Inference.** We process videos longer than 64 frames with overlapping windows. Each window is encoded once for $(\mathbf{c}_v, \mathbf{c}_{\text{task}})$. We integrate Eq. 5 via Euler steps in parallel with multiple noise samples; each endpoint $\mathbf{x}_1$ is one trajectory sample for that window. For adjacent windows, we linearly blend predictions in the overlap region by frame index. This keeps transitions continuous and avoids encoding the full video at once. Appendix D.3 gives the exact windowing and blending rule.

# 4 Experiments

## 4.1 Experimental Setup

**Training and Inference.** We optimize both the video encoder and the velocity network jointly in an end-to-end manner. The encoder is adapted via LoRA, applying rank-16 low-rank [16] updates to its query and value projections, while the velocity network is trained with full parameters. The model is trained by minimizing the conditional flow-matching objective $\mathcal{L}_{\text{CFM}}$ in Eq. 6 using AdamW [39] ($\beta_1$=0.9, $\beta_2$=0.999, weight decay $10^{-4}$), for 80 epochs with a base learning rate of $10^{-4}$ under a cosine schedule decaying to $10^{-6}$. All experiments are run on 4×NVIDIA H100 GPUs in bfloat16 mixed precision. Remaining hyperparameters and configurations are summarized in Appendix E.4. At inference, we draw $K$=50 trajectory samples per clip in parallel as a Monte Carlo of the learned joint distribution. Each sample is generated by integrating the learned velocity field $\mathbf{v}_\theta$ using the explicit Euler scheme over $S$=50 uniform steps, starting from an independent draw of $\mathcal{N}(\mathbf{0}, \mathbf{I})$. It takes ∼3.4 s to inference a clip of $T$=64 frames on a single H100 (Appendix F.2). With $K$=30 and $S$=5, inference is 4.6× faster with no loss in accuracy (Appendix F.2).

**Per-Frame Heatmaps.** At each frame, we convert the $K$=50 sampled gaze points into a per-frame heatmap by placing a Gaussian kernel at each point and averaging. This makes GazeFlow directly comparable to heatmap-output baselines under the heatmap-based per-frame metrics defined below.

**Datasets.** We evaluate on two egocentric gaze datasets. EGTEA Gaze+ [33] contains cooking videos recorded at 640 × 480 resolution and 24 fps, with eye-tracker gaze annotations, and provides 8,299 training and 2,022 validation clips. Ego4D [12] spans a broader range of daily activities; we use the subset with gaze annotations, recorded at 1088 × 1080 resolution and 30 fps, with 12,178 training and 5,202 validation clips. Data preparation details are provided in Appendix E.1.

**Baselines.** We compare against three families of prior methods. For the first two families, we include representative state-of-the-art models.

- *Marginal density estimation.* GLC [26], a state-of-the-art egocentric gaze model that outputs a per-frame fixation heatmap independently for each frame.
- *Autoregressive.* AT [17], the first egocentric gaze model to learn attention transitions using an LSTM conditioned on previous-frame gaze, and EgoM2P [31], a multimodal model pretrained on large-scale egocentric datasets that decodes gaze as a discrete autoregressive token sequence.
- *Simple Priors.* Center Bias always predicts the gaze position as the image center; Random Walk is a Gaussian random walk whose step variance is set to the human frame-to-frame gaze motion calculated from the dataset.

We retrain AT, use the official checkpoint for GLC, and evaluate EgoM2P in two settings: the official zero-shot checkpoint and LoRA fine-tuning. All baselines are evaluated on both datasets, with full configuration details provided in Appendix E.2.

**Metrics.** We use two commonly used metrics: per-frame prediction accuracy and alignment with human temporal dynamics. The first asks whether gaze is predicted at the correct spatial location on each frame. The second asks whether the predicted trajectory has a human-like temporal structure. Full definitions, interpretations, and computation details of all metrics are provided in Appendix E.3.

**(1) Per-Frame Accuracy.** We report the *Area Under the Curve* (AUC) [5, 21], *F1* [26], *Precision* (P) [26], *Recall* (R) [26], and *Average Angular Error* (AAE) [17].

**(2) Temporal Dynamics Alignment.** We compare each predicted trajectory against the human trajectory using the *Average Displacement Error* (ADE) [23] and *Dynamic Time Warping distance* (DTW) [44, 23], each reported as mean and best-of-$K$ over $K$=50 samples. Additionally, we report the mean and median *per-frame displacement* as ratios to the human reference, and the *Jensen–Shannon divergence* (JSD) [9] between the predicted and human displacement histograms.

Table 2: **Per-frame accuracy on EGTEA Gaze+ and Ego4D.** GAZEFLOW achieves the best performance; results on more metrics are in Table 9 in Appendix F.1. Best per column is in bold.

| | | EGTEA Gaze+ | | | | | Ego4D | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Paradigm | Method | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ |
| *Simple Priors* | Center Bias | 0.906 | 0.185 | 0.162 | 0.215 | 16.07 | 0.930 | 0.222 | 0.194 | 0.258 | 13.92 |
| | Random Walk | 0.824 | 0.147 | 0.090 | 0.413 | 16.34 | 0.763 | 0.109 | 0.059 | **0.662** | 14.38 |
| *Marginal density* | GLC [26] | 0.953 | 0.421 | 0.348 | 0.531 | 10.21 | 0.947 | 0.355 | 0.267 | 0.530 | 11.52 |
| *Autoregressive* | EgoM2P (zero-shot) [31] | 0.921 | 0.280 | 0.195 | 0.498 | 13.18 | 0.911 | 0.253 | 0.180 | 0.422 | 13.86 |
| | EgoM2P (LoRA) [31] | 0.921 | 0.293 | 0.205 | 0.515 | 12.81 | 0.930 | 0.297 | 0.217 | 0.470 | 12.69 |
| | AT [17] | 0.956 | 0.419 | 0.339 | 0.547 | 9.88 | 0.954 | 0.346 | 0.268 | 0.487 | 12.01 |
| *Distribution Mapping* | **GAZEFLOW (ours)** | **0.964** | **0.491** | **0.414** | **0.603** | **9.01** | **0.958** | **0.392** | **0.307** | 0.540 | **10.46** |

Table 3: **Alignment with human gaze dynamics on EGTEA Gaze+.** Disp Mean and Disp Med are per-step gaze displacements shown as ratios to the human trajectory (subheader H, measured in pixels; closer to $1\times$ is better). Best per column in bold.

| | | ADE ($\times 10^2$ px)↓ | | DTW ($\times 10^2$ px)↓ | | Disp Mean | Disp Med | JSD↓ |
|---|---|---|---|---|---|---|---|---|
| Paradigm | Method | Mean | Best | Mean | Best | (H: 12.5 px) | (H: 4.0 px) | |
| *Simple Priors* | Center Bias | 1.35 | 1.35 | 1.35 | 1.35 | $0.00\times$ | $0.00\times$ | 0.125 |
| | Random Walk | 2.07 | 1.05 | 1.97 | 0.92 | $2.54\times$ | $7.43\times$ | 0.294 |
| *Marginal density* | GLC [26] | 1.13 | 1.01 | 1.03 | 0.87 | $5.77\times$ | $11.75\times$ | 0.352 |
| *Autoregressive* | EgoM2P (zero-shot) [31] | 1.47 | 0.97 | 1.35 | 0.83 | $2.20\times$ | $3.05\times$ | 0.058 |
| | EgoM2P (LoRA) [31] | 1.50 | 0.96 | 1.37 | 0.81 | $2.69\times$ | $3.03\times$ | 0.058 |
| | AT [17] | 1.32 | 1.12 | 1.22 | 0.99 | $9.70\times$ | $24.55\times$ | 0.476 |
| *Distribution Mapping* | **GAZEFLOW (ours)** | **1.04** | **0.60** | **0.95** | **0.48** | $\mathbf{0.82\times}$ | $\mathbf{1.08\times}$ | **0.002** |

**Protocol.** All per-frame evaluations follow the GLC-format protocol [26]. For each validation clip, we sample uniformly spaced frames, compute metrics against the ground-truth fixation maps, average over frames within the clip, and then average over clips. The main comparisons in Section 4.2 use the LoRA variant of GAZEFLOW on the full validation split. Ablations in Section 4.4 and design analyses in Section 4.5 use the frozen-encoder variant on a fixed validation subset, since repeating $K{=}50$ LoRA sampling per variant is computationally expensive. Details on subset construction, frozen-encoder evaluation, and sample-count selection are provided in Appendix E.3 and Appendix F.2.

### 4.2 GAZEFLOW Outperforms Prior Methods

Joint distribution modeling achieves the best per-frame accuracy across datasets. Table 2 shows that GAZEFLOW outperforms all baselines on both EGTEA Gaze+ and Ego4D. Compared to the strongest baseline, GAZEFLOW improves F1 by $16.6\%$ and $10.4\%$ on the two datasets, reflecting better binarized fixation prediction under the GLC evaluation protocol. It also reduces AAE by $8.8\%$ and $9.2\%$ on the two datasets, corresponding to lower angular error of the predicted fixation with respect to human gaze. Table 9 shows the full breakdown.

GAZEFLOW also matches human gaze dynamics much more closely than prior methods. Table 3 reports the alignment-with-human-dynamics metrics. GAZEFLOW achieves the lowest mean and best-of-$K$ ADE and DTW, indicating smaller path-wise error to the ground-truth trajectory under Euclidean and temporally aligned comparisons, respectively. On best-of-$K$, GAZEFLOW reduces this error by roughly $1.6\times$ compared to the strongest baseline, EgoM2P with LoRA, with ADE decreasing from $0.96$ to $0.60$ and DTW from $0.81$ to $0.48$. Our per-step displacement statistics are also closest to the human reference: the median is $1.08\times$ human and the mean follows the same pattern, whereas GLC and AT are $11.75\times$ and $24.55\times$, respectively. The displacement JSD is $0.002$, the lowest among all methods and closest to the human reference distribution.

### 4.3 Task Tokens Encode Task Semantics

We have hypothesized that task information can be beneficial to gaze prediction, since human gazing is task dependent. This hypothesis motivates us to extract the task-dependent tokens $\mathbf{c}_{\text{task}}$, which are used to condition the flow matching. We now show that $\mathbf{c}_{\text{task}}$ does carry task-level semantics.

Table 4: **Linear probing of the task-token bank** on EGTEA action/verb/noun labels. The $\cup$ indicates a clip is counted correct if at least one probe is correct.

| | Action (106) | | Verb (19) | | Noun (53) | |
|---|---|---|---|---|---|---|
| Probe input | T1 | T5 | T1 | T5 | T1 | T5 |
| $\mathbf{q}_{1:M}$ alone | 40.5 | 66.6 | 64.1 | 95.9 | 47.9 | 78.7 |
| $\mathbf{p}$ alone | 49.1 | 76.6 | 68.3 | 98.2 | 58.7 | 87.3 |
| $\mathbf{p} \cup \mathbf{q}_{1:M}$ | **54.6** | **80.2** | **76.0** | **98.7** | **64.4** | **89.8** |

Table 5: **Design alternatives to GAZEFLOW.** *w/o Learnable Queries*: drop the $M{=}4$ learnable queries (keep only the global mean-pool token). *Concat*: replace task cross-attention with channel-wise concatenation. *Learnable PE*: replace RoPE with a learnable absolute position embedding.

| Variant | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ |
|---|---|---|---|---|---|
| **Default** | **0.970** | **0.493** | **0.401** | **0.638** | **8.81** |
| w/o Learnable Queries | 0.967 | 0.484 | 0.395 | 0.626 | 8.93 |
| Concat | 0.966 | 0.469 | 0.379 | 0.615 | 9.10 |
| Learnable PE | 0.967 | 0.476 | 0.380 | 0.635 | 8.94 |

Each video clip in EGTEA Gaze+ is annotated at three label hierarchies: 19 verbs (e.g., "cut"), 53 nouns (e.g., "tomato"), and 106 actions formed as verb-noun pairs (e.g., "cut tomato"). We freeze GAZEFLOW, extract its task-token bank for each clip, and train a linear classifier to predict the action, verb, and noun labels. We report Top-$k$ accuracy, which denotes the percentage of validation clips for which the ground-truth class appears among the linear classifier's $k$ highest-scoring predictions. As shown in Table 4, the global token $\mathbf{p}$ is already strongly predictive, reaching $49.1\%$, $68.3\%$, and $58.7\%$ Top-1 accuracy on action, verb, and noun, respectively. The learnable task queries $\mathbf{q}_{1:M}$ are weaker in isolation, but provide complementary information: combining $\mathbf{p}$ and $\mathbf{q}_{1:M}$ improves Top-1 by $+5.5$, $+7.7$, and $+5.7$ points. These results show that task-relevant semantics emerge in the task-token bank despite the absence of explicit supervision, consistent with the design in Eq. 4.

### 4.4 Ablation Studies

We isolate the contribution of each architectural component by removing one component at a time, as shown in Table 6. The four ablation rows test the four architectural choices made in Section 3: joint-distribution modeling via flow matching, the bottom-up pathway, the top-down pathway, and 1D temporal RoPE for time-aware self-attention (SA).

**Flow Matching vs. Regression.** Replacing the flow-matching head with a deterministic regression head over the same encoder and conditioning inputs reduces AUC from $0.970$ to $0.903$ and inflates the displacement JSD against human gaze by roughly $66\times$. This is the most direct evidence for the central claim of Section 3.2: gaze should be modeled as a joint distribution over the full trajectory. A regression head, which neither models a distribution nor couples frames, predicts each gaze point in isolation and therefore fails to capture the temporal coupling between frames.

**Flow Matching vs. Diffusion.** We also replace the flow-matching objective with a diffusion objective, keeping the architecture, conditioning, and training setup the same. At $50$ sampling steps, flow matching reaches F1 $0.493$ and diffusion $0.485$. The gap grows as the step count drops: at $5$ steps, flow matching reaches F1 $0.506$ while diffusion drops to $0.359$. Flow matching thus needs fewer steps and less inference time for the same accuracy (Appendix F.4).

**Dual-Path Conditioning.** Dropping the per-frame visual cross-attention sub-layer of the velocity network, which carries the bottom-up condition $\mathbf{c}_v$ from Section 3.1, lowers F1 from $0.493$ to $0.431$. Dropping the task cross-attention sub-layer, which carries the top-down condition $\mathbf{c}_{\text{task}}$, lowers F1 to $0.476$. Both pathways are individually necessary, supporting the dual-pathway design in Section 3.1.

**Temporal RoPE.** Removing the relative-time RoPE from the time-aware self-attention sub-layer of the velocity network has little effect on per-frame accuracy but weakens the displacement JSD by roughly $2.5\times$ and worsens the median displacement to $1.4\times$ of that of human. This confirms that the relative-offset encoding is what enables time-aware self-attention to capture inter-frame temporal coupling, as claimed in Section 3.2.

### 4.5 Design Choice Analysis

**Temporal Gaze Correlation.** Time-aware gaze-token self-attention is where our denoise block models the temporal coupling needed for joint-distribution modeling, and its temporal positional encoding decides how each gaze token reasons about its position relative to other frames. We compare two choices: *(i)* RoPE (default, details in Appendix C.2), which injects each pairwise frame offset

Table 6: **Ablation study. Full** is the default GAZEFLOW; each remaining row ablates one architectural component while keeping all others identical, removing either the bottom-up visual feature ($\mathbf{c}_v$, frame-local visual cross-attention), the top-down task guidance ($\mathbf{c}_{\text{task}}$, task cross-attention), the relative-time RoPE in time-aware self-attention, or the joint-distribution flow-matching head (replaced with a deterministic regression head over the same conditioning). Per-column best in **bold**.

| Variant | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ | ADE↓ | DTW↓ | Disp Mean (H: 12.5 px) | Disp Med (H: 4.0 px) | JSD↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Full** | **0.970** | **0.493** | **0.401** | **0.638** | **8.81** | **0.57** | **0.45** | **0.91×** | **1.08×** | **0.0011** |
| - Bottom-up Visual Feature | 0.961 | 0.431 | 0.333 | 0.613 | 9.22 | 0.65 | 0.53 | 0.89× | 1.17× | 0.0012 |
| - Top-down Task Guidance | 0.965 | 0.476 | 0.380 | 0.638 | 9.14 | 0.58 | 0.46 | 0.92× | 1.09× | 0.0013 |
| - Temporal RoPE | 0.968 | 0.477 | 0.388 | 0.620 | 8.94 | 0.59 | 0.46 | 1.21× | 1.40× | 0.0027 |
| - Joint Distribution Modeling | 0.903 | 0.451 | 0.394 | 0.526 | 9.12 | 0.84 | 0.79 | 0.71× | 2.01× | 0.0729 |

directly into the attention dot product, and *(ii)* a learnable absolute position embedding, which assigns a learned vector to each frame index. Replacing RoPE with the absolute embedding lowers F1 from 0.493 to 0.476 (Table 5). This is because gaze dynamics are properties of how far apart two frames are, not of where a frame sits inside an arbitrary clipped window.

**Top-Down Task Guidance.** Top-down task information from the task-token bank is injected into gaze tokens via a per-block task cross-attention sub-layer. We compare two forms of task guidance: *(i)* cross-attention (default; Appendix C.3), where each gaze token queries the bank at every denoiser block, and *(ii)* channel-wise concatenation of the bank to the gaze-token input. Concatenation degrades all metrics, including F1 ($0.493 \rightarrow 0.469$) and AAE ($8.81 \rightarrow 9.10$); see Table 5 for details.

Removing the $M{=}4$ learnable queries, leaving only the global mean-pool token, also reduces F1 to 0.484 (Table 5), confirming that the queries enable task-aware selective extraction beyond the low-bandwidth global token alone.

**Task-Token Bank Bandwidth.** We sweep $M \in \{1, 2, 4, 8, 16\}$, the number of learnable task queries, and the results are shown in Table 13 in Appendix F.3. The performance peaks at $M = 4$, indicating that too few queries cannot represent the diversity of task-relevant factors, while too many dilute the per-query specialization that cross-attention relies on.

### 4.6 Interpretations

**Interpreting Self-Attention.** Self-attention in our velocity network decides, when predicting gaze at frame $i$, how strongly to draw on gaze at each other frame $j$. For each head we measure how attention mass distributes over $j$ and find two regimes: *look-back* heads concentrate on past frames and *look-ahead* heads concentrate on future frames. This pattern parallels two well-known modes of human gaze in egocentric tasks — anticipating upcoming events [41, 29] and lagging ongoing ones to verify completion [3, 13]. A third *split-attention* category and full visualizations are in Appendix A.1.

**Interpreting Bottom-Up Saliency.** The visual cross-attention sub-layer provides bottom-up guidance that decides, when predicting gaze at frame $t$, which spatial regions of the video to focus on. Using off-the-shelf hand and active-object segmentation algorithm [38], we measure how much attention mass each attention head places on the hand, the active object, and the rest of the scene. We find that the heads cluster into hand-focused, object-focused, and coupled groups, and the cluster also varies with the action being performed. This finding is consistent with the manipulation literature, in which gaze concentrates on the hands and the objects they act on [28, 13]. Details are in Appendix A.2.

### 4.7 Limitations

GAZEFLOW's failures concentrate on clips with saccades whose endpoint lies outside the camera's field of view: the saccade target was never captured by the recorded video on which the model conditions, so there is no visual evidence for us to make the prediction (Appendix A.3).

GAZEFLOW models the conditional distribution of gaze trajectories given a fixed input video, so the head movement that produced that video is part of the conditioning. The gaze distribution we recover is therefore of the residual eye movement, not the full distribution that would arise across observers

freely interacting with the same scene, where head movement itself varies across trials [55]. We leave to future work to jointly model head and eye movements.

Additionally, GAZEFLOW's prediction is also conditioned upon the *entire* video sequence, which limits its applicability to scenarios that require real-time gaze prediction from egocentric videos such as gaze-contingent rendering [34, 6, 8, 35], where one has no access to future frames. We also train a causal variant that sees only past and current frames; it enables online gaze prediction at some cost in accuracy (details in Appendix F.4).

## 5 Conclusion

GAZEFLOW is designed to account for the three factors that influence the stochasticity of human gaze: structured temporal dynamics, top-down task influence, and bottom-up scene saliency. GAZEFLOW sets a new state of the art on per-frame prediction accuracy while simultaneously achieving the closest alignment with human gaze dynamics. We hope GAZEFLOW can broaden the use of gaze in applications that have traditionally required dedicated eye-tracking hardware.

## Acknowledgments and Disclosure of Funding

Funding: The authors received no third-party funding or support for this work. Competing interests: The authors declare no competing interests.

## References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Dana H. Ballard, Mary M. Hayhoe, and Jeff B. Pelz. Memory representations in natural tasks. *Journal of Cognitive Neuroscience*, 7:66–80, 1995.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757, 2019.

[6] Ethan Chen, Sushant Kondguli, Carl Marshall, and Yuhao Zhu. Modeling and exploiting the time course of chromatic adaptation for display power optimizations in virtual reality. *ACM Transactions on Graphics (TOG)*, 44(6):1–21, 2025.

[7] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[8] Budmonde Duinkharjav, Kenneth Chen, Abhishek Tyagi, Jiayi He, Yuhao Zhu, and Qi Sun. Color-perception-guided display power reduction for virtual reality. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022.

[9] Bent Fuglede and Flemming Topsøe. Jensen-shannon divergence and hilbert space embedding. *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, page 31, 2004.

[10] Wilson S Geisler and Jeffrey S Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In *Human vision and electronic imaging III*, volume 3299, pages 294–305. SPIE, 1998.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[13] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 9(4):188–194, 2005.

[14] John M Henderson. Gaze control as prediction. *Trends in cognitive sciences*, 21(1):15–23, 2017.

[15] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[17] Yifei Huang, Minjie Cai, Zhenqiang Li, and Yoichi Sato. Predicting gaze in egocentric video by learning task-dependent attention transition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 754–769, 2018.

[18] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE transactions on image processing*, 13(10):1304–1318, 2004.

[19] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10-12):1489–1506, 2000.

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

[21] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. 2012.

[22] Anton S Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo. Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[23] Ozgur Kara, Harris Nisar, and James Matthew Rehg. Diffeye: Diffusion-based continuous eye-tracking data generation conditioned on natural images. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[24] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Gaze-based dual resolution deep imitation learning for high-precision dexterous robot manipulation. *IEEE Robotics and Automation Letters*, 6:1630–1637, 2021.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] Bolin Lai, Miao Liu, Fiona Ryan, and James M Rehg. In the eye of transformer: Global–local correlation for egocentric gaze estimation and beyond. *International Journal of Computer Vision*, 132(3):854–871, 2024.

[27] Bolin Lai, Fiona Ryan, Wenqi Jia, Miao Liu, and James M Rehg. Listen to look into the future: Audio-visual egocentric gaze anticipation. In *European Conference on Computer Vision*, pages 192–210. Springer, 2024.

[28] Michael Land, Neil Mennie, and Jennifer Rusted. The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28(11):1311–1328, 1999.

[29] Michael F Land and Mary Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision research*, 41(25-26):3559–3565, 2001.

[30] Yue Leng, Chi-Chun Chen, Qiuyue Sun, Jian Huang, and Yuhao Zhu. Energy-efficient video processing for virtual reality. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 91–103, 2019.

[31] Gen Li, Yutong Chen, Yiqian Wu, Kaifeng Zhao, Marc Pollefeys, and Siyu Tang. Egom2p: Egocentric multimodal multitask pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10830–10843, 2025.

[32] Yin Li, Alireza Fathi, and James M. Rehg. Learning to predict gaze in egocentric video. *2013 IEEE International Conference on Computer Vision*, pages 3216–3223, 2013.

[33] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018.

[34] Weikai Lin, Yu Feng, and Yuhao Zhu. Metasapiens: Real-time neural rendering with efficiency-aware pruning and accelerated foveated rendering. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pages 669–682, 2025.

[35] Weikai Lin, Sushant Kondguli, Carl Marshall, and Yuhao Zhu. Powergs: Display-rendering power co-optimization for neural rendering in power-constrained xr systems. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.

[36] Weikai Lin, Sheng Zhao, Ian Ross, Carl Marshall, Sushant Kondguli, and Yuhao Zhu. Lowpowar: Power-constrained tone mapping for augmented reality. *arXiv preprint arXiv:2607.19509*, 2026.

[37] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[38] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[40] Amrita Mazumdar, Brandon Haynes, Magda Balazinska, Luis Ceze, Alvin Cheung, and Mark Oskin. Perceptual compression for video storage and processing systems. In *Proceedings of the ACM symposium on cloud computing*, pages 179–192, 2019.

[41] Neil Mennie, Mary M. Hayhoe, and Brian T. Sullivan. Look-ahead fixations: anticipatory eye movements in natural tasks. *Experimental Brain Research*, 179:427–442, 2007.

[42] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver's focus of attention: The dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018.

[43] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[44] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, 2009.

[45] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya K. Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao Wu, Ross B. Girshick, Piotr Doll'ar, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *ArXiv*, abs/2408.00714, 2024.

[46] Constantin A Rothkopf, Dana H Ballard, and Mary M Hayhoe. Task and context determine where you look. *Journal of vision*, 7(14):16–16, 2007.

[47] Michele Rucci, Ehud Ahissar, and David Burr. Temporal coding of visual space. *Trends in cognitive sciences*, 22(10):883–895, 2018.

[48] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[49] Benjamin W Tatler, Mary M Hayhoe, Michael F Land, and Dana H Ballard. Eye guidance in natural vision: Reinterpreting salience. *Journal of vision*, 11(5):5–5, 2011.

[50] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*.

[51] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

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

[53] Alfred L Yarbus. Eye movements and vision. *Eye movements and vision.*, page 171, 1967.

[54] Mengmi Zhang, Keng Teck Ma, Joo Hwee Lim, Qi Zhao, and Jiashi Feng. Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3548, 2017.

[55] Sheng Zhao, Junrui Zhu, Shuning Zhang, Xueyang Wang, Hongyi Li, Fang Yi, Xin Yi, and Hewu Li. Coordauth: Hands-free two-factor authentication in virtual reality leveraging head-eye coordination. In *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 738–748. IEEE, 2025.

[56] Sheng Zhao, Weikai Lin, and Yuhao Zhu. Multidimensional observer model and perceptual dimensions of human image quality assessment. *arXiv preprint arXiv:2609.38487*, 2026.

# A Additional Visualization Results

This appendix gives the figures behind the analyses in Section 4.6 and the failure cases in Section 4.7.

## A.1 Per-Head Patterns of the Time-aware Self-Attention

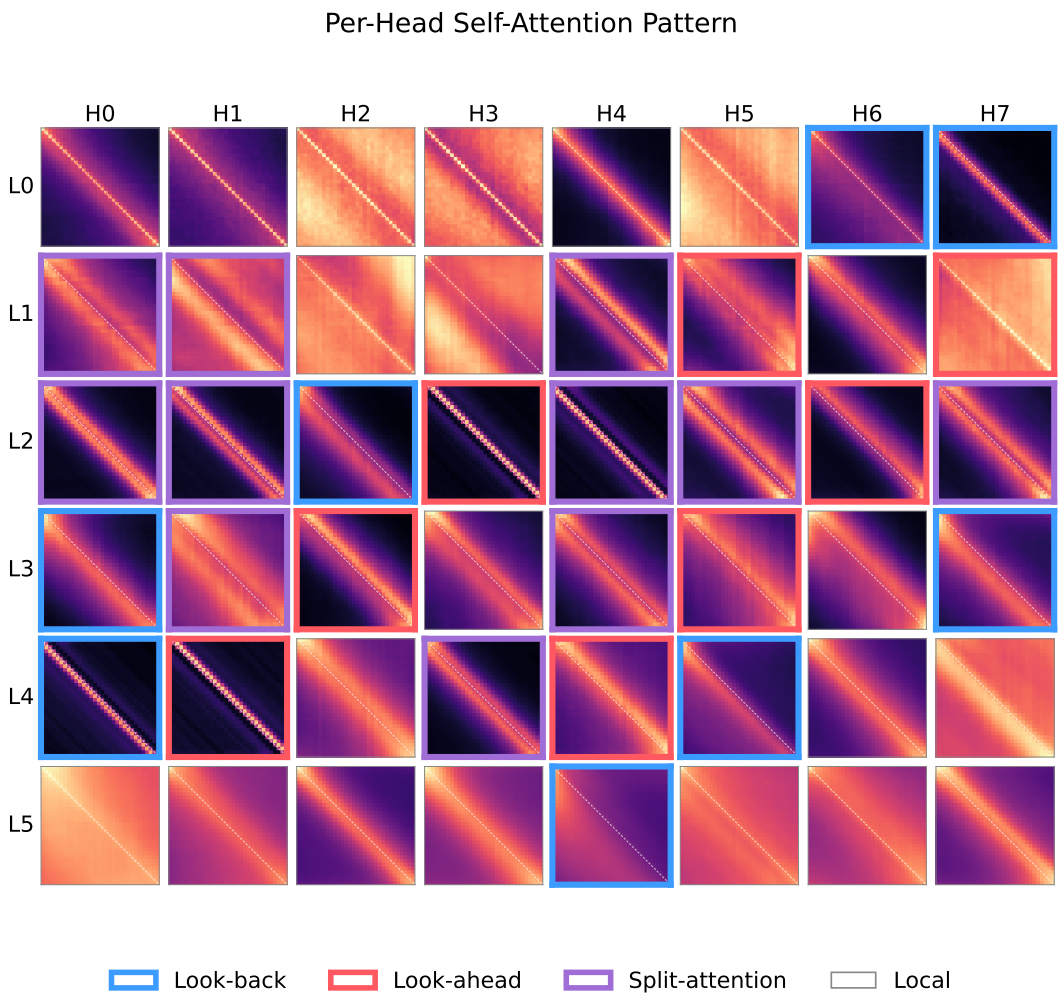



Figure. 3: **Per-head patterns of the time-aware self-attention aggregated across the EGTEA Gaze+ validation set.** Rows index denoiser layers, columns index attention heads. Each cell shows the canonical attention pattern (query frame on the vertical axis, source frame on the horizontal, both normalized to clip-relative position); brighter pixels correspond to higher attention weight after row-wise softmax. The dashed diagonal marks *source* = *query*, separating two halves with opposite temporal meaning: mass in the *lower triangle* (*source* < *query*) is attention to *past* frames (look-back), mass in the *upper triangle* (*source* > *query*) is attention to *future* frames (look-ahead), and mass on both sides marks split-attention heads with peaks straddling the current frame. Cell-frame color encodes this role: look-back (blue), look-ahead (red), split-attention (purple), local (grey).

These two figures support the look-back and look-ahead heads described in Section 4.6. We forward EGTEA Gaze+ validation clips through the trained main model at the midpoint of denoising and capture the gaze self-attention matrix of every (layer, head). Because clip length varies, each matrix is resized to a common grid where both axes are normalized to clip-relative position; the per-head pattern shown is the average across clips. We classify each head from its 1-D attention-vs-offset curve into one of four temporal roles: *look-back*, *look-ahead*, *split-attention* (an inverted-M shape with peaks on both sides of the current frame), or *local*; the same color scheme is used in both figures

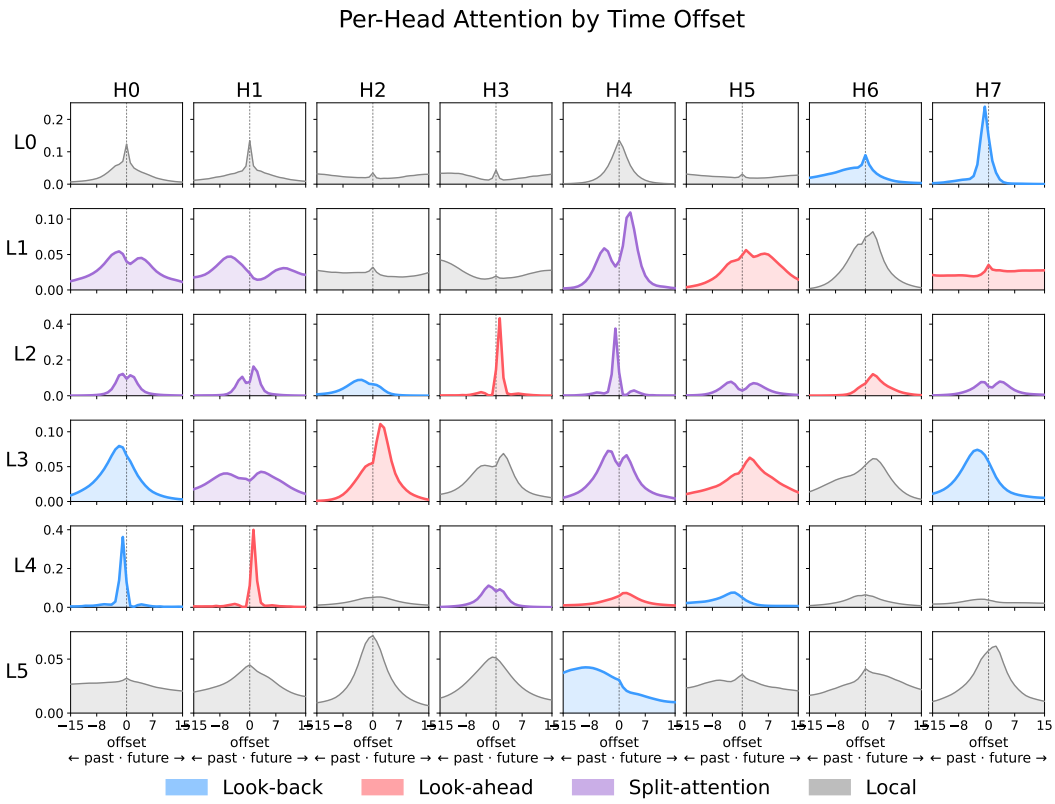



Figure. 4: **Per-head time-aware self-attention as a function of temporal offset.** Same heads as Fig. 3, summarized as a 1-D curve by averaging the attention pattern along each diagonal. The horizontal axis is the offset between source and query frame (negative = past, positive = future); the vertical dashed line marks zero offset. Look-ahead heads peak right of zero, look-back heads peak left, and split heads exhibit two peaks straddling zero with a clear valley at the current frame.

and the legend. Figure 3 shows the full 2-D pattern per head: look-back heads light up the lower triangle, look-ahead heads light up the upper triangle, and split heads light up both triangles. Figure 4 collapses the same data to a 1-D temporal-offset curve, where the three role types are immediately distinguishable from peak position.

### A.2 Per-Head Semantic Attribution

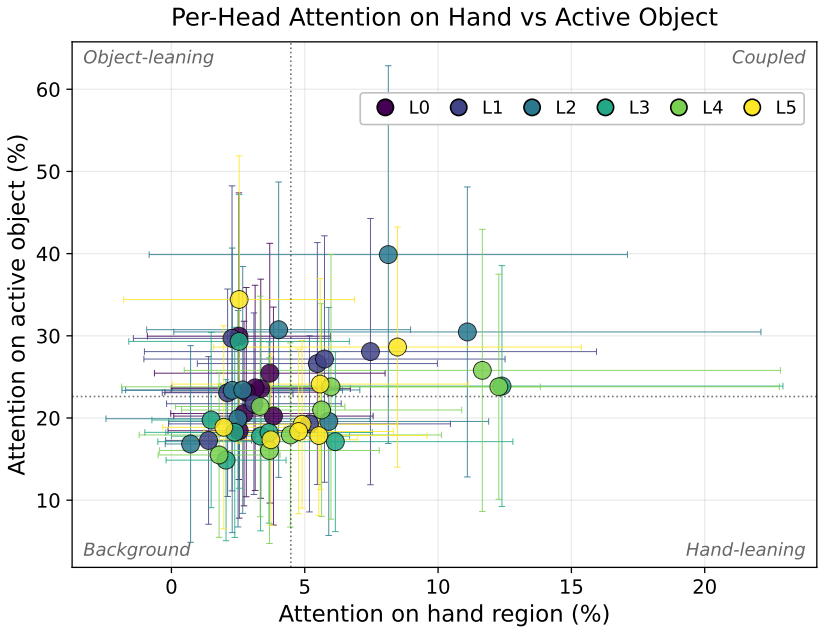



Figure. 5: **Hand vs. active-object attention mass per (layer, head)** on the EGTEA Gaze+ validation set. Each marker is one of the 48 visual-cross-attention heads, plotted by mean attention mass on the hand (x-axis) and active-object (y-axis) regions; markers are colored by layer, error bars are clip-wise standard deviation. Dotted lines mark the global means (hand $= 4.5\%$, object $= 22.6\%$). Heads in the lower-right quadrant focus on the hand, those in the upper-left on the active object, and those in the upper-right couple both.

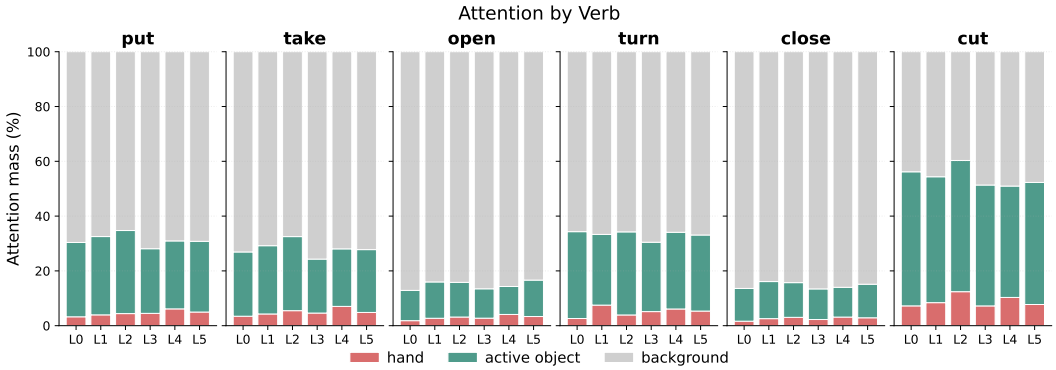



Figure. 6: **Layer-wise hand / active-object / background attention mass conditioned on verb.** For each of the six most frequent EGTEA verbs we average the per-region attention mass across heads at each layer. Manipulation verbs ("cut", "put", "take", "turn") keep 25–50% of the mass on the active object at every layer, while "open" and "close" route most mass to the background regardless of layer.

These two figures support the hand-focused and object-focused heads described in Section 4.6. For each clip in the EGTEA Gaze+ validation set we extract per-frame Hand and Active-Object masks via GroundingDINO [38] and SAM 2 [45], downsampled to the $16\times16$ V-JEPA 2 token grid. We then run a forward pass of the trained main model and capture the per-(layer, head) attention weights of the per-frame visual cross-attention sub-layer (Section 3.2) over the spatial tokens. For each clip, layer, and head we sum the attention mass inside the hand mask, inside the active-object mask, and outside both (background); the three values are normalized to 1 per (clip, layer, head) and then averaged

across clips. Figure 5 plots each head as one point in the (hand, object) attention plane: heads in the lower-right quadrant concentrate on the hand, those in the upper-left on the active object, and those in the upper-right couple both, all relative to the global means (hand $= 4.5\%$, object $= 22.6\%$). Figure 6 groups clips by the first verb of their EGTEA action label, and the pattern differs across verbs. "cut", "put", "take", and "turn" act on an object already in hand, and the heads keep 25–50% of the mass on the active object across all six layers. "open" and "close" act on a fixture such as a drawer, lid, or container, and the heads route most of the mass to the background regardless of layer.

### A.3 Failure Case Analysis

GAZEFLOW's failure cases concentrate on clips containing saccades whose endpoint lies outside the camera's field of view (Figure 7). A large saccade traverses tens of visual degrees in 30–80 ms [47], so at our 24–30 fps recording rate the eye can leave the captured frame within a single sample. Once the eye target is outside the camera field of view (FOV), the model has no visual evidence on which to ground a prediction: GAZEFLOW is conditioned solely on the recorded video, and the saccade endpoint by definition was never captured by it. The predicted trajectories therefore stay on objects that remain visible, while the eye tracker reports a gaze position that is clamped to the image boundary. Closing this gap would require a wider-FOV camera that captures the eventual saccade target, or a saccade-aware model that recognizes when the gaze has left the camera view, both of which we leave to future work.




Figure. 7: **Failure cases.** Three rows from two validation clips, each row showing 8 evenly-spaced frames along time. Cyan dots are the $K{=}50$ GAZEFLOW predictions at that frame; the filled red circle is the recorded ground-truth gaze when it lies within the frame, and the hollow red ring with a yellow halo flags frames where the recording is clamped to the image boundary (off-frame). When the wearer saccades to a target outside the camera's view, the predicted cloud stays on the visible scene while the ground-truth marker drifts to the boundary.

## B Conditional Flow Matching: Derivation

This appendix shows that the conditional flow matching loss $\mathcal{L}_{\mathrm{CFM}}$ used in Section 3.3 (Eq. 6) is a sample-based $L^2$ regression on a velocity field whose global minimum yields, at the population level, $q_\theta = p^\star$. Section B.1 shows that minimizing $\mathcal{L}_{\mathrm{CFM}}$ recovers $p^\star$, and Section B.2 shows that per-frame factorization cannot recover the joint distribution.

### B.1 Why $\mathcal{L}_{\mathrm{CFM}}$ recovers $p^\star$

For a noise sample $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{2T})$ and a trajectory $\mathbf{g} \sim p^\star(\cdot \mid \mathcal{V})$, the linear interpolant

$$\mathbf{x}_s \;=\; (1-s)\,\mathbf{x}_0 + s\,\mathbf{g}, \qquad s \in [0,1], \tag{7}$$

starts at the Gaussian at $s = 0$ and ends at $p^\star$ at $s = 1$. Its per-sample velocity is $\mathbf{g} - \mathbf{x}_0$. Flow matching [37] shows that the marginal velocity field

$$\mathbf{u}^\star_s(\mathbf{x} \mid \mathcal{V}) \;=\; \mathbb{E}\big[\mathbf{g} - \mathbf{x}_0 \,\big|\, \mathbf{x}_s = \mathbf{x},\, \mathcal{V}\big] \tag{8}$$

moves the Gaussian to $p^\star$ along this path. So if $\mathbf{v}_\theta = \mathbf{u}^\star_s$, integrating Eq. 5 from $\mathbf{x}_0$ gives $q_\theta = p^\star$. The loss $\mathcal{L}_{\mathrm{CFM}}$ in Eq. 6 regresses $\mathbf{v}_\theta$ onto $\mathbf{g} - \mathbf{x}_0$. Because $\mathbf{v}_\theta$ sees only $(\mathbf{x}_s, s, \mathbf{c})$, the minimizer of this $L^2$ loss is the conditional expectation above, that is, $\mathbf{u}^\star_s$. Minimizing $\mathcal{L}_{\mathrm{CFM}}$ therefore recovers $p^\star$ at the population level. On EGTEA Gaze+, we multiply the loss by a per-frame validity mask $\mathbf{m}$ that removes blink frames; this does not change the minimizer on valid frames.

### B.2 Per-Frame Factorization Gap

Let $\mathcal{V}$ denote the video and $\mathbf{g} = (\mathbf{g}_1, \ldots, \mathbf{g}_T)$ the gaze trajectory, with ground-truth distribution $p^\star(\mathbf{g} \mid \mathcal{V})$. A marginal model provides the trajectory distribution as

$$q_{\mathrm{marg}}(\mathbf{g} \mid \mathcal{V}) \;=\; \prod_{t=1}^{T} q_t(\mathbf{g}_t \mid \mathcal{V}), \tag{9}$$

where each $q_t$ is the parametric conditional distribution at frame $t$. For any video $\mathcal{V}$, the best possible factorized approximation is obtained when each $q_t$ matches the corresponding true marginal $p^\star(\mathbf{g}_t \mid \mathcal{V})$. Therefore,

$$\begin{aligned} \min_{q_1,\ldots,q_T} D_{\mathrm{KL}}\big(p^\star(\mathbf{g} \mid \mathcal{V}) \,\big\|\, q_{\mathrm{marg}}(\mathbf{g} \mid \mathcal{V})\big) \;&=\; D_{\mathrm{KL}}\Big(p^\star(\mathbf{g} \mid \mathcal{V}) \,\Big\|\, \prod_{t=1}^{T} p^\star(\mathbf{g}_t \mid \mathcal{V})\Big) \\ &=\; \mathrm{TC}(\mathbf{g}_1, \ldots, \mathbf{g}_T \mid \mathcal{V}), \end{aligned} \tag{10}$$

where TC denotes total correlation, defined as the KL divergence between a joint distribution and the product of its marginals. Since KL divergence is nonnegative, $\mathrm{TC}(\mathbf{g}_1, \ldots, \mathbf{g}_T \mid \mathcal{V}) = 0$ if and only if

$$p^\star(\mathbf{g} \mid \mathcal{V}) \;=\; \prod_{t=1}^{T} p^\star(\mathbf{g}_t \mid \mathcal{V}), \tag{11}$$

meaning that gaze positions are conditionally independent across time given the video. Human gaze clearly contains temporal dependencies. Therefore, even if a marginal model perfectly estimates every per-frame distribution, it can still have an irreducible error in the joint trajectory distribution.

## C Velocity Network Architecture

This section gives the parts of the velocity network $\mathbf{v}_\theta$ that Section 3.2 does not cover: the encoder and projection (Section C.1), the rotary position embedding (Section C.2), and the four sub-layers of a denoiser block (Section C.3).

### C.1 Encoder and Projection

**Backbone selection.** We use V-JEPA 2 ViT-L because its features capture both fine-grained motion and clip-level action structure [1]: the former supplies the local visual evidence read by our per-frame

cross-attention (bottom-up), and the latter the task-level structure pooled by our task-token bank (top-down). A linear probe on frozen features supports this choice (Table 7): at the native resolution $256\times256$, V-JEPA 2 matches Qwen3-VL-8B [2] in AUC and exceeds the best Qwen3-VL layer in F1 (0.378 vs. 0.370) and Precision (0.292 vs. 0.280), while a scale-matched input $240\times320$ degrades every metric.

Table 7: **Linear probing of frozen backbone features on EGTEA Gaze+**. Top block: V-JEPA 2 ViT-L at its native pre-training resolution and at a 4:3 input that matches the dataset aspect ratio. Bottom block: Qwen3-VL-8B at four LM layers, native resolution. All probes are ridge regressions on the frozen spatial token grid; per-column best in bold.

| Backbone | Resolution | Layer | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ |
|---|---|---|---|---|---|---|---|
| **V-JEPA 2 ViT-L** | **256×256** (native) | last | **0.944** | **0.378** | **0.292** | 0.535 | 11.39 |
| V-JEPA 2 ViT-L | 240×320 (matched) | last | 0.913 | 0.296 | 0.216 | 0.469 | 12.87 |
| Qwen3-VL-8B | native | 2 | 0.944 | 0.370 | 0.280 | **0.549** | 11.37 |
| Qwen3-VL-8B | native | 4 | 0.944 | 0.370 | 0.280 | 0.547 | 11.37 |
| Qwen3-VL-8B | native | 8 | 0.943 | 0.369 | 0.278 | **0.549** | 11.34 |
| Qwen3-VL-8B | native | 16 | 0.942 | 0.368 | 0.279 | 0.541 | **11.24** |

**Projection and spatial encoder.** A linear layer projects each encoder token from $D$=1024 to $d$=256. We add a 2D sinusoidal spatial position encoding and pass the tokens of each frame through a two-layer Transformer encoder, which lets the $N$ spatial tokens of a frame exchange information. This encoder works on each frame on its own and never mixes information across time.

### C.2 Rotary Position Embedding

For the frame-to-frame self-attention (Sub-layer 1 below) we use 1D RoPE [48] indexed by the frame position $p \in \{0, \dots, T-1\}$, with base $\theta = 10000$. RoPE rotates each query/key pair $(q_{2j}, q_{2j+1})$ by $\phi_{p,j} = p \cdot \theta^{-2j/d_h}$ before the attention dot-product; values are not rotated. By construction the logit $a_{p,p'} = \langle \mathrm{RoPE}(\mathbf{q}_p), \mathrm{RoPE}(\mathbf{k}_{p'}) \rangle / \sqrt{d_h}$ depends only on the relative offset $p - p'$, which is the structure gaze dynamics require (consecutive-frame coupling and saccade step length scale with $|p - p'|$, not with absolute $p$). See Section 4.4 and Section 4.5 for the effect of RoPE.

### C.3 DenoiserBlock: Four Sub-layers

Each denoiser block combines the tensors defined in the previous subsections: gaze tokens $\mathbf{h}$ from the input pipeline, projected encoder tokens $\mathbf{V}$ from Section C.1, and task tokens $\mathbf{c}_{\text{task}}$ from Eq. 4. Each of the $L$=6 blocks applies four pre-norm residual sub-operations to the gaze tokens $\mathbf{h} \in \mathbb{R}^{T\times d}$. All attentions use $H = 8$ heads with $d_h = d/H = 32$ and layer-specific QKV/output projections (multi-head attention, MHA, follows the standard definition).

**Sub-layer 1: Frame-to-frame self-attention with RoPE.**

$$\mathbf{h} \leftarrow \mathbf{h} + \mathrm{MHA}\big(\mathrm{RoPE}(\mathbf{Q}),\ \mathrm{RoPE}(\mathbf{K}),\ \mathbf{V};\ \mathbf{h}\big), \tag{12}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are linear projections of $\mathrm{LayerNorm}(\mathbf{h})$ and RoPE is applied to queries and keys but not to values. This is the only pathway along which gaze tokens at different frames exchange information, and is therefore where the joint structure of $p(\mathbf{g} \mid \mathcal{V})$ enters the network.

**Sub-layer 2: Per-frame visual cross-attention.** Each gaze token $\mathbf{h}_t$ cross-attends *only* to its own frame's $N$ spatial tokens $\mathbf{V}_t$:

$$\mathbf{h}_t \leftarrow \mathbf{h}_t + \mathrm{MHA}\big(\mathrm{LayerNorm}(\mathbf{h}_t),\ \mathbf{V}_t,\ \mathbf{V}_t\big). \tag{13}$$

Long-range temporal aggregation is left to Sub-layer 1; this sub-layer reads the encoder tokens of the frame being denoised. Unlike Sub-layer 3, this sub-layer has no gate, so the visual condition acts at full strength from the start of training.

**Algorithm 1** Training and inference for GAZEFLOW.

**Require:** Dataset $\mathcal{D}$; encoder $E_\psi$; velocity network $\mathbf{v}_\theta$; Euler steps $S$; samples per clip $K$.

**Training.**

**repeat**
  Sample $(\mathcal{V}, \mathbf{g}, \mathbf{m}) \sim \mathcal{D}$; $\mathbf{c} \leftarrow E_\psi(\mathcal{V})$.
  Sample $s \sim U[0,1]$ and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\mathbf{x}_s \leftarrow (1-s)\,\mathbf{x}_0 + s\,\mathbf{g}$.
  $\mathcal{L} \leftarrow \|\mathbf{m} \odot (\mathbf{v}_\theta(\mathbf{x}_s, s, \mathbf{c}) - (\mathbf{g} - \mathbf{x}_0))\|_2^2$; update $(\psi, \theta)$.
**until** convergence

**Inference.**

$\mathbf{c} \leftarrow E_\psi(\mathcal{V})$. ▷ once per clip
**for** $k = 1, \ldots, K$ **in parallel do**
  $\mathbf{x}^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
  **for** $i = 0, \ldots, S-1$ **do**
    $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k)} + \frac{1}{S}\,\mathbf{v}_\theta(\mathbf{x}^{(k)}, i/S, \mathbf{c})$.
  **end for**
**end for**
**return** $\{\mathbf{x}^{(k)}\}_{k=1}^{K}$.

**Sub-layer 3: Task cross-attention.** Each gaze token cross-attends to the clip-level task tokens $\mathbf{c}_{\text{task}} \in \mathbb{R}^{(M+1)\times d}$ defined in Eq. 4:

$$\mathbf{h} \;\leftarrow\; \mathbf{h} + \sigma(g^{(\ell)}) \cdot \mathrm{MHA}\big(\mathrm{LayerNorm}(\mathbf{h}),\, \mathbf{c}_{\text{task}},\, \mathbf{c}_{\text{task}}\big). \tag{14}$$

The per-layer gate $g^{(\ell)} \in \mathbb{R}^d$ is a channel-wise learnable vector applied element-wise to the cross-attention residual (broadcast over the $T$ gaze tokens). All channels are initialized at $-2$ so that $\sigma(g_i^{(\ell)}) \approx 0.12$ uniformly at the start of training: the task input starts weak and grows as the Task Injection module learns useful features, letting the rest of the network train without interference from a randomly initialized task pathway.

# D Training and Sampling Algorithms

This section gives the training and inference algorithm (Appendix D.1), self-conditioning (Appendix D.2), and the sliding window for long clips (Appendix D.3).

## D.1 Training and Inference Algorithm

Algorithm 1 summarizes training and inference. For clarity, it omits self-conditioning (Appendix D.2) and the parameter EMA.

## D.2 Self-Conditioning Details

Self-conditioning [7] feeds the model's previous estimate of the clean trajectory, $\hat{\mathbf{x}}_1 = \mathbf{x}_s + (1-s)\,\mathbf{v}_\theta$, back as an extra input. During training, we use this estimate on half of the steps and a zero input on the other half. At inference, each Euler step uses the estimate from the previous step, starting from zero.

## D.3 Sliding-Window Inference for Long Clips

The velocity network is trained on fixed windows of $W = 64$ frames. At inference, clips longer than $W$ are split into overlapping windows of stride $W_s = W/2 = 32$, and Algorithm 1 is run independently inside each window. In the overlap region between consecutive windows $i$ and $i{+}1$, predictions are linearly blended along the frame axis $\tau$ to avoid window seams:

$$\mathbf{x}_\tau^{\text{blend}} \;=\; (1-\alpha_\tau)\,\mathbf{x}_\tau^{(i)} + \alpha_\tau\,\mathbf{x}_\tau^{(i+1)}, \qquad \alpha_\tau \;=\; \frac{\tau - \tau_{\text{start}}^{(i+1)}}{W_s} \in [0,1], \tag{15}$$

where $\tau_{\text{start}}^{(i+1)}$ is the first frame of window $i{+}1$ and $\alpha_\tau$ ramps from 0 to 1 across the overlap. We pad clips shorter than $W$ by repeating the last frame, and we cut the output back to the original length.

# E Experimental Details

This appendix supplements Section 4.1: Section E.1 describes the datasets, Section E.2 the baselines, Section E.3 the metrics and evaluation protocol, and Section E.4 our hyperparameters.

## E.1 Datasets

**EGTEA Gaze+.** EGTEA Gaze+ [33] contains 86 cooking videos (106 action classes with 19 verbs and 53 nouns) recorded at $640 \times 480$ / 24 fps together with synchronized gaze from an SMI head-mounted eye tracker at 30 Hz. We follow the official train / val split from Li et al. [33], yielding 8,299 training and 2,022 validation clips after aligning video and gaze timelines. Eye-tracker samples are classified into Fixation, Saccade, and Blink events by the authors' BeGaze output; we carry forward all event types into the trajectory target but exclude Blink frames from the loss via a binary validity mask $\mathbf{m}$.

**Ego4D (Aria-glass gaze subset).** The Ego4D [12] benchmarks include a subset of clips captured with Project Aria glasses equipped with an integrated eye tracker, providing per-frame gaze at $1088 \times 1080$ / 30 fps. We use the released train / val partition (12,178 / 5,202 clips). Raw gaze is already aligned to the video grid by the Aria SDK; we rescale coordinates to the native $1088 \times 1080$ resolution, normalize to $[-1, 1]$ for training. Ego4D has no action or verb labels, so the task-token probe in Section 4.3 uses EGTEA Gaze+ only.

**Windowing.** All experiments operate on fixed $T = 64$-frame windows drawn from each clip: clips shorter than $T$ are right-padded with the last valid frame and their padding positions are zeroed in the validity mask; clips longer than $T$ are trained with random crops and evaluated with a sliding window (stride 32, linear blend in the overlap, Appendix D.3). Clip-length statistics and per-split window counts are summarized in Table 8.

Table 8: Dataset summary.

| Dataset | Resolution | fps | Train clips | Val clips |
|---|---|---|---|---|
| EGTEA Gaze+ | $640 \times 480$ | 24 | 8,299 | 2,022 |
| Ego4D (Aria gaze) | $1088 \times 1080$ | 30 | 12,178 | 5,202 |

## E.2 Baselines

**Attention Transition (AT).** AT [17] is trained in three stages: (SP) a two-stream CNN predicts a per-frame heatmap, (LSTM) a recurrent module classifies each frame as Fixation or Saccade, and (LF) a late-fusion CNN re-weights SP by the fixation–saccade posterior. On EGTEA we retrain the full SP+LSTM+LF pipeline with the authors' released recipe on the 8,299 train clips. On Ego4D, fixation/saccade event labels are not provided, so only Stage SP is retrained and the LSTM and LF stages are disabled. AT inference heatmaps are resized to $480 \times 640$ before evaluation.

**GLC.** GLC [26] is a transformer saliency model with a "global–local correlation" head; we use the authors' released checkpoints for EGTEA and Ego4D without modification. Predictions are produced at $64 \times 64$ and upscaled to $480 \times 640$ for evaluation to match our saliency protocol.

**EgoM2P.** EgoM2P [31] is an autoregressive multimodal foundation model pretrained on eight egocentric datasets that emits discrete gaze tokens one at a time. We report two variants: (i) the released zero-shot checkpoint, and (ii) a LoRA-finetuned variant with rank $r = 16$ applied to the query and value projections of all transformer layers, trained on each dataset's train split with the same optimizer as our own model. For both variants we draw $K{=}50$ trajectories per clip at temperature 1.0, matching our sampler.

**Prior-only references.** Two simple priors serve as reference points in Table 2 and Table 3. *Center Bias* is a static 2D Gaussian centered at the image center with per-axis standard deviation matched to the empirical fixation spread of each dataset's train split; it is evaluated with the same saliency protocol as every other baseline and has no trainable parameters. *Random Walk* takes the previous frame's ground-truth fixation (or the clip centroid at $t{=}0$) and samples the next predicted fixation from an isotropic Gaussian with $\sigma$ matched to the empirical frame-to-frame displacement of the train split; it is explicitly a first-order Markov prior and carries no visual input at all.

### E.3 Metrics and Evaluation Protocol

The metrics below mirror the two-axis structure of Section 4.1: *per-frame prediction accuracy* scores each predicted heatmap against the per-frame ground-truth gaze, while *alignment with human gaze temporal dynamics* scores sampled trajectories against the human trajectory either directly or as a step-size distribution.

**Per-frame prediction accuracy.** In egocentric gaze datasets, each frame is annotated with a ground-truth gaze point. Following the GLC protocol [26], we place a Gaussian kernel at this point to form a fixation map, which is then compared with the predicted heatmap. *Area Under the Curve* (AUC, the AUC-Judd variant [5, 21]) treats the predicted heatmap as a continuous per-pixel score and measures how well it ranks ground-truth gaze locations above non-gaze locations. *F1*, *Precision*, and *Recall* instead binarize the predicted heatmap and the ground-truth fixation map at a fixed threshold and measure the overlap between the two. *Average Angular Error* (AAE) [17] measures the angular deviation between the predicted gaze point and the ground-truth gaze point.

Per-frame metrics aggregate hierarchically: per frame, then averaged within each clip (the clip-mean), then averaged across clips (the global mean), which follows GLC's original protocol and makes each clip count equally regardless of length. For F1, Precision, and Recall, the threshold is selected by sweeping over the heatmap; following Lai et al. [26], we report Adaptive F1 (the best F1 over the sweep) with Precision and Recall taken at the same threshold. Beyond AUC-Judd, we additionally report four standard saliency metrics following Bylinskii et al. [5]: NSS (the mean normalized saliency at fixation locations), the linear correlation coefficient CC, the histogram similarity SIM, and the KL divergence from the ground-truth to the predicted distribution.

**Alignment with human gaze temporal dynamics.** Per-frame metrics score each frame independently and therefore do not measure whether the predicted gaze forms a natural scanpath over time. The metrics below evaluate how closely the dynamics of sampled trajectories match those of human gaze. *Average Displacement Error* (ADE) and *Dynamic Time Warping distance* (DTW) [44, 23] compare each predicted trajectory against the human trajectory directly; DTW is the minimum Euclidean alignment cost between the two under a monotone temporal warping, length-normalised by the trajectory length. For both, we report the mean over $K = 50$ samples and the best-of-$K$ value.

We additionally compare step-size statistics. For each trajectory, we measure how far the gaze moves between consecutive frames, and build a displacement histogram, which represents the empirical distribution of these per-frame distances, aggregated across all frames and clips. We report the mean and median displacement as ratios to the human reference, and the Jensen–Shannon divergence (JSD) between the predicted and human displacement histograms. The JSD uses base $e$ on a 100-bin histogram over the union range of all method and human displacements, which captures distribution shape rather than only its first two moments.

**Trajectory-to-heatmap conversion.** Trajectory generators – our model, EgoM2P, and pseudo-trajectory baselines – are converted to per-frame heatmaps so that all methods are comparable on the saliency metrics above. For each frame $t$ we (i) denormalize all $K$ predicted gaze points from $[-1, 1]$ to pixel coordinates (the native evaluation resolution of the target dataset); (ii) place a Gaussian kernel with $\sigma = 25$ px at each prediction; (iii) average over the $K$ samples to obtain $\mathbf{S}_t$. The bandwidth $\sigma$=25 px matches the one used by the released GLC predictions.

**Frozen-encoder protocol for ablations.** The ablation study (Section 4.4) and the design-choice analysis (Section 4.5) evaluate the frozen-encoder variant of GAZEFLOW on a fixed validation subset, rather than the LoRA variant on the full validation split. We make this choice for two reasons. First, end-to-end LoRA training together with $K$=50 sampling for every architectural variant is computationally prohibitive. Second, every component varied in those studies (the visual and task cross-attention pathways, the task-token bank, the temporal RoPE, and the flow-matching head) operates on top of the encoder representation and does not change the encoder itself. We therefore expect the ranking of variants to hold for the LoRA-adapted encoder as well.

**Validation subset.** The subset is a random sampled subset of the EGTEA Gaze+ validation split, drawn once with a fixed random seed and held identical across every variant in the ablation and design-choice studies. After applying the GLC-format Fixation filter, the subset contains 1,536 frame-level paired observations per metric.

**Significance testing.** We compare each result in Table 2, Table 6, and Section 4.5 with its reference using a paired Wilcoxon signed-rank test on the frame-level scores, since these scores are heavy-tailed. Almost every comparison is significant at $p < 0.05$; the exception is *Recall* on Ego4D in Table 2, where Random Walk is not beaten by GAZEFLOW.

### E.4 Training and Sampling Hyperparameters

**Video feature extraction.** The visual encoder is V-JEPA 2 ViT-L [1], fine-tuned with LoRA jointly with the velocity network ("LoRA configuration" below). Frames are resized to $256 \times 256$ and read at 24 fps on EGTEA Gaze+ and 30 fps on Ego4D, and we take the last hidden state of the encoder. This yields $N = 256$ spatial tokens of dimension $D = 1024$ per frame.

**Optimizer and schedule.** AdamW with $\beta_1$=0.9, $\beta_2$=0.999, weight decay $10^{-4}$; learning rate $10^{-4}$ with cosine annealing to $10^{-6}$ over 80 epochs; effective batch size 16; EMA decay $\beta_{\text{EMA}}$=0.995, and all evaluations use the EMA weights; gradient clipping at norm 1.0. Early stopping uses patience 15 on validation loss computed at end of each epoch.

**LoRA configuration.** The V-JEPA 2 ViT-L encoder is fine-tuned with LoRA [16] rank $r$=16, $\alpha$=32, dropout 0.05, applied to the query and value projections of every transformer block. All other encoder weights are frozen; only LoRA adapters, the input projection $\mathbf{W}_{\text{proj}}$, and the velocity network are trainable.

**Flow-matching parameters.** We use the OT-form conditional interpolant $\mathbf{x}_s = (1-(1-\sigma_{\min})s)\,\mathbf{x}_0 + s\,\mathbf{g}$ with $\sigma_{\min} = 10^{-3}$, which differs from the linear interpolant in Eq. (7) only by an $O(\sigma_{\min})$ correction below pixel precision; self-conditioning is applied with probability 0.5 during training and deterministically on at inference; training uses the uniform schedule $s \sim \mathcal{U}(0, 1)$ as in Lipman et al. [37].

**Sampling.** Euler ODE integration with $S = 50$ steps and $K = 50$ trajectories per clip. Sliding window $W = 64$, stride 32, linear blend in overlap (Appendix D.3).

## F Additional Results

This appendix gives results that support Section 4: the full main comparison (Section F.1), the inference cost and the effect of the sample count $K$ and step count $S$ (Section F.2), the task-query sweep (Section F.3), and two further ablations (Section F.4).

### F.1 Full Main Comparison Table

Table 9 expands Table 2 with all nine standard metrics, split into the saliency block (NSS, AUC, CC, KL, SIM) and the localization block (F1, P, R, AAE).

Table 9: Full saliency and localization comparison on EGTEA Gaze+ and Ego4D.

| | **EGTEA Gaze+** | | | | | | | | | **Ego4D** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Saliency | | | | | Localization | | | | Saliency | | | | | Localization | | | |
| Method | NSS↑ | AUC↑ | CC↑ | KL↓ | SIM↑ | F1↑ | P↑ | R↑ | AAE↓ | NSS↑ | AUC↑ | CC↑ | KL↓ | SIM↑ | F1↑ | P↑ | R↑ | AAE↓ |
| *Prior-only reference* | | | | | | | | | | | | | | | | | | |
| Center Bias | 0.679 | 0.906 | 0.095 | 6.635 | 0.110 | 0.185 | 0.162 | 0.215 | 16.07 | 0.780 | 0.930 | 0.122 | 5.873 | 0.136 | 0.222 | 0.194 | 0.258 | 13.92 |
| Random Walk | 0.623 | 0.824 | 0.096 | 3.926 | 0.101 | 0.147 | 0.090 | 0.413 | 16.34 | 0.113 | 0.763 | 0.017 | 4.256 | 0.072 | 0.109 | 0.059 | **0.662** | 14.38 |
| *Marginal density estimation* | | | | | | | | | | | | | | | | | | |
| GLC [26] | 2.216 | 0.953 | 0.279 | 3.288 | 0.249 | 0.421 | 0.348 | 0.531 | 10.21 | 1.960 | 0.947 | 0.256 | 3.438 | 0.231 | 0.355 | 0.267 | 0.530 | 11.52 |
| *Autoregressive* | | | | | | | | | | | | | | | | | | |
| EgoM2P (zero-shot) [31] | 1.586 | 0.921 | 0.231 | 2.946 | 0.189 | 0.280 | 0.195 | 0.498 | 13.18 | 1.499 | 0.911 | 0.218 | 3.150 | 0.178 | 0.253 | 0.180 | 0.422 | 13.86 |
| EgoM2P (LoRA) [31] | 1.735 | 0.921 | 0.247 | 2.737 | 0.198 | 0.293 | 0.205 | 0.515 | 12.81 | 1.886 | 0.930 | 0.267 | 2.715 | 0.206 | 0.297 | 0.217 | 0.470 | 12.69 |
| AT [17] | 2.722 | 0.956 | 0.378 | 1.949 | 0.269 | 0.419 | 0.339 | 0.547 | 9.88 | 2.289 | 0.954 | 0.327 | **2.156** | 0.197 | 0.346 | 0.268 | 0.487 | 12.01 |
| *Flow matching (ours)* | | | | | | | | | | | | | | | | | | |
| **GAZEFLOW (ours)** | **3.534** | **0.964** | **0.438** | **1.936** | **0.357** | **0.491** | **0.414** | **0.603** | **9.01** | **2.676** | **0.958** | **0.358** | 2.213 | **0.282** | **0.392** | **0.307** | 0.540 | **10.46** |

### F.2 Efficiency and Accuracy–Cost Trade-off

We measure the cost of GAZEFLOW on a single NVIDIA H100 (80 GB) in BF16. Unless noted, the test clip is a $T$=64-frame EGTEA Gaze+ segment, which lasts about 2.7 s at 24 fps, and we draw $K$=50 trajectories with $S$=50 Euler steps.

**Model size.** The velocity network has $\sim$10.4M parameters, of which $\sim$7.9M are in the six DenoiserBlocks of Section 3.2. The spatial-token encoder adds $\sim$1.6M, the visual-token projection $\mathbf{W}_{\text{proj}}$ and the learnable-query cross-attention that builds $\mathbf{c}_{\text{task}}$ add $\sim$0.5M together, and the coordinate, timestep, and self-conditioning encoders and the output layer add the last $\sim$0.4M. The V-JEPA 2 ViT-L encoder has $\sim$327.8M parameters, and we train only its LoRA adapters, which add $\sim$1.9M. In total, we train $\sim$12.2M parameters.

**Inference cost.** One forward pass of the velocity network on one 64-frame trajectory costs $\sim$ 100 GFLOPs. With $K$=50 samples and $S$=50 steps, we call the network 2,500 times, for $\sim$250 TFLOPs per clip. The encoder runs only once per clip and adds less than 5% to this cost. In wall-clock time, one clip takes $\sim$3.37 s, of which the encoder with LoRA takes $\sim$450 ms and Euler sampling takes $\sim$2.92 s (mean over five runs), with a peak GPU memory of 9.2 GB.

**Training time.** End-to-end training with LoRA takes $\sim$3 days on EGTEA Gaze+ and $\sim$5 days on Ego4D, both on 4$\times$ H100 GPUs in BF16 mixed precision.

**Accuracy–cost trade-off.** Inference cost depends on two settings: $K$, the number of trajectory samples per clip, and $S$, the number of Euler steps per sample. The main comparison uses a conservative $K$=50 and $S$=50. Table 10 sweeps both with the LoRA model and reports F1 and the inference time per 64-frame clip on a single H100. At $K$=30 and $S$=5, F1 is 0.493 and inference takes 0.73 s, 4.6$\times$ faster than the default setting. We recommend $K$ between 10 and 30 with $S$=5.

Table 10: **F1 and inference time over the sample count $K$ and the Euler step count $S$** (EGTEA Gaze+, GAZEFLOW LoRA). Time is per 64-frame clip on a single H100. The recommended setting is in bold.

| | F1↑ | | | | | Time (s)↓ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $K \setminus S$ | 1 | 2 | 5 | 10 | 50 | 1 | 2 | 5 | 10 | 50 |
| 1 | 0.453 | 0.465 | 0.444 | 0.427 | 0.405 | 0.49 | 0.49 | 0.51 | 0.54 | 0.77 |
| 10 | 0.460 | 0.481 | 0.484 | 0.479 | 0.468 | 0.52 | 0.53 | 0.57 | 0.63 | 1.14 |
| 30 | 0.460 | 0.484 | **0.493** | 0.492 | 0.486 | 0.58 | 0.62 | **0.73** | 0.91 | 2.35 |
| 50 | 0.460 | 0.485 | 0.495 | 0.495 | 0.491 | 0.65 | 0.71 | 0.89 | 1.21 | 3.37 |

Table 11 compares inference time across methods on the same H100. With the recommended setting, GAZEFLOW is faster than every baseline.

Table 11: **Inference time across methods** on a single H100. The ms / frame column gives the time to predict gaze for one video frame, and throughput is its inverse.

| Method | ms / frame↓ | Throughput (fps)↑ |
|---|---|---|
| GLC [26] | 12.5 | 80 |
| AT [17] | 219 | 4.6 |
| EgoM2P [31] | 506 | 2.0 |
| GAZEFLOW ($K$=50, $S$=50) | 53 | 19 |
| **GAZEFLOW ($K$=30, $S$=5)** | **11.4** | **88** |

**Sample count $K$.** Table 12 sweeps the sample count $K \in \{1, 5, 10, 30, 50\}$ for GAZEFLOW (LoRA), using the same $K$=50 samples as Table 3 and subsampling the first $K$. The mean over $K$ barely changes (1.05 $\rightarrow$ 1.04 for ADE, 0.96 $\rightarrow$ 0.95 for DTW). The gain in best-of-$K$ therefore comes from drawing more samples, not from better individual samples. Best-of-$K$ changes little after $K$=30 (ADE 0.64 vs. 0.60 at $K$=50), so $K$=50 in the main comparison is a safe choice.

Table 12 also reports the per-frame metrics for the same $K$ values. They also rise with $K$ and flatten out: $K$=10 keeps 96% of the F1 at $K$=50, and $K$=30 is close to $K$=50 on every metric.

Table 12: **Sample count $K$ sweep on EGTEA Gaze+.** ADE and DTW are reported in $\times 10^2$ px following Table 3; "Mean" and "Best" are mean-over-$K$ and best-of-$K$. At $K{=}1$ the two coincide. The $K{=}50$ row matches Table 2 and Table 3.

| | | | | | | ADE ($\times 10^2$ px)↓ | | DTW ($\times 10^2$ px)↓ | |
|---|---|---|---|---|---|---|---|---|---|
| $K$ | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ | Mean | Best | Mean | Best |
| 1 | 0.866 | 0.398 | 0.350 | 0.462 | 11.10 | 1.05 | 1.05 | 0.96 | 0.96 |
| 5 | 0.942 | 0.457 | 0.378 | 0.578 | 9.61 | 1.05 | 0.80 | 0.95 | 0.68 |
| 10 | 0.953 | 0.469 | 0.395 | 0.579 | 9.28 | 1.05 | 0.73 | 0.95 | 0.60 |
| 30 | 0.961 | 0.487 | 0.399 | 0.623 | 9.05 | 1.04 | 0.64 | 0.95 | 0.51 |
| 50 | 0.964 | 0.491 | 0.414 | 0.603 | 9.01 | 1.04 | 0.60 | 0.95 | 0.48 |

### F.3 Capacity Sweep over Task Queries

Table 13 reports the full sweep over $M \in \{1, 2, 4, 8, 16\}$, the number of learnable task queries, with the global pool token kept (Section 4.5). $M{=}4$ gives the best AUC and F1.

Table 13: **Capacity sweep over learnable task queries.** $M$ is the number of learnable task queries, and every variant keeps the global pool token. Per-column best in bold.

| $M$ | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ |
|---|---|---|---|---|---|
| 1 | 0.968 | 0.492 | **0.404** | 0.630 | **8.68** |
| 2 | 0.967 | 0.486 | 0.392 | 0.639 | 8.87 |
| **4** | **0.970** | **0.493** | 0.401 | 0.638 | 8.81 |
| 8 | 0.968 | 0.483 | 0.390 | 0.635 | 8.79 |
| 16 | 0.968 | 0.480 | 0.381 | **0.650** | 8.75 |

### F.4 Additional Ablations

**Causal variant.** We train a causal variant of GAZEFLOW that sees only past and current frames. It adds a causal mask to the time-aware self-attention, builds the task-token bank from past and current frames, and extracts visual features with a causal sliding window. It loses at most $12.0\%$ across the five metrics (Table 14).

**Flow matching vs. diffusion.** We train a diffusion version of GAZEFLOW that changes only the objective (noise prediction, cosine schedule, DDIM sampling). Both heads call the same network once per step, so they take the same time at each step count $S$. As Table 15 shows, flow matching is best at $S{=}5$, while diffusion needs all 50 steps, $3.8\times$ the time, and is still less accurate.

Table 14: **Bidirectional vs. causal GAZEFLOW.** The causal variant sees only past and current frames.

| Variant | AUC↑ | F1↑ | P↑ | R↑ | AAE↓ |
|---|---|---|---|---|---|
| Bidirectional | 0.970 | 0.493 | 0.401 | 0.638 | 8.81 |
| Causal | 0.953 | 0.437 | 0.353 | 0.574 | 8.71 |

Table 15: **Flow matching vs. diffusion over the number of sampling steps $S$ ($K{=}50$).** Time is per 64-frame clip on a single H100.

| $S$ | 2 | 5 | 10 | 50 |
|---|---|---|---|---|
| F1↑, flow matching (ours) | 0.496 | **0.506** | 0.503 | 0.493 |
| F1↑, diffusion | 0.072 | 0.359 | 0.462 | 0.485 |
| Time (s)↓ | 0.71 | 0.89 | 1.21 | 3.37 |

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state that we model gaze trajectories as a video-conditioned distribution and present GAZEFLOW, a flow-matching model with dual-pathway video conditioning; these claims map to the architecture in Sec. 3 and the empirical results in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4.7 discusses several limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We do not introduce new theoretical results. The conditional flow-matching derivation reproduced in App. B follows the standard arguments of Lipman et al. [37], Tong et al. [50] and is included for self-containedness.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The architecture is specified to operator level in App. C; the training and sampling procedure is given as Algorithm 1 in App. D; data preprocessing, optimizer, schedule, and other hyperparameters are listed in App. E. We additionally include the inference code and the trained checkpoint in the anonymized supplementary archive.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include an anonymized supplementary archive containing the inference code, the trained model checkpoint, configuration files, the manifest schema, and a README that documents the environment setup and the exact command to reproduce evaluation. The two datasets used (EGTEA Gaze+ and Ego4D) are publicly available from their original distributors. The full training code together with the same checkpoint will be released open-source upon paper acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4.1 states the datasets, metrics, and compute environment; App. E.1 details the splits and gaze processing, and App. E.4 lists the optimizer, learning-rate schedule, batch size, EMA decay, gradient-clip threshold, and early-stopping criterion.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We test statistical significance of every architectural comparison via a paired Wilcoxon signed-rank test on per-frame scores; the full procedure (test choice, pairing, sample size) is described in the Significance testing paragraph of App. E.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4.1 states that all experiments run on $4\times$NVIDIA H100 GPUs in bfloat16 mixed precision; App. E.4 lists the full training recipe, and App. D notes the overhead added by the two-pass self-conditioning schedule.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We use existing public datasets under their stated terms of use; the work does not involve new human-subject data collection, crowdsourcing, or the release of an asset with high misuse potential.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our contribution is a methodological advance in conditional generative modeling of gaze trajectories on egocentric video. We do not foresee a direct path from this work to a deployment with significant positive or negative societal impact beyond those already inherent to general-purpose egocentric vision research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The model is a gaze trajectory generator for egocentric video; it is not a generative model of people, scraped web content, or language, and does not pose the dual-use risks the question targets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: External assets are cited and used under their original terms: EGTEA Gaze+ [33] and Ego4D [12] are accessed under the dataset providers' standard academic-use agreements, and the V-JEPA 2 ViT-L backbone [1] is loaded from the publicly released model. Versions and per-asset access details appear in App. E.1.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The anonymized supplementary archive ships with a top-level README documenting the package layout, environment setup, the JSONL manifest, the inference command, and the contents of the released checkpoint; the package is distributed under the MIT license (see the bundled `LICENSE` file).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work does not involve crowdsourcing or new research with human subjects; we use only previously released gaze datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No new human-subject data collection was conducted by the authors; the eye-tracking data used for training comes from third-party releases whose original review processes are documented by their authors.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We used an LLM only for light writing polish on the manuscript; it is not an important, original, or non-standard component of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.