StreamGaze: Gaze-Guided Temporal Reasoning and Proactive Understanding in Streaming Videos

Daeun Lee[1] Subhojyoti Mukherjee[2] Branislav Kveton[2] Ryan A. Rossi[2] Viet Dac Lai[2]
Seunghyun Yoon[2] Trung Bui[2] Franck Dernoncourt[2] Mohit Bansal[1]
[1]UNC Chapel Hill [2]Adobe Research

Code Project page

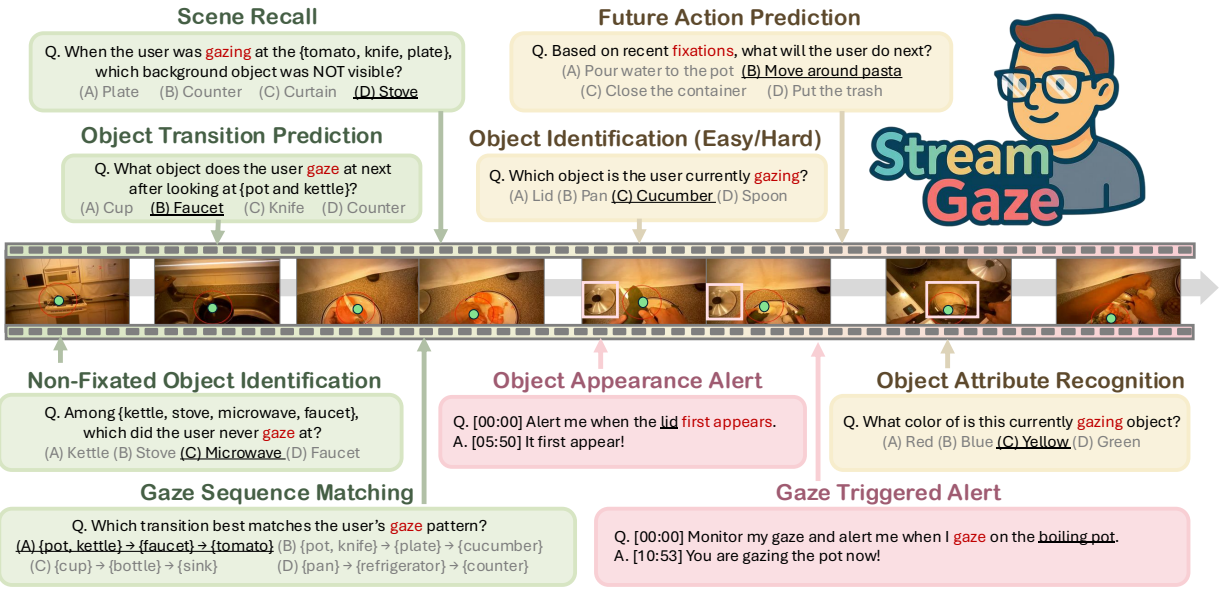


Figure 1. **StreamGaze's task taxonomy.** We introduce StreamGaze, the first benchmark designed to evaluate how effectively MLLMs use gaze for temporal and proactive reasoning in streaming videos. We introduce gaze-guided past, present, and proactive streaming video understanding tasks.

Abstract

Streaming video understanding requires models not only to process temporally incoming frames, but also to anticipate user intention for realistic applications like AR glasses. While prior streaming benchmarks evaluate temporal reasoning, none measure whether MLLMs can interpret or leverage human gaze signals within a streaming setting. To fill this gap, we introduce StreamGaze, *the first benchmark designed to evaluate how effectively MLLMs use gaze for temporal and proactive reasoning in streaming videos.* StreamGaze *introduces gaze-guided past, present, and proactive tasks that comprehensively evaluate streaming video understanding. These tasks assess whether models can use real-time gaze to follow shifting attention and infer user intentions from only past and currently observed frames. To build* StreamGaze, *we develop a gaze–video QA generation pipeline that aligns egocentric videos with raw gaze trajectories via fixation extraction, region-specific visual prompting, and scanpath construction. This pipeline produces spatio-temporally grounded QA pairs that closely reflect human perceptual dynamics. Across all* StreamGaze *tasks, we observe substantial performance gaps between state-of-the-art MLLMs and human performance, revealing fundamental limitations in gaze-based temporal reasoning, intention modeling, and proactive prediction. We further provide detailed analyses of gaze-prompting strategies, reasoning behaviors, and task-specific failure modes, offering deeper insight into why current MLLMs struggle and what capabilities future models must develop. All data and code will be publicly released to support continued research in gaze-guided streaming video understanding.*

1. Introduction

Recent advances in Video Large Language Models (VideoLLMs) [12, 26, 33] have significantly improved multimodal understanding of dynamic visual environments. A growing line of work now investigates *streaming video understanding* [2, 4, 29, 34], where models must process temporally incoming frames and respond in real time without access to future context. Such capability is essential for real-world applications such as robotics, embodied agents, and AR-glass assistants, where perception and decision-making must occur continuously as events unfold.

Benchmarks play a critical role in tracking progress in this rapidly developing domain. While several streaming benchmarks exist [2, 4, 28, 31, 36], they capture only some challenges faced by realistic streaming agents. As summarized in Table 1, some benchmarks [29, 31] focus primarily on past or present recognition, leaving out proactive understanding, the ability to anticipate future events and user intentions. Moreover, existing benchmarks rarely incorporate the human perceptual signals that actually drive real-world decision-making, even when they rely on egocentric video sources and implicitly assume AR-glasses usage scenarios. Among these perceptual cues, *eye gaze* is the most direct and reliable indicator of where a user is attending, what they are processing, and what they are likely to do next [7, 10, 11, 35]. Omitting gaze removes the key perceptual signal humans use to filter relevant information, plan actions, and anticipate what will happen next. This leads to benchmark evaluations that diverge from the way humans actually perceive and reason about dynamic environments.

However, incorporating gaze into video understanding has been challenging [7, 19]. Unlike conventional QA that relies solely on visible content, gaze-guided QA requires interpreting *dynamic gaze trajectories*, modeling how attention shifts over time, and grounding these signals within a moving egocentric viewpoint. Raw gaze streams are noisy and contain fast, unstable fluctuations [9, 16, 24], while egocentric videos continuously shake and rotate with user motion, making even *"what the user is looking at"* a non-trivial spatio-temporal grounding problem. These challenges are further amplified in the *streaming* setting, where models must reason *causally* using only past and currently observed frames. Together, these factors make automatic construction of temporally grounded QA pairs highly challenging.

To close this gap, we introduce StreamGaze, the first benchmark for **gaze-guided streaming video understanding**. As shown in Figure 1, StreamGaze provides a unified suite of gaze-conditioned tasks spanning *past*, *present*, and *proactive* reasoning, enabling comprehensive evaluation of how well MLLMs use gaze for temporal understanding and future prediction. Solving these tasks requires models to interpret gaze online, follow shifting attention over time, and infer intention using only past and currently observed frames, mirroring the constraints faced by real streaming agents.

To construct StreamGaze, we develop a semi-automatic gaze-guided data generation pipeline that aligns gaze trajectories with egocentric videos and converts them into temporally grounded QA pairs. We first detect stable gaze moments (*i.e.*, fixations) throughout each video and extract objects within the corresponding gaze regions using region-specific visual prompting. Next, we capture the temporal dynamics of gaze (*i.e.*, scanpaths) to model how a user's attention shifts across spatial regions over time. Building on these signals, we automatically generate past, present, and proactive streaming tasks using an LLM, followed by human verification for quality assurance.

Across all StreamGaze tasks, we observe substantial performance gaps between state-of-the-art MLLMs (e.g., GPT-4o, InternVL-3.5) and both human performance and specialized streaming video understanding models. In particular, current MLLMs struggle to leverage gaze signals for temporal reasoning and proactive inference, revealing fundamental limitations in gaze-conditioned understanding. We further conduct detailed analyses of gaze-prompting strategies, gaze-based reasoning behaviors, and task-specific proactive responses on StreamGaze. Together, these evaluations position StreamGaze as the first comprehensive testbed for assessing gaze-driven causal reasoning and proactive prediction in streaming video scenarios.

Our contributions are threefold:

- We propose the first gaze-guided data construction pipeline that integrates gaze trajectories with egocentric video to produce spatio-temporally aligned, gaze-guided QA pairs. Our pipeline models full *scanpath* dynamics, tracking how attention evolves over time and how new objects enter the FOV through ego-motion, enabling truly streaming, temporally grounded supervision that static gaze–based pipelines cannot provide.
- We introduce StreamGaze, the first benchmark specifically designed for streaming gaze-guided video understanding, comprising 8521 QA pairs across 10 tasks spanning past, present, and proactive timestamps. This represents a larger and more diverse evaluation suite compared to existing streaming QA benchmarks.
- We evaluate state-of-the-art MLLMs on StreamGaze, uncovering substantial and consistent gaps relative to human performance. Our in-depth analyses of gaze-prompting strategies, reasoning patterns, and task-specific behaviors reveal that current MLLMs struggle to interpret raw gaze signals, overly rely on frame-local visual cues, and fail to generalize across different temporal reasoning requirements. These findings provide concrete design guidance for future gaze-aware streaming models.

Table 1. **Comparison of streaming video understanding benchmarks with StreamGaze.** MC: Multiple-choice; OE: Open-ended; All-time: Covers past, present, and future tasks; Proac.: Proactive. ✓ marks the presence of a feature, and △ indicates partial inclusion.

Dataset	Avg video (s)	QA pairs	QA type	Annotation	Proac.	Gaze	All-time	Ego.
Gaze-based QA benchmarks								
GazeVQA [7]	150	25040	MC	Human	✗	✓	✗	✓
EgoGazeVQA [19]	52	1757	MC	Auto	✗	✓	✗	✓
Streaming QA benchmarks								
StreamingBench [13]	268	4500	MC+OE	Auto & Human	✓	✗	✓	✗
SVBench [31]	147	7374	OE	Auto & Human	✗	✗	✗	△
OVO-Bench [15]	428	2814	MC+OE	Auto & Human	✓	✗	✓	△
OmniMMI [28]	324	2290	OE	Auto & Human	✓	✗	✗	△
ProAssist [36]	710	1020	OE	Auto	✓	✗	✗	✓
StreamBench [29]	270	1800	OE	Auto & Human	✗	✗	✗	✓
Vispeak-Bench [4]	21	1000	OE	Auto & Human	✓	✗	✗	✗
StreamGaze (Ours)	815	8521	MC+OE	Auto & Human	✓	✓	✓	✓

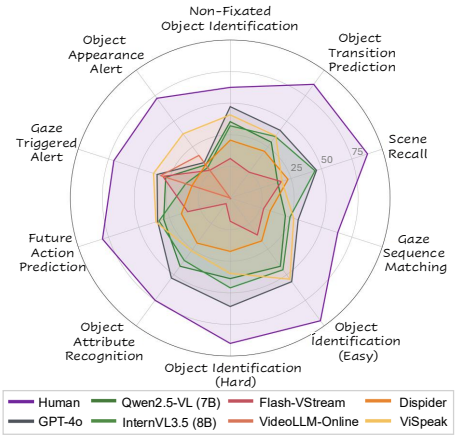


Figure 2. **MLLMs performance across StreamGaze tasks.**

2. Related Work

Streaming QA benchmarks. Streaming video understanding requires models to interpret and respond to temporally incoming frames without access to future context. StreamingBench [13] and OVO-Bench [15] benchmark online video understanding across past, present, and future tasks, providing a comprehensive evaluation of temporal reasoning capabilities. Meanwhile, OmniMMI [28], ProAssist [36], and ViSpeak-Bench [4] extend this paradigm to interactive dialogue settings, assessing how effectively MLLMs perceive user intentions and sustain coherent interactions in streaming contexts. In contrast, our proposed StreamGaze introduces *a novel dimension by integrating eye-gaze dynamics into the streaming setting*, enabling systematic evaluation of how models leverage human gaze for temporal reasoning, intention inference, and proactive prediction.

Gaze-based QA benchmarks. GazeVQA [7] introduces the first gaze-based VQA dataset designed for collaborative interactions in assembly processes. EgoGazeVQA [19] focuses primarily on intent understanding in short egocentric videos, but relies only on frame-wise gaze signals without extracting fixations or establishing spatio-temporal grounding for QA generation. Moreover, none of these works address the online setting that encompasses past, present, and proactive tasks, an essential component for real-world gaze-guided applications. In contrast, StreamGaze is the first benchmark to integrate gaze dynamics into the *streaming* setting, enabling systematic evaluation of how well MLLMs perform temporal reasoning and proactively align with human attention over time.

3. Gaze-Guided Streaming Data Construction

We design a novel data construction pipeline to build gaze-guided streaming video understanding tasks. In contrast to EgoGazeVQA [19], which processes short clips with static per-frame gaze points, our pipeline extracts the full scanpath, a sequence of fixations that captures how attention shifts over time and how objects enter the user's FOV (Field of View) through ego-motion. This enables streaming and temporally grounded data construction beyond static-gaze settings.

We first preprocess raw gaze trajectories by projecting them using camera parameters (Section 3.1) and extract stable fixation moments that capture meaningful, attention-driven events (Section 3.2). Next, we divide each frame into FOV and out-of-FOV regions and extract the corresponding objects. Finally, we construct scanpaths and conduct human verification—not only to ensure quality and consistency, but also to confirm that our pipeline's data construction closely correlates with human annotations (Section 3.3).

3.1. Preprocessing

We use three public egocentric video datasets spanning diverse domains: EGTEA+ (*cooking*) [11], EgoExoLearn (*cooking*, *lab*) [6], and HoloAssist (*assembly*) [27]. Each video is represented as a sequence of frames, $\mathcal{V} = \{v_t\}_{t=1}^{T}$, where T is the number of frames and v_t is the frame at time t. To obtain the gaze trajectory $\mathcal{G} = \{(x_t, y_t)\}_{t=1}^{T}$, where (x_t, y_t) denotes the gaze coordinate on the image plane corresponding to frame v_t, we project raw gaze in world coordinates onto the 2D image plane using officially provided camera parameters. For datasets that directly provide 2D gaze coordinates (e.g., [6, 11]), we use them as given.

3.2. Fixation Extraction

Based on the unified gaze trajectory $\mathcal{G}$ obtained in Sec. 3.1, we identify *query moments* within $\mathcal{V}$—time points that likely correspond to meaningful user attention—for streaming video QA tasks. This step is essential for gaze-guided QA generation, as it identifies the most informative moments at which user attention should be queried. We primarily target *fixation moments*, intervals where the gaze remains relatively stable within a localized region, as they more reliably capture users' visual attention compared to rapid eye shifts (*i.e.*,

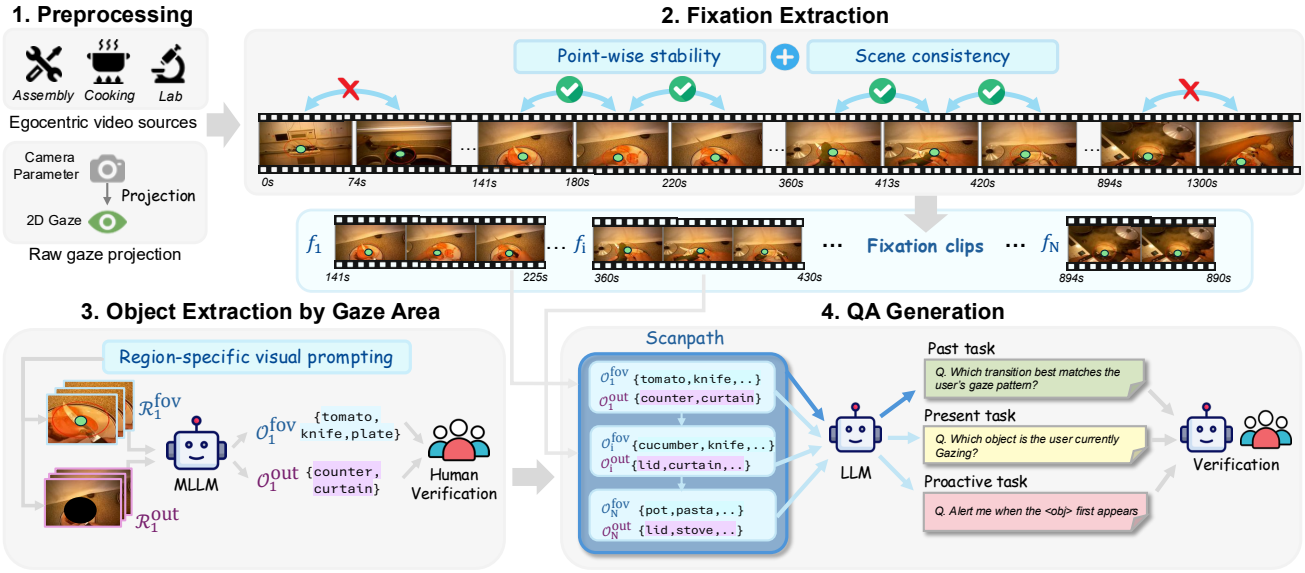


Figure 3. **Gaze-guided streaming data construction pipeline for StreamGaze.** Given egocentric video sources and raw gaze projections (Sec. 3.1), we first extract fixation moments across the entire video (Sec. 3.2). Next, we divide each frame into FOV and out-of-FOV regions and extract objects within the gaze area (Sec. 3.3). Finally, we construct scanpaths and generate streaming QA pairs (Sec. 4).

saccadic) [7, 8, 23].

From a raw gaze trajectory $\mathcal{G}$, we first identify fixation intervals. Each fixation is characterized by a spatial centroid $(\bar{x}_i, \bar{y}_i)$ and a temporal span $[t_i^s, t_i^e] \subseteq [1, T]$, where t_i^s and t_i^e denote the start and end timestamps. If N fixations are detected, the resulting set is written as

$$\mathcal{F} = \{f_i = (\bar{x}_i, \bar{y}_i, t_i^s, t_i^e)\}_{i=1}^{N}. \quad (1)$$

Each fixation is represented by its spatial centroid $(\bar{x}_i, \bar{y}_i)$ and temporal span $[t_i^s, t_i^e] \subseteq [1, T]$, corresponding to the start and end timestamps of the fixation. The spatial centroid is obtained by averaging all gaze points within the interval. To extract $\mathcal{F}$, we apply two criteria: *(i) point-wise stability* and *(ii) scene consistency*.

Point-wise stability. We identify $[t_i^s, t_i^e]$ as a fixation when the gaze points remain spatially concentrated and temporally stable. We enforce spatial stability by requiring that

$$d_t = \|(x_t, y_t) - (\bar{x}_i, \bar{y}_i)\|_2 \le r_{\text{thresh}}, \quad \forall t \in [t_i^s, t_i^e], \quad (2)$$

where r_{thresh} denotes the maximum spatial dispersion allowed around the fixation centroid $(\bar{x}_i, \bar{y}_i)$. To ensure comparability across datasets with different resolutions, we normalize r_{thresh} by the frame width. Furthermore, we ensure temporal stability by enforcing a minimum duration:

$$t_i^e - t_i^s \ge \tau_{\text{dur}}, \quad (3)$$

where τ_{dur} specifies the minimum time required for a valid fixation.

Scene consistency. Even if a fixation satisfies spatial and temporal stability, abrupt scene changes may occur due to camera motion or cuts. To ensure that each fixation corresponds to a visually continuous segment, we compute frame-wise Hue–Saturation histograms [22] H_t for all frames in $\mathcal{V}_i = \{v_t\}_{t=t_i^s}^{t_i^e}$ corresponding to fixation f_i. Each H_t is a normalized histogram over the hue and saturation channels of frame v_t. We then measure the minimum Pearson correlation between consecutive histograms:

$$S_{\min} = \min_{t \in [t_i^s, t_i^e - 1]} \rho(H_t, H_{t+1}), \quad (4)$$

where $\rho(\cdot)$ denotes the Pearson correlation between normalized histograms. A fixation is retained only if $S_{\min} \ge \tau_{scene}$, ensuring that scene-consistent fixations are preserved while discontinuous segments are discarded.

3.3. Object Extraction by Gaze Area

Given a fixation f_i, we extract objects according to their locations within or outside the FOV from fixation video clip $\mathcal{V}_i$ defined in Sec. 3.2. Our objective is to distinguish objects that lie *inside* the user's FOV from those *outside* it, enabling controllable task difficulty and gaze-grounded QA generation.

Definition of FOV and out-of-FOV regions. Motivated by classical eye-tracking literature, which models the foveal and parafoveal regions as circular areas around the gaze point and scales their pixel radius by screen resolution [7, 9, 16, 19, 24], we define the FOV region in each frame v_t as a circular patch:

$$\mathcal{R}_{i,t}^{\text{fov}} = \{(x, y) \mid \|(x, y) - (\bar{x}_i, \bar{y}_i)\|_2 \le \tau_{fov}\}, \quad (5)$$

where τ_{fov} denotes the FOV radius and (x, y) is pixel coordinate on frame v_t and $(\bar{x}_i, \bar{y}_i)$ is spatial centroid defined in Sec. 3.2. Following the normalized dispersion strategy commonly used for fixation modeling [7], we target a consistent FOV size across video sources, ensuring that FOV extraction captures comparable spatial extent regardless of video resolution. The remaining portion of the frame defines the out-of-FOV region:

$$\mathcal{R}_{i,t}^{\text{out}} = v_t \setminus \mathcal{R}_{i,t}^{\text{fov}}. \tag{6}$$

Across the entire fixation interval, these per-frame regions form two time-indexed sequences:

$$\mathcal{R}_i^{\text{fov}} = \{\mathcal{R}_{i,t}^{\text{fov}}\}_{t=t_i^s}^{t_i^e}, \qquad \mathcal{R}_i^{\text{out}} = \{\mathcal{R}_{i,t}^{\text{out}}\}_{t=t_i^s}^{t_i^e}. \tag{7}$$

Region-specific visual prompting. To extract objects from each region, we employ a MLLM (InternVL3.5-38B) with spatially guided visual prompts. For FOV region $\mathcal{R}_{i,t}^{\text{fov}}$, we crop a circular patch centered at $(\bar{x}_i, \bar{y}_i)$ with radius r_{fov}, and overlay a small red dot at the fixation center to explicitly indicate the user's point of gaze. This patch is then provided to the model as the inside-FOV visual input. For the out-of-FOV region $\mathcal{R}_{i,t}^{\text{out}}$, we take the original frame v_t and mask the circular FOV area by replacing all pixels within radius τ_{fov} with a solid black disk (see Fig. 3). This removes all gaze-relevant content while preserving the surrounding contextual background, ensuring that the model extracts only non-attended objects.

Given the region sequences $\mathcal{R}_i^{\text{fov}}$ and $\mathcal{R}_i^{\text{out}}$, the MLLM outputs two corresponding object sets:

$$\mathcal{O}_i^{\text{fov}} = \text{MLLM}(\mathcal{R}_i^{\text{fov}}), \qquad \mathcal{O}_i^{\text{out}} = \text{MLLM}(\mathcal{R}_i^{\text{out}}), \tag{8}$$

where each detected object is accompanied by a brief (1–2 sentence) caption used for downstream QA generation.

Scanpath generation. Based on the fixation-level object sets, we construct a scanpath $\mathcal{S}$ that represents the temporal evolution of gaze-guided object observations. This scanpath preserves the temporal order of fixations, thereby representing how attention shifts across different spatial regions and semantic contexts in the video. Given a sequence of N fixations $\mathcal{F}$ defined in Sec. 3.2, we define $\mathcal{S} = \{(\mathcal{O}_i^{\text{fov}}, \mathcal{O}_i^{\text{out}})\}_{i=1}^{N}$ where each element corresponds to the FOV and out-of-FOV object sets extracted from the fixation clip $\mathcal{V}_i$. For each fixation f_i, the corresponding object sets $\mathcal{O}_i^{\text{fov}}$ and $\mathcal{O}_i^{\text{out}}$ provide the local visual context, while the sequential arrangement of these fixations forms a gaze-conditioned trajectory over time. This ordered sequence serves as a structured representation of user perception, which will be used to generate temporally grounded QA pairs in subsequent steps.

Human verification. We finally employ human annotators to verify the generated scanpaths and extracted objects. Annotators review each fixation episode and determine whether each $\mathcal{O}_i^{\text{fov}}$ and $\mathcal{O}_i^{\text{out}}$ should be included or excluded. In particular, for $\mathcal{O}_i^{\text{fov}}$, annotators correct any mislabeled or missing objects and captions. Only human-verified scanpaths and object sets are used for the final benchmark. This verification process ensures high-quality yet scalable annotations, achieving an average correctness rate of approximately 83%. Please refer to Sec. A.5 for more details.

4. StreamGaze

We now introduce the StreamGaze QA generation process and its task taxonomy spanning past, present, and proactive tasks, designed to quantitatively evaluate gaze-guided streaming video understanding.

4.1. Benchmark Overview

Benchmark overview. StreamGaze comprises over 8,521 QA pairs from 285 videos. We also provide detailed statistics in Section D. As shown in Table 1, StreamGaze covers 10 tasks including past, present, and proactive. Also containing gaze modality corresponding with egocentric videos, representing realistic scenario.

Problem setup. We formulate a gaze-guided streaming video understanding task, where the model must answer time-sensitive questions while observing temporally incoming frames and user gaze. At a query time t_q, the MLLM $\mathcal{M}$ receives the video–gaze context $(\mathcal{V}, \mathcal{G})$ (see Sec. 3.1) within a specified temporal range and produces an answer A for a question Q. We denote the short temporal window by ω, where ω is 60 seconds following [13, 15]. Based on the accessible portion of the stream, we define novel gaze-based three streaming tasks as follows:

$$\begin{aligned}
\textbf{Past:} \quad & A = \mathcal{M}\big(Q;\, \mathcal{V}_{[0,\,t_q]},\, \mathcal{G}_{[0,\,t_q]}\big)\,, \\
\textbf{Present:} \quad & A = \mathcal{M}\big(Q;\, \mathcal{V}_{(t_q-\omega,\,t_q]},\, \mathcal{G}_{(t_q-\omega,\,t_q]}\big)\,, \\
\textbf{Proactive:} \quad & A = \mathcal{M}\big(Q;\, \mathcal{V}_{(t_q,\,\infty]},\, \mathcal{G}_{(t_q,\,\infty]}\big)\,.
\end{aligned}$$

4.2. Task Taxonomy

Given $\mathcal{S}$ from Sec. 3.3, we construct QA pairs that require spatio-temporal reasoning over gaze transitions.

4.2.1. Past Task

Past tasks focus on temporal reasoning over gaze and capture the dynamic characteristics of the user's gaze, modeling how attention shifts across objects over time.

- **Non-Fixated Object Identification (NFI):** Evaluates implicit visual awareness by identifying objects that were visible but never directly fixated. For each timestamp t, we sample one never-gazed object from $\mathcal{O}_t^{\text{out}}$ as the correct answer and three visible objects from $\mathcal{O}_t^{\text{fov}}$ as distractors. See Sec. A.4 for more details.
- **Object Transition Prediction (OTP):** Assesses temporal continuity in gaze behavior by predicting the next

object to be fixated. Given the current fixation on object group $\mathcal{O}_i^{\text{fov}}$ within the scanpath $\mathcal{S}$, the correct answer is the next newly attended object $\mathcal{O}_{i+1}^{\text{fov}}$, while distractors are sampled from the global object pool excluding the current group. See Sec. A.4 for more details.

- **Gaze Sequence Matching (GSM):** Measures how well models capture human-like scanpath patterns through sequential gaze transitions. We extract triplets of consecutive fixation groups $\mathcal{O}_{i-1}^{\text{fov}} \rightarrow \mathcal{O}_i^{\text{fov}} \rightarrow \mathcal{O}_{i+1}^{\text{fov}}$ as correct transitions, while negative options include shuffled orderings of the same groups or randomly sampled triplets that preserve transition structure.
- **Scene Recall (SR):** Tests contextual memory by recalling background objects previously visible during a fixation. At each fixation $\mathcal{O}_i^{\text{fov}}$, we sample the correct answer from background objects visible at other timestamps $\mathcal{O}_{\setminus i}^{\text{out}} = \bigcup_{t \neq i} \mathcal{O}_t^{\text{out}}$ but not at the current one, with three visible background objects as distractors.

4.2.2. Present Task

Present tasks capture the user's perceptual state and intention. We control question difficulty using both $\mathcal{O}_i^{\text{fov}}$ and $\mathcal{O}_i^{\text{out}}$.

- **Object Identification (OI, Easy/Hard):** Evaluates recognition of the currently attended object within the fixation region $\mathcal{O}_i^{\text{fov}}$. The correct answer corresponds to the fixated object itself. For the *easy* setting, distractors are randomly sampled from earlier object pools, while for the *hard* setting, distractors are chosen from visually similar $\mathcal{O}_i^{\text{out}}$ appearing in the same frame.
- **Object Attribute Recognition (OAR):** Assesses fine-grained perceptual understanding by predicting visual attributes (e.g., color, shape, or texture) of the currently fixated object within $\mathcal{O}_i^{\text{fov}}$. To avoid overlapping or ambiguous attribute options, we use Qwen3-VL-30B to generate distinct and contextually consistent distractors.
- **Future Action Prediction (FAP):** Models intention inference by predicting the user's next action from the recent gaze-conditioned context $\{\mathcal{O}_{i-k}^{\text{fov}}, \ldots, \mathcal{O}_i^{\text{fov}}\}$, where $k < i$ denotes the number of past fixation steps considered. Ground-truth future actions are derived from fine-grained human action captions aligned with gaze sequences in the original video sources. Semantically related but incorrect action descriptions are generated by Qwen3-VL-30B to serve as distractors.

4.2.3. Proactive Task

Proactive tasks involve gaze-based reasoning, where the model anticipates or assists future user behavior based on gaze and contextual cues. Unlike prior proactive settings [13, 15, 36], we leverage gaze as an anticipatory signal to predict upcoming attention or object events.

- **Gaze-Triggered Alert (GTA):** Triggers an alert when the user gaze a specified object within $\mathcal{R}_i^{\text{fov}}$. This task evaluates the model's ability to indicate user's fixation.
- **Object Appearance Alert (OAA):** Triggers an alert when the specified object first appears in the peripheral region $\mathcal{R}_i^{\text{out}}$. This task assesses whether the model can proactively recognize and react to newly emerging objects in streaming video.

Overall, we filter out ambiguous questions through both Qwen3-VL-30B–based validation and human verification. Detailed task-specific configurations and generation procedures are provided in Sec. A.4.

5. Experiments

5.1. Experimental Setup

Baselines. We evaluate four categories of existing models in a zero-shot setting on StreamGaze: (1) *Closed-source MLLMs*, including GPT-4o [17], Claude Sonnet4 and Opus4 [18]; (2) *Gaze-based models*, including AssistGaze [7]; (3) *Open-source MLLMs*, including Qwen2.5-VL [1], InternVL3.5 [26], MiniCPM-V [32], Kangaroo [14] and VITA-1.5 [3]; (4) *Open-source streaming MLLMs*, including Flash-VStream [34], VideoLLM-Online [2], Dispider [21] and ViSpeak [4]. Following prior streaming benchmarks [13, 15], because non-streaming models cannot ingest live video streams, we convert all streaming tasks into offline inference by providing each model with the corresponding video clip. We additionally report human oracle performance for all tasks.

Gaze input strategy. To incorporate gaze into each model, we employ a visual prompting strategy inspired by [19]. For every video, we overlay (i) a green dot indicating the user's gaze center and (ii) a red circular region representing FOV. We also prepend an instruction prompt that explicitly explains these cues to the model, e.g., *"The user's gaze center is indicated by a green dot. The red circle represents the area where the user's gaze is focused."* Full instruction templates are provided in the Sec. B.

Evaluation metrics. For past and present tasks, which are multiple-choice questions, we use Accuracy based on exact matching (with optional fuzzy matching for semantically equivalent responses). In particular, for VideoLLM-online [2], which generates free-form text responses, we applied keyword-based matching by checking if the ground truth option keyword appeared in the response using regular expressions. For proactive remind tasks, we adopt a multi-triggering query protocol following [13, 15] to simulate online decision-making. At each timestamp, the model is prompted with the cumulative video observed so far and must answer *"Yes"* if the target object has appeared in the field of view and *"No"* otherwise. We use accuracy as the primary metric, as it penalizes premature or excessive triggers and thus best reflects the reliability of proactive alerts. Additional recall results are provided in Sec. C.

Table 2. **Comparison of various MLLMs across StreamGaze tasks.** We evaluate four categories of models, reporting accuracy for past and present tasks and precision for proactive tasks. Gaze information is provided to each model through visual prompting. Each section is sorted by overall performance, and the best per task across all models (excluding Human) is **bolded**.

Method	Params	Frames	Past				Present				Proactive		Overall
			NFI	OTP	SR	GSM	OI (E)	OI (H)	OAR	FAP	GTA	OAA	
Human	-	-	0.700	0.889	0.903	0.707	0.960	0.920	0.800	0.840	0.765	0.780	0.827
Closed-Source MLLMs													
GPT-4o [17]	-	16	**0.601**	0.449	0.535	**0.580**	**0.729**	**0.730**	**0.596**	0.370	0.597	0.149	**0.535**
Claude Sonnet4 [18]	-	16	0.500	**0.554**	0.425	0.325	0.521	0.533	0.561	**0.439**	0.535	0.350	0.474
Claude Opus4 [18]	-	16	0.372	0.392	0.460	0.166	0.431	0.436	0.430	0.351	0.466	**0.490**	0.399
GazeQA-based Fine-tuned Models													
AssistGaze [7]	26M	32	0.294	0.131	0.310	0.294	0.278	0.250	0.109	0.254	N/A	N/A	0.223
Open-Source MLLMs													
Qwen2.5-VL [25]	7B	Adaptive	0.518	0.350	0.450	0.483	0.590	0.558	0.548	0.391	0.486	0.407	0.478
InternVL3.5 [26]	8B	Adaptive	0.490	0.311	**0.573**	0.548	0.627	0.628	0.466	0.372	0.373	0.051	0.444
VITA 1.5 [3]	7B	16	0.474	0.365	0.346	0.378	0.455	0.396	0.437	0.370	0.351	0.267	0.384
MiniCPM-V [32]	8B	32	0.430	0.374	0.354	0.296	0.334	0.379	0.438	0.345	0.480	0.216	0.365
Kangaroo [14]	7B	64	0.363	0.365	0.319	0.275	0.454	0.484	0.402	0.412	0.242	0.198	0.351
Open-Source Streaming MLLMs													
ViSpeak [4]	7B	1 fps	0.463	0.358	0.417	0.473	0.572	0.581	0.406	0.309	**0.635**	0.458	0.467
Dispider [21]	7B	1 fps	0.366	0.365	0.381	0.263	0.336	0.338	0.353	0.321	0.252	0.261	0.323
Flash-VStream [34]	7B	1 fps	0.249	0.202	0.336	0.220	0.289	0.147	0.044	0.280	0.443	0.217	0.243
VideoLLM-online [2]	8B	2 fps	0.000	0.000	0.000	0.000	0.006	0.006	0.002	0.000	0.458	0.333	0.080

5.2. Overall Performance

In Table 2, we report performances from all StreamGaze tasks from baseline MLLMs. Our evaluation brings several important findings regarding gaze-guided streaming video understanding as follows:

General-purpose MLLMs struggle to leverage gaze for long-term temporal reasoning. StreamGaze's past and present tasks require integrating gaze information over longer temporal windows, where models must aggregate multiple gaze-conditioned observations rather than rely on single-frame cues. Although they outperform traditional gaze-based models [7], open-source MLLMs still struggle to utilize gaze signals effectively. Even when provided with gaze as a visual prompt, their limited model capacity hinders the incorporation of gaze information during inference. Furthermore, these models often rely on key-frame or sparsely sampled inputs, causing them to process frames independently rather than as a continuously evolving stream. As a result, they fail to accumulate gaze evidence over time and cannot maintain coherent long-term temporal reasoning.

Streaming MLLMs struggle with gaze-guided proactive tasks. As observed in prior work [4, 15], dialogue-based streaming models such as VideoLLM-online [2] often fail to follow task instructions and instead produce generic descriptions (e.g., *"You look at the camera."*), due to their conversational training data. In contrast, non-dialogue streaming model, ViSpeak [4] outperforms open-source non-streaming MLLMs on proactive tasks, demonstrating the benefit of frame-by-frame online processing. However, other streaming MLLMs perform substantially worse, and all models remain far below human performance, indicating that robust gaze-guided proactive reasoning is still an open challenge. This limitation likely arises from the absence of explicit mechanisms for linking gaze shifts to predictive temporal reasoning within current streaming architectures.

Gaze-based model fails to generalize to streaming settings. Although explicitly designed to utilize gaze information, AssistGaze [7] achieves only 0.223 average accuracy on our benchmark. While such models can effectively exploit gaze cues in static or short-range scenarios, they struggle to maintain temporal coherence, integrate gaze shifts over time, and track evolving user intentions. Furthermore, since AssistGaze is trained on narrowly defined GazeVQA tasks, it lacks the flexibility to generalize to proactive or temporally extended streaming tasks.

A substantial gap remains between human and model performance. Humans achieve an average accuracy of 0.827, substantially higher than all evaluated MLLMs. In particular, the large disparity between Scene Recall (SR) and Object Transition Prediction (OTP) suggests that current models over-focus on foreground gaze objects while neglecting the surrounding scene context and struggle to model gaze transitions. This pronounced human–model gap underscores the need for future models to develop broader situational perception and richer temporal memory beyond object-centric cues.

Table 3. **Ablation of gaze input prompting.** We evaluate the effect of gaze input strategies on Qwen2.5-VL.

	Past	Present	Proactive	Avg.
Qwen2.5-VL [1] (w/o gaze)	**0.423**	0.500	0.384	0.446
+ test prompt	0.403	0.499	0.341	0.429
+ visual prompt	0.398	0.503	0.342	0.429
+ salience map	0.394	**0.546**	**0.386**	**0.454**

5.3. Additional Analysis

We provide further analyses to better understand the behavior and characteristics of StreamGaze.

Ablation of gaze input prompting. We analyze different strategies for incorporating gaze information into MLLMs using Qwen2.5-VL (7B), as shown in Table 3. Specifically, we compare three prompting methods—*textual prompt*, *visual prompt*, and *salience map*—to examine which representation most effectively enables the model to leverage gaze-specific cues, following [19]. In the textual prompt, fixation coordinates are provided alongside the question for each video clip. In the visual prompt, a green dot indicating the gaze center and a red circular region marking the FOV area are overlaid on each frame. In the salience-map prompt, the entire gaze trajectory is aggregated into a single heatmap, which is then provided as an additional image input. Among these methods, the salience-map prompt achieves the best performance, particularly on present and proactive tasks, suggesting that spatially aggregated cues are more compatible with the model than raw coordinates or frame-level overlays. However, none of the prompting strategies consistently outperform the gaze-free baseline, indicating that Qwen2.5-VL is not inherently equipped to interpret or reason over raw gaze signals. More details are in Sec. C.

Gaze-based reasoning in MLLMs. To investigate how MLLMs utilize gaze signals during inference, we ablate the contributions of text-, gaze-, and visual-based reasoning using GPT-4o in Table 4. For textual reasoning, we apply standard chain-of-thought prompting (e.g., *"Let's think step by step"*). For gaze-based reasoning, GPT-4o is first instructed to estimate the gaze point before generating an answer. For visual reasoning, the model is guided to identify visible objects with bounding boxes and use this visual evidence when answering. The best average performance is achieved when all reasoning strategies are combined. However, their effectiveness varies across tasks: incorporating visual cues benefits Scene Recall (SR) by improving contextual grounding, but can hinder Non-Fixated Object Identification (NFI), where excessive focus on gaze regions may suppress exploration of unseen objects. More details and qualitative examples are in Sec. C.

Task-Specific Behavior in StreamGaze. As shown in Figure 4 and Tab. 4, interestingly different StreamGaze tasks respond unevenly to the same input gaze or reasoning strategies. These patterns indicate that each StreamGaze task require distinct visual, temporal, and attentional requirements, and therefore a single, uniform prompting or reasoning strategy is insufficient. A more adaptive, task-aware, or dynamically integrated approach is needed to effectively model the full spectrum of streaming gaze understanding.

Table 4. **Ablation of different reasoning strategies.** We evaluate the effectiveness of text, gaze, and visual reasoning on GPT-4o.

Reasoning			Past		Present		
Text	Gaze	Visual	NFI	SR	OI (Easy)	FAP	Avg.
			0.52	0.60	0.64	0.41	0.542
✓			0.10	0.62	0.68	**0.47**	0.467
✓	✓		0.36	0.62	0.72	0.40	0.525
✓	✓	✓	0.44	**0.66**	**0.72**	0.44	**0.565**

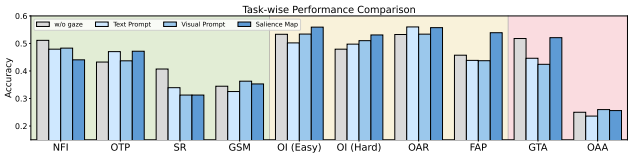


Figure 4. **Task-wise effect of gaze input strategies on StreamGaze.** Different tasks respond unevenly to textual, visual, and salience-map prompts, revealing strong task-specific behavior in streaming gaze understanding.

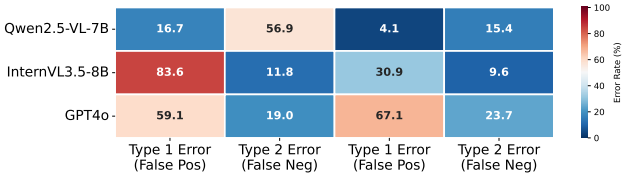


Figure 5. **Offline MLLMs' behavior on StreamGaze proactive tasks.** We evaluate type1 (false positive) and type2 (false negative) error from Qwen2.5-VL, InternVL-3.5, and GPT4o.

Proactive Behavior of MLLMs. In Figure 5, we analyze how offline MLLMs behave on StreamGaze proactive remind tasks by measuring Type 1 (false positive) and Type 2 (false negative) error rates across each task. This evaluation reveals strong model-specific biases: InternVL3.5-8B consistently over-triggers with very high false positive rates, GPT4o deteriorates as tasks become harder with rising errors of both types, while Qwen2.5-VL-7B shows the most balanced pattern, remaining accurate in hard settings despite conservative behavior on easy ones. These results demonstrate that proactive gaze prediction is highly sensitive to model characteristics and underscore the need to examine error-type distributions when designing reliable gaze-based assistive systems.

6. Conclusion

In this work, we introduce StreamGaze, the first benchmark for evaluating gaze-guided temporal and proactive rea-

soning in streaming video understanding. Our gaze–video QA pipeline provides human-aligned supervision across diverse past, present, and proactive tasks. Experiments reveal substantial gaps between state-of-the-art MLLMs and human annotators, especially in intention modeling and proactive prediction. We hope StreamGaze advances the development of more human-aligned, attention-aware streaming models.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6, 8

[2] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024. 2, 6, 7

[3] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, et al. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025. 6, 7

[4] Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. *arXiv preprint arXiv:2503.12769*, 2025. 2, 3, 6, 7, 5

[5] 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. 2, 6

[6] Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, et al. Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22072–22086, 2024. 3, 2, 5, 6

[7] Muhammet Ilaslan, Chenan Song, Joya Chen, Difei Gao, Weixian Lei, Qianli Xu, Joo Lim, and Mike Shou. Gazevqa: a video question answering dataset for multiview eye-gaze task-oriented collaborations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10462–10479, 2023. 2, 3, 4, 5, 6, 7

[8] Shun Inadumi, Seiya Kawano, Akishige Yuguchi, Yasutomo Kawanishi, and Koichiro Yoshino. A gaze-grounded visual question answering dataset for clarifying ambiguous japanese questions. *arXiv preprint arXiv:2403.17545*, 2024. 4

[9] Lisa M Kroell and Martin Rolfs. Foveal vision anticipates defining features of eye movement targets. *Elife*, 11:e78106, 2022. 2, 4

[10] 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. 2

[11] Yin Li, Miao Liu, and James M Rehg. In the eye of the beholder: Gaze and actions in first person video. *IEEE transactions on pattern analysis and machine intelligence*, 45(6): 6731–6747, 2021. 2, 3, 5

[12] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2

[13] Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*, 2024. 3, 5, 6

[14] Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024. 6, 7

[15] Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913, 2025. 3, 5, 6, 7, 4

[16] Antje Nuthmann and Teresa Canas-Bajo. Visual search in naturalistic scenes from foveal to peripheral vision: A comparison between dynamic and static displays. *Journal of Vision*, 22(1):10–10, 2022. 2, 4

[17] OpenAI. Gpt-4 technical report, 2024. 6, 7

[18] Anthropic PBC. Introducing claude 4. Online blog post, 2025. https://www.anthropic.com/news/claude-4 (accessed 2025-11-14). 6, 7

[19] Taiying Peng, Jiacheng Hua, Miao Liu, and Feng Lu. In the eye of mllm: Benchmarking egocentric video intent understanding with gaze-guided prompting. *arXiv preprint arXiv:2509.07447*, 2025. 2, 3, 4, 6, 8, 5

[20] Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Kumar Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, et al. Hd-epic: A highly-detailed egocentric video dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23901–23913, 2025. 5, 6

[21] Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24045–24055, 2025. 6, 7

[22] Nisreen I. Radwan, Nancy M. Salem, and Mohamed I. El Adawy. Histogram correlation for video scene change detection. In *Advances in Computer Science, Engineering & Applications: Proceedings of the Second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012), May 25–27, 2012, New Delhi, India, Volume 1*, pages 647–655, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 4

[23] Jun Rekimoto. Gazellm: Multimodal llms incorporating human visual attention, 2025. 4

[24] Emma EM Stewart, Matteo Valsecchi, and Alexander C Schütz. A review of interactions between peripheral and foveal vision. *Journal of vision*, 20(12):2–2, 2020. 2, 4

[25] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. 7

[26] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 2, 6, 7

[27] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20270–20281, 2023. 3, 1, 2, 5

[28] Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. Omnimmi: A comprehensive multimodal interaction benchmark in streaming video contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18925–18935, 2025. 2, 3

[29] Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv preprint arXiv:2501.13468*, 2025. 2, 3

[30] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 3, 4

[31] Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. *arXiv preprint arXiv:2502.10810*, 2025. 2, 3

[32] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 6, 7

[33] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2

[34] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams, 2024. 2, 6, 7

[35] Mengmi Zhang, Keng Teck Ma, Joo Hwee Lim, Qi Zhao, and Jiashi Feng. Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4372–4381, 2017. 2

[36] Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. Proactive assistant dialogue generation from streaming egocentric videos. *arXiv preprint arXiv:2506.05904*, 2025. 2, 3, 6

StreamGaze: Gaze-Guided Temporal Reasoning and Proactive Understanding in Streaming Videos

Supplementary Material

A. StreamGaze Data Construction Details

In this section, we elaborate details about StreamGaze data construction pipeline. We first introduce details of gaze projection (Sec. A.1) during preprocessing Sec. 3.1 and fixation extraction (Sec. A.2). Then, we introduce object extraction details with prompt (Sec. A.3) and QA generation details by each task (Sec. A.4). Finally, we introduce human annotation process details (Sec. A.5). We also provide the full data construction procedure in Algorithm 1.

A.1. Gaze Projection

To obtain frame-level gaze positions on the egocentric RGB video, we convert each 3D gaze ray into a 2D pixel coordinate using camera poses and calibrated intrinsics for the HoloAssist dataset [27]. This method follows the hand–eye projection procedure implemented in official code.[1] Each gaze sample provides a world-coordinate origin $o(t)$ and direction $d(t)$, while the camera stream provides per-frame extrinsic matrices and intrinsics.

Temporal alignment. All signals, including video frames, camera poses, and gaze samples, share a common timestamp domain. For each video frame at time t, we select the nearest timestamped camera pose and gaze measurement:

$$(o_t, d_t) = (o(t^*), d(t^*)) \quad \text{where} \quad t^* = \arg\min_{\tau} |\tau - t|.$$

This nearest-neighbor alignment ensures consistent pairing despite differing sampling rates.

3D gaze point construction. Given the aligned 3D gaze ray, we form a virtual fixation point by moving a fixed distance d_{eye} along the normalized gaze direction:

$$p_t^{\text{world}} = o_t + d_{\text{eye}} \frac{d_t}{\|d_t\|}.$$

We treat p_t^{world} as the gaze target for frame v_t.

World-to-camera transformation. Let $T_{\text{cam}}^{\text{world}}(t)$ be the camera pose associated with frame v_t. We convert the world

Algorithm 1 Gaze-Guided Data Construction Pipeline

Input: $\mathcal{V} = \{v_t\}_{1:T}$, $\mathcal{G} = \{(x_t, y_t)\}_{t=1}^{T}$, $r_{\text{thresh}}, \tau_{\text{dur}}, \tau_{\text{scene}}$

`# Fixation Extraction`
$\mathcal{F} \leftarrow \emptyset$
for each candidate interval $[t_i^s, t_i^e]$ **do**
  Compute centroid $(\bar{x}_i, \bar{y}_i)$
  if $\forall\, t \in [t_i^s, t_i^e],\ \|(x_t, y_t) - (\bar{x}_i, \bar{y}_i)\|_2 \le r_{\text{thresh}}$ **then**
    if $t_i^e - t_i^s \ge \tau_{\text{dur}}$ **then**
      $S_{\min} \leftarrow \min_{t=t_i^s}^{t_i^e - 1} \rho(H_t, H_{t+1})$
      if $S_{\min} \ge \tau_{\text{scene}}$ **then**
        $\mathcal{F} \leftarrow \mathcal{F} \cup \{(\bar{x}_i, \bar{y}_i, t_i^s, t_i^e)\}$
      end if
    end if
  end if
end for

`# Object and scanpath extraction`
for each $f_i \in \mathcal{F}$ **do**
  for $t = t_i^s \dots t_i^e$ **do**
    Define $\mathcal{R}_{i,t}^{\text{fov}}$ and $\mathcal{R}_{i,t}^{\text{out}}$
  end for
  $\mathcal{O}_i^{\text{fov}} \leftarrow \text{MLLM}(\mathcal{R}_i^{\text{fov}})$
  $\mathcal{O}_i^{\text{out}} \leftarrow \text{MLLM}(\mathcal{R}_i^{\text{out}})$
end for
$\mathcal{S} \leftarrow \{(\mathcal{O}_i^{\text{fov}}, \mathcal{O}_i^{\text{out}})\}_{i=1}^{N}$

`# QA Generation`

[1] https://github.com/taeinkwon/PyHoloAssist/blob/main/hand_eye_project.py

point into camera coordinates via

$$p_{t,\text{cam}} = T_{\text{axis}} \left(T_{\text{cam}}^{\text{world}}(t)\right)^{-1} \begin{bmatrix} p_t^{\text{world}} \\ 1 \end{bmatrix}.$$

Pinhole projection. With the intrinsic matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix},$$

we project the 3D camera-coordinate point (X_t, Y_t, Z_t) onto the image plane:

$$u_t = \frac{X_t}{Z_t} f_x + c_x, \quad v_t = \frac{Y_t}{Z_t} f_y + c_y.$$

A point is marked as in-frame if (u_t, v_t) lies within the image boundaries.

Output. For each frame v_t, the resulting projected coordinates (u_t, v_t) define the 2D gaze position on the image plane. We therefore set $(x_t, y_t) = (u_t, v_t)$ to construct the gaze trajectory $\mathcal{G} = \{(x_t, y_t)\}_{t=1}^{T}$ used in the main paper. In addition, we store the original gaze validity flag and an in-frame indicator. For visualization, we simply draw a dot at (x_t, y_t) on the RGB frame.

A.2. Fixation Extraction

Handling short gaze interruptions. Human gaze often contains brief microsaccades or flicks that momentarily violate spatial stability. To avoid prematurely terminating a fixation, we permit short interruptions within a fixation interval. For each timestamp t, if the gaze deviation exceeds the spatial threshold ($d_t > \tau_{\text{thresh}}$), we examine the temporal gap between consecutive gaze samples:

$$\Delta t = t - (t-1).$$

If $\Delta t \leq 200ms$ (a short interruption threshold), the deviation is treated as a transient flick rather than a fixation termination, and the fixation interval continues without interruption.

Scene consistency. To ensure that each detected fixation corresponds to a visually coherent segment, we further apply a scene-consistency filter based on frame-wise Hue–Saturation histogram similarity. For a fixation interval $f_i = (\bar{x}_i, \bar{y}_i, t_i^s, t_i^e)$, we uniformly sample N frames from the corresponding video subsequence $\mathcal{V}_i = \{v_t\}_{t=t_i^s}^{t_i^e}$ and convert each sampled frame into the HSV color space. For every frame v_t, we compute a normalized 2D histogram over the Hue–Saturation channels:

$$H_t(h,s) = \text{Hist}^{HSV}(v_t(h,s)), \quad h \in [0,180],\ s \in [0,256],$$

with $\sum_{h,s} H_t(h,s) = 1$.

Next, we measure visual continuity between consecutive frames by computing the histogram correlation using Pearson's coefficient:

$$\rho(f_i(t), f_i(t+1)) = \frac{\sum_{h,s}(H_t(h,s) - \mu_t)\,(H_{t+1}(h,s) - \mu_{t+1})}{\sqrt{\sum_{h,s}(H_t(h,s) - \mu_t)^2}} \times \frac{1}{\sqrt{\sum_{h,s}(H_{t+1}(h,s) - \mu_{t+1})^2}}. \tag{9}$$

where μ_t and μ_{t+1} denote the mean values of H_t and H_{t+1}.

Collecting similarities across the entire episode gives

$$S = \{\rho(f_i(t_1), f_i(t_2)),\ \ldots,\ \rho(f_i(t_{N-1}), f_i(t_N))\}. \tag{10}$$

$$S_{\min} = \min(S). \tag{11}$$

A fixation is retained only if

$$S_{\min} \geq \tau_{\text{scene}},$$

where τ_{scene} is an empirically set threshold (typically $\tau_{\text{scene}} \approx 0.9$). Intervals failing to satisfy this constraint are considered to include scene changes and are discarded.

A.3. Object Extraction by Gaze Area.

FOV definition. When camera intrinsics are available [27], we compute the horizontal field-of-view (HFOV) using the intrinsic matrix K and frame image width W_{pixels}:

$$HFOV_{\text{rad}} = 2\arctan\left(\frac{W_{\text{pixels}}}{2f_x}\right)$$

$$HFOV_{\text{deg}} = HFOV_{\text{rad}} \cdot \frac{180}{\pi}$$

If intrinsics are unavailable [5, 6, 11], we assume a canonical $HFOV_{\text{deg}} = 90°$. Following standard visual-angle conventions in [9, 16, 24], the perifoveal region spans approximately 5°–15°. For our FOV threshold, we adopt the upper perifoveal bound and set $r_{\text{deg}} = 15°$, which specifies the angular radius that is later converted into the pixel threshold τ_{fov}. We then convert this angular radius into a pixel radius via

$$\text{px/deg} = \frac{W_{\text{pixels}}}{HFOV_{\text{deg}}}, \qquad \tau_{\text{fov}} = r_{\text{deg}} \times \text{px/deg}.$$

This τ_{fov} serves as the pixel threshold used in Sec. 3.3 to determine whether an object lies inside or outside the user's FOV.

Region-specific visual prompting. To extract objects conditioned on human gaze, we design region-specific visual prompting strategies for InternVL-3.5 (38B). As shown in Fig. 17 and Fig. 18, we provide two distinct prompts: one

for identifying objects within the user's FOV and another for discovering objects located outside the FOV.

For the in-FOV setting, we provide the model with the fixation-aligned video clip and explicitly instruct it to (i) generate a brief scene-level description, (ii) identify the precisely gazed object indicated by the green dot, and (iii) describe additional objects located within the perifoveal region. The prompt enforces consistent naming by providing an object pool and requires detailed, attribute-rich captions for each recognized object. For the out-of-FOV setting, we mask the central region of the frame and prompt the model to detect only objects outside the masked area. The prompt includes similar constraints on naming, appearance grounding, and relational descriptions, while removing all references to the gazed object. Both prompts further include human-centric rules—e.g., describing persons only when their full or upper body is visible, and requiring identity strings such as "person wearing [clothing description]" rather than generic labels.

These region-specific prompts guide the MLLM to reliably extract spatially grounded object sets that are aligned with gaze-conditioned context, enabling consistent FOV-aware and out-of-FOV object discovery across datasets.

A.4. QA Generation and Filtering

In this section, we provide details of QA generation and filtering for each StreamGaze task. We also elaborate how we decide query time t_q for each task.

A.4.1. Non-Fixated Object Identification (NFI)

NFI evaluates whether a model can identify objects that were visible in the scene but never directly fixated by the user.

QA generation and filtering. We collect all FOV/out-FOV objects and define the scene-wide object pool as $\mathcal{A} = \bigcup_i (\mathcal{O}_i^{\text{fov}} \cup \mathcal{O}_i^{\text{out}})$. Objects that were visible yet never fixated are defined as $\mathcal{N}_i = \mathcal{A} \setminus \mathcal{O}_i^{\text{fov}}$. For each index i, if $\mathcal{N}_i$ is non-empty, we sample the correct answer from $\mathcal{N}_i$ and choose three distractors from the prefix FOV object set $\mathcal{O}_{\leq i}^{\text{fov}}$. Since NFI does not require visual verification, we rely directly on human-verified scanpaths and perform only formatting and consistency checks.

Query time. Let f_i denote the set of distinct fixation objects observed up to timestamp i as defined in Sec. 3.2. We define the query timestamp t_q as the earliest time at which three unique fixation objects have been accumulated:

$$t_q = \min\{\, i \mid |f_i| \geq 3 \,\}.$$

At this point, we determine which objects have already been seen and which have not. To compute the response window, we gather all intervals in which the distractor objects (i.e., previously fixated objects) appear within the FOV. We then take the earliest and latest of these intervals and expand the window by two seconds on both ends, producing a consistent response interval that fully captures the temporal range in which distractor objects are visible.

A.4.2. Scene Recall (SR)

SR evaluates contextual memory by determining which background object was previously visible but not visible during the current fixation.

QA generation and filtering. We construct the global background pool as $\mathcal{B} = \bigcup_i \mathcal{O}_i^{\text{out}}$. At fixation i, the correct answer is sampled from $\mathcal{B} \setminus \mathcal{O}_i^{\text{out}}$, while distractors are chosen from the currently visible background objects $\mathcal{O}_i^{\text{out}}$. For filtering, we apply three consistency checks with Qwen3-VL: (1) the question must be clear and unambiguous, (2) the selected answer must be an object that was never fixated, and (3) the answer object must appear in the video.

Query time. We define the query timestamp t_q as the moment when the user first begins to fixate on any of the objects mentioned in the question. This ensures that the model is evaluated exactly when the relevant scene context becomes available to the user.

A.4.3. Object Transition Prediction (OTP)

OTP evaluates temporal continuity in gaze by predicting the next newly attended object.

QA generation and filtering. For consecutive timestamps $(i, i+1)$, we use $\mathcal{O}_i^{\text{fov}}$ and $\mathcal{O}_{i+1}^{\text{fov}}$. The correct answer is the first object in $\mathcal{O}_{i+1}^{\text{fov}}$ that does not belong to $\mathcal{O}_i^{\text{fov}}$, corresponding to the newly attended object. Distractors are sampled from the global object pool, excluding both $\mathcal{O}_i^{\text{fov}}$ and the correct answer. For filtering, since OTP transitions come directly from human-verified scanpaths, no additional visual filtering is required.

Query time. We define the query timestamp t_q as the moment when the fixation on the current object group $\mathcal{O}_i^{\text{fov}}$ first begins, marking the onset of the transition. To compute the response window, we use the full temporal span of the transition event: we start two seconds before $\mathcal{O}_i^{\text{fov}}$ becomes visible in the FOV and end two seconds after the fixation on the newly attended object $\mathcal{O}_{i+1}^{\text{fov}}$ concludes. This window provides continuous context across the transition, capturing both the preceding fixation pattern and the complete fixation period of the next object.

A.4.4. Gaze Sequence Matching (GSM)

GSM evaluates whether models capture sequential gaze patterns across time.

QA generation and filtering. We extract consecutive 3-step fixation sequences $[\mathcal{O}_i^{\text{fov}}, \mathcal{O}_{i+1}^{\text{fov}}, \mathcal{O}_{i+2}^{\text{fov}}]$ and serialize them as $\mathcal{O}_i^{\text{fov}} \rightarrow \mathcal{O}_{i+1}^{\text{fov}} \rightarrow \mathcal{O}_{i+2}^{\text{fov}}$ to form the correct sequence. Three negative options are generated: one using a shuffled ordering of the same groups, and two using random groups sampled from the global object pool while preserving structural validity. To prevent ambiguous QA pairs, we automatically remove samples with Qwen3-VL [30] where any distractor sequence overlaps with the correct sequence in more than 50% of segments.

Query time. We define the query timestamp t_q as the end of the third fixation group $\mathcal{O}_{i+2}^{\text{fov}}$, which marks the completion of the three-step transition sequence used for the question. The response window spans the entire sequence interval, beginning from the onset of the first fixation group $\mathcal{O}_i^{\text{fov}}$ and extending to the end of the third group $\mathcal{O}_{i+2}^{\text{fov}}$. This window provides continuous temporal context for evaluating whether the model can recognize the correct sequential gaze pattern across all three segments.

A.4.5. Object Identification (OI, Easy/Hard)

OI (Easy/Hard) evaluates whether models can correctly recognize the object currently fixated within the gaze-aligned cropped region $\mathcal{O}_i^{\text{fov}}$.

QA generation and filtering. In the easy task, distractors are sampled from all objects appearing anywhere in the video, excluding the target object and all objects within the cropped FOV region. In the hard task, distractors are drawn exclusively from $\mathcal{O}_i^{\text{out}}$, i.e., objects visible in the same frame but outside the fixation region. If fewer distractors than required are available, the corresponding sample is discarded to avoid trivial or ambiguous questions.

Query time. Each prediction is evaluated within the response interval aligned with the corresponding fixation episode. We therefore define the query timestamp t_q as the start time of the fixation.

A.4.6. Object Attribute Recognition (OAR)

OAR evaluates fine-grained perceptual understanding by asking the model to infer a specific visual attribute (e.g., color, material, shape, texture, size, or state) of the object currently fixated.

QA generation and filtering. To construct reliable attribute questions, we employ a constrained LLM-based template system. We define a fixed dictionary of attribute types and pair each type with a corresponding question template (e.g., color $\rightarrow$ "What color is this object?"). Given the caption of $\mathcal{O}_i^{\text{fov}}$, Qwen3-VL-30B [30] is prompted to: (1) select the most appropriate attribute type, (2) extract a concise correct answer grounded in the caption, and (3) generate three visually plausible but incorrect distractors that do not overlap semantically with the correct attribute. To remove ambiguous or semantically entangled distractors, we apply an additional verification step using Qwen3-VL [30], retaining only unambiguous multiple-choice sets.

Query time. Similar to the OI task, we define the query timestamp t_q as the start time of the fixation.

A.4.7. Future Action Prediction (FAP)

FAP evaluates proactive intention inference by predicting the user's upcoming action based on the recent sequence of fixations.

QA generation and filtering. For each fixation index i, we extract a 3-step fixation sequence capturing the short-term progression of the user's attention. We then align this sequence with the official action annotations of the video and identify the earliest future action whose timestamp occurs at least a small temporal margin (3 seconds to 1 minute) after the fixation ends. This action is assigned as the correct label. Distractor options are sampled from other actions within the same video, while removing near-duplicate or semantically ambiguous actions to ensure clear discrimination.

Query time. We define the query timestamp t_q as the moment when the fixation sequence ends, ensuring that the model makes predictions immediately after observing all relevant gaze information. The response window begins two seconds before the actual action timestamp, preventing temporal leakage while anchoring the prediction to the correct action segment.

A.4.8. Gaze-Triggered Alert (GTA)

GTA evaluates whether a model can proactively monitor the user's fixation and trigger an alert as soon as a specified target object enters the fixation region $\mathcal{R}_i^{\text{fov}}$. For each video, we first gather all objects that were directly fixated at least once, excluding background furniture or static structures using Qwen3-based filtering [30]. We then sample multiple evaluation timestamps relative to the object's first fixation moment (e.g., $t-20$, $t-10$, t, $t+10$, $t+20$ seconds), following the multi-query design of OVO-Bench [15]. At each timestamp, we determine whether the user is currently fixating on the object and assign a binary label accordingly (type 1 for fixation, type 0 for non-fixation). For quality control, we manually verify the fixation labels to ensure accuracy.

A.4.9. Object Appearance Alert (OAA)

OAA evaluates whether the model can proactively detect objects that appear in the peripheral region $\mathcal{R}_i^{\text{out}}$, i.e., objects that are visible in the frame but have not yet entered the user's fixation. We extract all $\mathcal{O}_i^{\text{out}}$ at each timestamp, removing background furniture or static structures using Qwen3-based filtering [30]. For each selected object, we generate evaluation timestamps (e.g., $t-20$, $t-10$, t, $t+10$, $t+20$ seconds) relative to its first appearance, using offsets centered around the event boundary. At each timestamp, we determine whether the object is currently visible in the frame but outside the fixation region, assigning type 1 when present and type 0 otherwise. For quality control, we use Qwen3-VL [30] to confirm that the object indeed appears within the 10-second evaluation window and additionally perform manual verification to ensure accuracy.

A.5. Human Verification

As shown in Fig. 22, we employ three annotators and provide them with an HTML-based interface for verification.

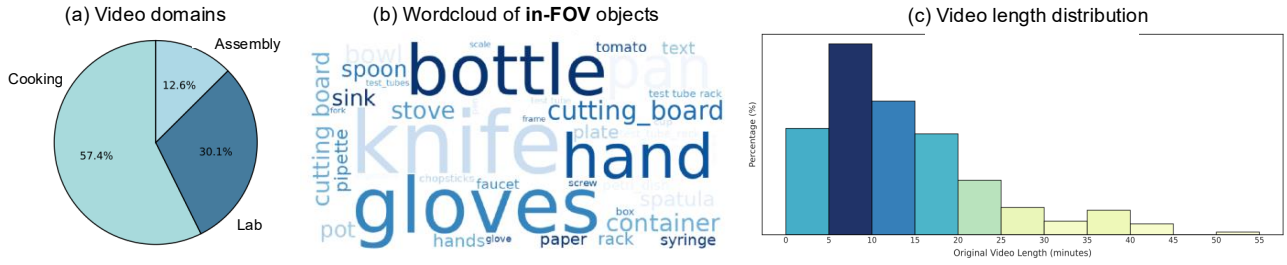


Figure 6. **Data statistics of StreamGaze.** We report domain proportion, video length, world cloud for FOV extracted objects.

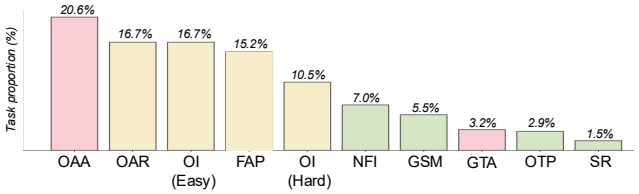


Figure 7. **Task proportion of StreamGaze.** We summarize the sample distribution for each task in StreamGaze.

Table 5. **Human verification results per each video source.** We report object inclusion/modification ratio after human verification.

	Inclusion Ratio (%)	Modified Ratio (%)
EGTEA-Gaze [11]	81.77	6.89
HoloAssist [27]	67.88	9.99
EgoExoLearn [6]	84.31	7.24

Each annotator is shown the fixation-level video clip along with the objects extracted by the MLLM: the precisely gazed object (marked by a green dot), other objects within the FOV region, and out-of-FOV objects. (Details of the provided instructions are shown in Fig. 24.) Annotators then determine whether each object should be included or excluded based on the visual evidence. For the gazed (pointing) object, annotators additionally correct or rewrite its identity and detailed caption when necessary. We report the inclusion ratio and modification ratio in Tab. 5. The inter-annotator agreement, measured using Fleiss' Kappa, is 0.60, indicating consistency among annotators.

B. StreamGaze Evaluation Details

Multiple-choice question evaluation. For all past and present tasks, we construct the model's input clip according to the prior streaming benchmarks [13, 15]. Past tasks receive the full history from the beginning of the video up to the query timestamp $[0, t_q]$, enabling long-range reasoning. Present tasks use a 60-second sliding window $[t_q - 60, t_q]$ (clamped to start at 0), reflecting real-time processing with limited recent context. All these tasks are evaluated as four-way multiple-choice questions. Because models may output answers in diverse natural-language formats, we extract the predicted option by deterministic parsing rules, prioritizing concise outputs such as a final standalone letter or expressions like "the answer is B," while also supporting formats such as "C. pot." Accuracy is reported as the proportion of correctly answered questions over the total number of questions for each task.

Proactive evaluation. The two proactive tasks follow an OVO-Bench–style configuration [15], where evaluation is conducted at a series of checkpoints r_t. Each r_t corresponds to an evaluation timestamp sampled around the target object's first fixation time (e.g., $t-20$, $t-10$, t, $t+10$, $t+20$ seconds), representing moments at which a proactive assistant should determine whether the user is attending to the target object. At each checkpoint, the model receives the cumulative video prefix $[0, r_t]$ and outputs a binary "yes/no" decision indicating whether the object lies within the fixation region. Each checkpoint is labeled as a positive case if the target is fixated and negative otherwise, and we compute accuracy across all checkpoints. This setup evaluates whether a model can continuously monitor gaze signals and trigger alerts with precise temporal fidelity.

C. Additional Analysis

C.1. Fine-tuning Results

To better understand the underlying failure of MLLMs and assess whether performance can be improved through gaze-supervised learning, we conduct LoRA-based fine-tuning experiments. Specifically, we fine-tune ViSpeak [4] on our gaze-guided training set to quantify the potential performance gains achievable through targeted adaptation.

Fine-tuning dataset. We construct four variants of training data for our fine-tuning experiments, drawing from both external resources and our own automatically generated data. First, we include external gaze-based QA datasets such as HD-EPIC [20] (2k) and EgoGazeVQA [19] (1.2k), which provide high-quality gaze–question–answer supervision despite not being strictly streaming-based. Second,

Table 6. **Fine-tuning results on ViSpeak [4].** We construct four variants of training data for our fine-tuning experiments, drawing from both external resources [19, 20] and our own automatically generated data based on Ego4D-Gaze [5] and EgoExoLearn [6].

Fine-tuning datasets			Past				Present				Proactive		Overall
External [19, 20]	EgoExoLearn [6]	Ego4D-Gaze [5]	NFI	OTP	SR	GSM	OI (E)	OI (H)	OAR	FAP	GTA	OAA	
			0.526	0.486	0.341	0.418	0.635	0.477	0.413	0.489	0.504	0.502	0.479
✓			0.467	0.477	0.300	0.347	0.368	0.562	0.420	0.392	0.147	0.381	0.386
✓	✓		0.592	0.563	0.469	0.385	0.622	0.616	0.544	0.359	0.474	0.792	0.542
✓		✓	0.517	0.545	0.495	0.373	0.540	0.452	0.496	0.351	0.487	0.737	0.500
✓	✓	✓	0.583	0.540	0.495	0.388	0.601	0.641	0.575	0.331	0.409	0.651	0.521

we leverage QA pairs generated by our data construction pipeline. Because the scanpath inclusion ratio between our pipeline and human annotators is approximately 78% (see Tab. 5), we further apply the same pipeline to EgoExoLearn [6] and Ego4D-Gaze [5], which do not overlap with StreamGaze video sources, to automatically synthesize additional StreamGaze-style training data. This serves as a form of data augmentation, expanding coverage and increasing the diversity of gaze-conditioned supervision.

Implementation details. The base model is ViSpeak, built on Qwen2.5-Instruct with an InternViT-300M vision tower and a frozen VITA-1.5 audio encoder. We enable LoRA with rank 128 and $\alpha = 256$, and train the MLP-based multimodal projector with a learning rate of 1×10^{-5}. We use a batch size of 2 per device (gradient accumulation 2), a maximum sequence length of 5500 tokens, and train for 2 epochs.

Negative sampling for proactive task fine-tuning

```
{
"conversations": [
{"from": "human",
"value": "Monitor and alert when I gaze <screen>",
"time": 91.0},
{"from": "gpt", "value": "", "time": 2783.0},
{"from": "gpt", "value": "", "time": 2793.0},
{"from": "gpt",
"value": "You are now gazing <screen>.",
"time": 2803.0}
],
  "proactive": true
}
```

Figure 8. **Example of negative sampling strategy for proactive task fine-tuning.** We apply negative sampling to fine-tune ViSpeak.

Proactive task fine-tuning. For proactive reminder tasks, we fine-tune the model using a negative–positive sampling strategy together with ViSpeak's informative head. As illustrated in Fig. 8, all timestamps before the target object's first fixation are treated as *negative* instances, where the model is trained to output an empty response so that it learns when it should remain silent. At the first fixation timestamp, a *positive* instance is provided and the model is trained to generate the alert message, teaching it the exact moment a notification should be triggered. Each training example is formatted as a short multi-turn sequence in which early turns contain empty responses and the first positive turn contains the alert (e.g., "You are now gazing <screen>."), clearly marking the transition from non-trigger to trigger conditions.

Quantitative results. As shown in Tab. 6, training only on external gaze QA datasets lowers the overall accuracy, indicating that these datasets alone do not transfer effectively to the streaming gaze reasoning setting. In contrast, adding automatically generated StreamGaze-style data leads to clear improvements. EgoExoLearn is particularly helpful because it is in-domain with respect to egocentric manipulation scenes, making its synthesized gaze supervision more aligned with the target distribution. Incorporating Ego4D-Gaze provides additional gains, and combining all data sources yields the best overall accuracy of 0.521. However, because StreamGaze covers a wide range of tasks, the resulting data distribution is inherently imbalanced, so performance does not increase uniformly across all tasks. Overall, these results show that in-domain synthetic gaze supervision is essential for improving ViSpeak, while external datasets alone are insufficient for this setting.

C.2. Details of Gaze Input Prompting

We provide each gaze-prompting instruction in Fig. 9, inspired by [19], to obtain the Qwen2.5-VL performances reported in Tab. 3. For the text gaze prompting setting, we extract the user's fixation center prior to the query time and embed the corresponding fixation coordinates into the textual prompt. To construct a salience-map prompt, we follow Algorithm 1 from Appendix B of [19] without modification, compressing the entire gaze trajectory into a single salience heatmap. We then prepend this salience map to the input video frames so that the MLLM jointly attends to both the visual stream and the gaze-derived salience distribution. Tab. 7 also provide detailed performance per each task on GPT-4o, InternVL-3.5 and ViSpeak.

C.3. Details of Gaze-based Reasoning

We illustrate the input formats for gaze-, text-, and visual-based reasoning in Figs. 19 to 21. We further provide qualitative comparisons between text-only reasoning and gaze- or visual-guided reasoning in Figs. 10 and 11. In Fig. 10, the gaze-based reasoning correctly identifies the attended object

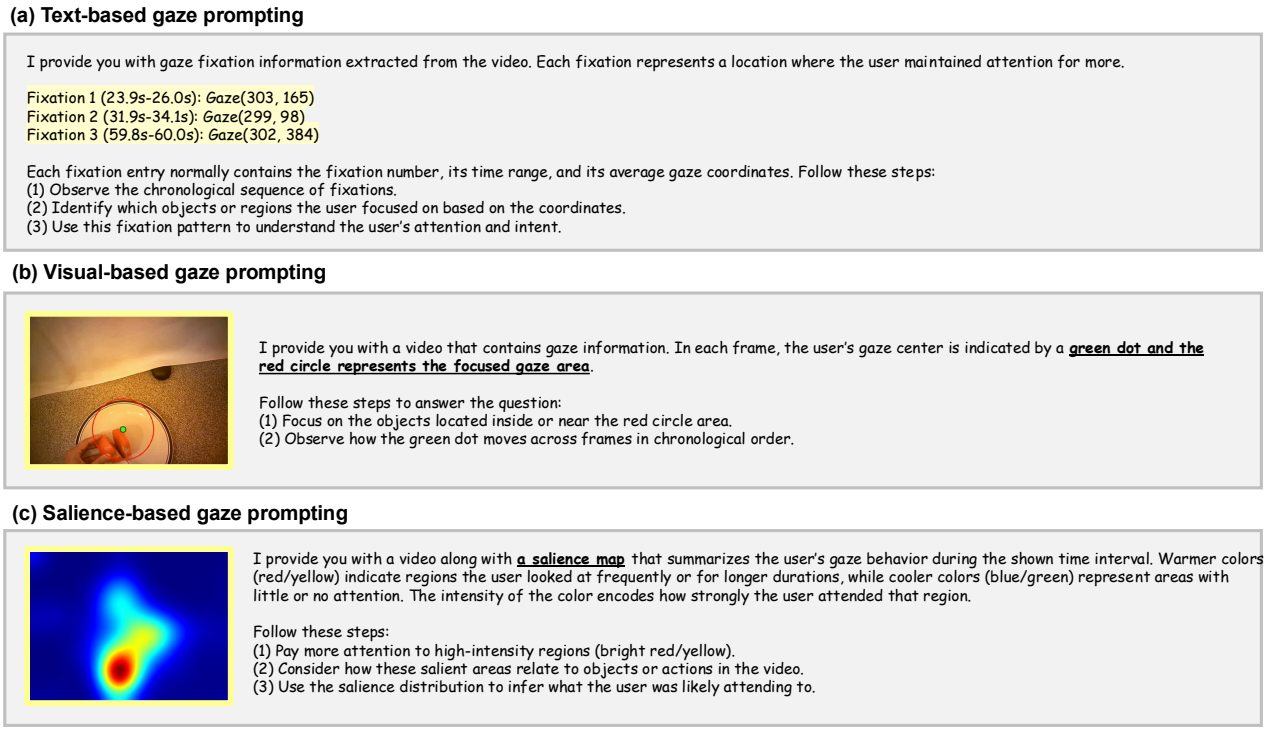


Figure 9. **Gaze prompting strategies for StreamGaze** We adopt three strategies for inputting gaze information to MLLMs.

Table 7. **Detailed performance of gaze input prompting across each StreamGaze task.** We ablate effect of gaze input prompting on GPT-4o [17], InternVL3.5 [26] and ViSpeak [4].

Method	Params	Frames	Past				Present				Proactive		Overall
			NFI	OTP	SR	GSM	OI (E)	OI (H)	OAR	FAP	GTA	OAA	
GPT-4o [17]	-	16	0.601	0.459	0.507	0.607	0.725	0.731	0.594	0.375	0.608	0.148	0.536
+ visual prompt	-	16	0.601	0.449	0.535	0.580	0.729	0.730	0.596	0.370	0.597	0.149	0.535
InternVL3.5 [26]	8B	Adaptive	0.469	0.370	0.526	0.510	0.626	0.637	0.463	0.382	0.376	0.048	0.441
+ visual prompt	8B	Adaptive	0.490	0.311	0.573	0.548	0.627	0.628	0.466	0.372	0.373	0.051	0.444
ViSpeak [4]	7B	1 fps	0.493	0.374	0.417	0.521	0.591	0.560	0.438	0.336	0.625	0.334	0.469
+ visual prompt	7B	1 fps	0.463	0.358	0.417	0.473	0.572	0.581	0.406	0.309	0.635	0.458	0.467

by leveraging the fixation location, whereas the text-only reasoning fails due to relying solely on language priors without visual grounding. In Fig. 11, the visual-based reasoning succeeds by using the detected object region to determine which background item was outside the user's field of view, while the text-only reasoning struggles because it lacks explicit spatial grounding.

D. StreamGaze Data Details

D.1. Data Statistics

Overall statistics. We provide StreamGaze data statistics in Figs. 6 and 7 and Tab. 8. From Fig. 7: (a) The videos cover three domains—cooking, laboratory work, and assembly—with cooking comprising the majority. This distribution reflects realistic egocentric environments while still offering diverse interaction patterns. (b) The word cloud highlights frequently appearing FOV objects such as hands, knives, gloves, bottles, and cutting boards, indicating that the extracted objects largely correspond to action-related tools and manipulable items. (c) Video lengths vary from just a few minutes to more than half an hour, with most clips between 5 and 15 minutes. This range provides both short segments for immediate reasoning and longer sequences suitable for multi-step or extended interactions.

Attribute statistics of OAR task. For the OAR (Object Attribute Recognition) task, we generate attribute-focused questions centered on the gazed object's properties. The distribution covers *color* (65.61%), *material* (19.3%), *state* (5.5%), *texture* (5.3%), and *shape* (4.1%), reflecting the dominant attribute types encountered in egocentric manipulation scenes.

Table 8. **Statistics of StreamGaze tasks.**

Task	# Samples
Gaze Sequence Matching	186
Non-Fixated Object Identification	650
Object Transition Prediction	494
Scene Recall	211
Future Action Prediction	921
Object Attribute Recognition	1,419
Object Identification (Easy)	1,487
Object Identification (Hard)	1,005
Gaze-Triggered Alert	283
Object Appearance Alert	1,865
Total	**8,521**

D.2. QA Examples

We provide data examples per each STREAMGAZE task for better understanding in Figs. 12 to 16. An orange question mark in each task figure denotes a query time.

E. Limitations and Future Work

Although STREAMGAZE offers the first comprehensive benchmark for gaze-guided streaming reasoning, some limitations suggest promising avenues for future work. STREAMGAZE spans ten heterogeneous tasks with inherently imbalanced sample sizes, and future work could explore curriculum or task-balanced training to improve uniformity across tasks. The benchmark is also grounded in manipulation-centric egocentric environments, leaving broader domains such as social interaction, navigation, and outdoor activities largely unexplored.

F. License

We provide licenses of datasets and models used in our work in Tab. 9.

Table 9. **Licenses of datasets and models.**

Resource	License
EgoExoLearn	MIT License (link)
HoloAssist	CDLAv2 License (link)
ViSpeak	Apache-2.0 (link)
InternVL 3.5	Apache-2.0 (link)
Qwen3	Apache-2.0 (link)

[Task: Object Identification]
What is this object currently gazing?

A. sink B. glassware **C. box** D. counter

(a) Text-based reasoning

Thinking: Analyzing the sequence of images, the gaze markers (green dots) consistently point to different objects within red circles. In this case, the gaze marker highlights various objects, but the focus remains on the surface where items like plates, a box, and glassware are placed. This surface is the counter.

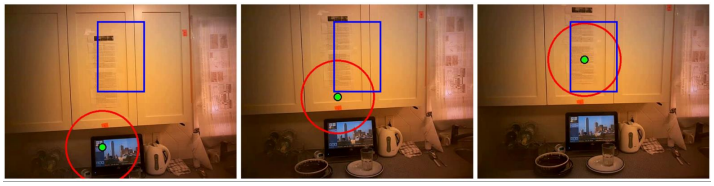
Answer: D. Counter

(b) Gaze-based reasoning

Thinking: Up to the given timestamp, the gaze center is closest to the box located on the counter. The green dot and red circle consistently focus on the box in the last few frames. Based on the visual evidence, the box is the target object.
Predicted Gaze: (427, 308)

Answer: C. box

Figure 10. **Qualitative comparison with text reasoning and gaze-based reasoning.** We visualize examples from the scene reconstruction task comparing (a) text and (b) gaze-based reasoning.



[Task: Scene Reconstruction]
When the user was gazing at the {paper}, which background object was NOT visible?

A. towel B. bowl C. spoon **D. pot**

(a) Text-based reasoning

Thinking: When the gaze is centered on the paper, the towel on the right side of the frame, bowl on the left bottom, spoon near the kettle, and pot on the left are all potential background objects. However, based on the field of view and gaze focus, the towel is out of view as it is located further to the right and not within the central focus area of the gaze.

Answer: A. towel

(b) Visual-based reasoning

Thinking: The gaze is centered on the paper, as indicated by the green dot at (0.5, 0.3) with a radius of 0.2. **The paper is grounded as the object nearest to the gaze center**. To determine which background object was NOT visible when the user was gazing at the paper, I assess the visibility of the listed options: towel, bowl, spoon, and pot. The towel is visible on the right edge, the bowl is visible on the left bottom, the spoon is visible near the kettle, and the pot is visible on the left bottom. Thus, none of these objects are excluded from the view.

Answer: D. Pot

Figure 11. **Qualitative comparison with text reasoning and visual-based reasoning.** We visualize examples from the object identification task comparing (a) text and (b) visual-based reasoning.

Object Transition Prediction

00:10 *00:18*

What object does the user gaze at next after looking at the {phone, table}?
A. Egg mixture B. Laptop C. Bowl. **D. Knob**

Non-Fixated Object Identification

03:08 *03:15*

Among {spatula, pan, pot, pancake}, which did the user never gaze at?
A. Spatula B. Pan **C. Pot** D. Pancake

Figure 12. **StreamGaze data example for OTP and NFI task.**

Scene Recall

00:18

When the user was gazing at the {paper, cabinet}, which background object was NOT visible?
A. Dish B. Cup C. Glass **D. Oil**

Gaze Sequence Matching

31:36 *32:07*

Which transition best matches the user's gaze pattern?
A. {pot, spoon} → {vegetables, sink, faucet} → {pepper, knife, hand}
B. {paper, fork} → {paper, tomato} → {bowl, egg}
C. {bell pepper, bottle} → {bowl, knife} → {cucumber, faucet}
D. {vegetables, sink, faucet} → {pot, spoon} → {pepper, knife, hand}

Figure 13. **StreamGaze data example for SR and GSM task.**

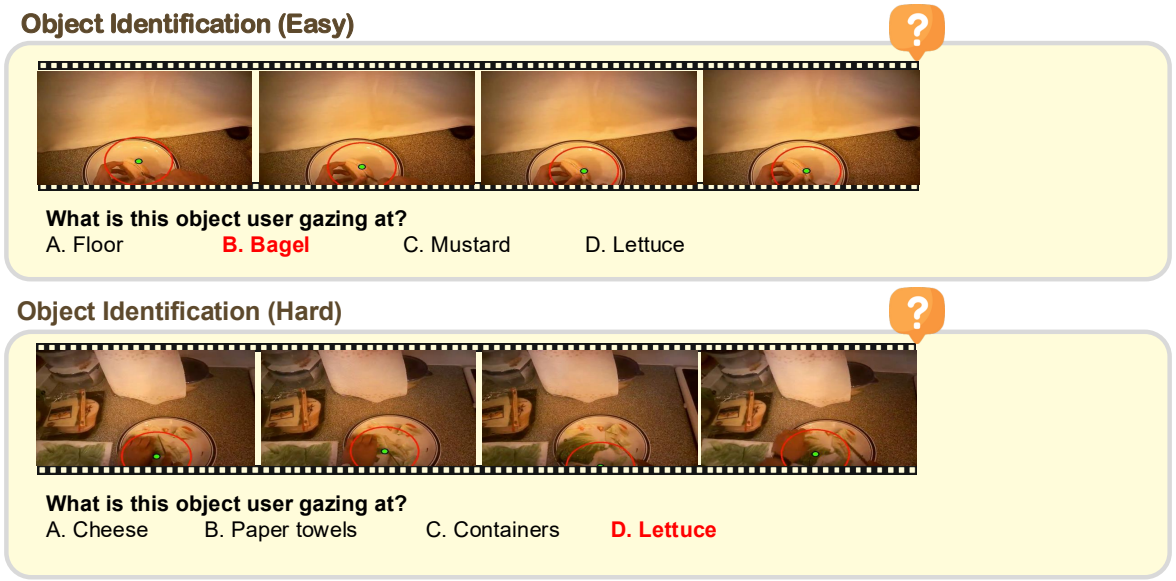


Figure 14. **StreamGaze data example for OI (Easy/Hard) task.**

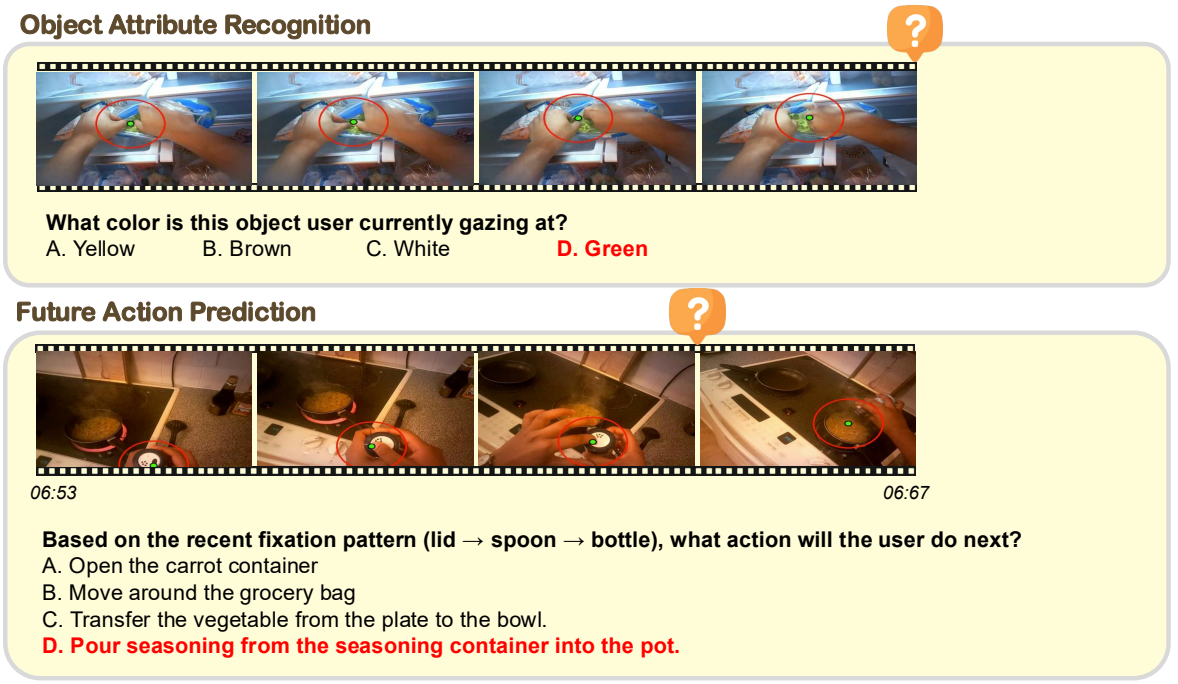


Figure 15. **StreamGaze data example for OAR and FAP task.**

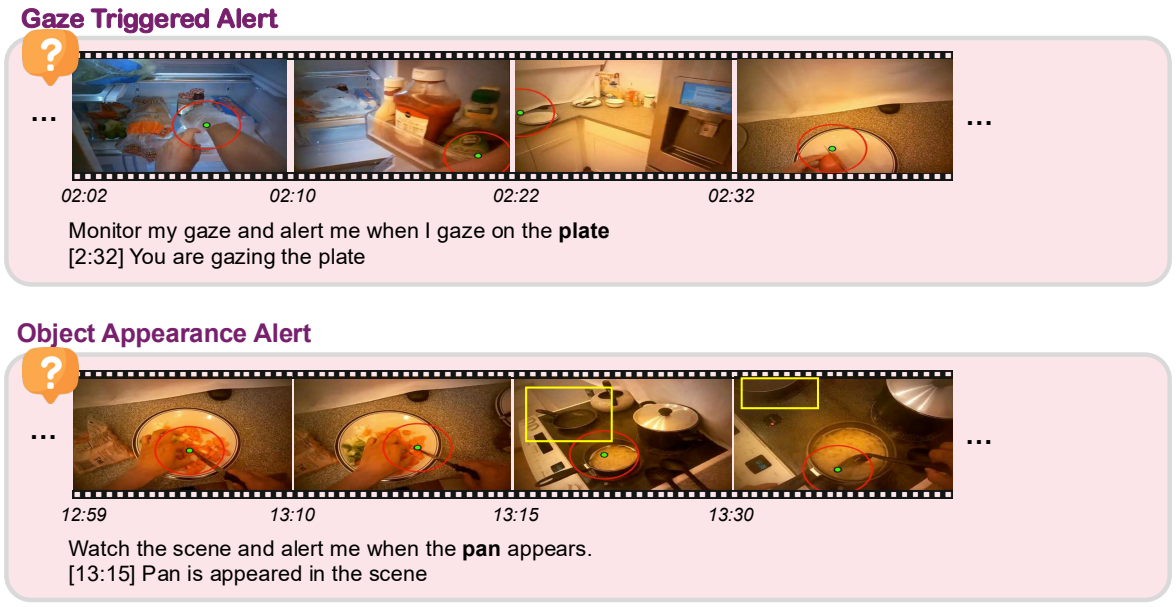


Figure 16. **StreamGaze data example for GTA and OAA task.**

Prompt for FOV region object extraction

```
Context: {action_caption}

Known objects: {object_pool}
IMPORTANT: Reuse these exact names if you see the same objects.
Only use new names for clearly different objects.

Analyze this video clip and return a JSON with this exact structure:

{
  "scene_caption": "Brief 2-3 sentence description of the overall scene
  and what's happening",
  "gaze_object": {
    "object_identity": "name of object at coordinate (x, y)",
    "detailed_caption": "Natural 3-5 sentence description of this object's appearance
    and characteristics"
  },
  "other_objects": [
    {
      "object_identity": "object_name (e.g., 'cup', 'person wearing blue shirt')",
      "detailed_caption": "Two sentences describing the object's appearance,
      attributes, and spatial position relative to other objects"
    }
  ]
}

Person rules: Only include if full/upper body visible (not just hands/arms).
CRITICAL: object_identity must be "person wearing [clothing description]"
NOT just "person".

Return only valid JSON. Be concise and direct.
```

Figure 17. **Prompt for FOV region object extraction.**

Prompt for out-of-FOV region object extraction

```
Context: {action_caption}

Known objects: {object_pool}
IMPORTANT: Reuse these exact names if you see the same objects.
Only use new names for clearly different objects.

Analyze this video clip (with the center region masked) and return a JSON
with this structure:

{
  "other_objects": [
    {
      "object_identity": "object_name (e.g., 'bottle', 'person wearing green jacket')",
      "detailed_caption": "Two sentences describing the object's appearance,
      attributes, and spatial position"
    }
  ]
}

Focus only on objects OUTSIDE the black masked region.

Person rules: Only include if full/upper body visible (not just hands/arms).
CRITICAL: object_identity must be "person wearing [clothing description]"
NOT just "person".

Return only valid JSON.
```

Figure 18. **Prompt for out-of-FOV region object extraction.**

Prompt for text-based reasoning

```
You are an expert at gaze-conditioned streaming video reasoning.

Rules:
1) Only use visual evidence up to the given timestamp.
2) Do your step-by-step reasoning ONLY inside <think>...</think>.
3) Give exactly ONE final choice (A/B/C/D) or yes/no with its short text
inside <answer>...</answer>.

{Question}
```

Figure 19. **Prompt for text-based reasoning.**

Prompt for gaze-based reasoning

```
You are an expert at gaze-conditioned streaming video reasoning.

Rules:
1) Only use visual evidence up to the given timestamp (no future leakage).
2) Do your step-by-step reasoning ONLY inside <think>...</think>.
3) Return exactly three tags in order: <gaze>...</gaze> <think>...</think>
<answer>...</answer>.
4) In <gaze>, estimate the most recent gaze center and FOV radius from the green dot
and red circle in the frames up to the timestamp.
5) If the green dot/Red circle is not visible, output "unknown" in <gaze>
but still reason and answer using other available evidence.
6) In <answer>, give exactly ONE final choice (A/B/C/D) or Yes/No with a short text.

{Question}
```

Figure 20. **Prompt for gaze-based reasoning.**

Prompt for visual-based reasoning

```
You are an expert at gaze-conditioned streaming video reasoning.

Rules:

1) Only use visual evidence up to the given timestamp (no future leakage).
2) Produce your reasoning ONLY inside <think>...</think> and never reveal
reasoning outside the tag.
3) Return exactly FOUR tags in this order: <gaze>...</gaze> <objects>...</objects>
<think>...</think> <answer>...</answer>.
4) In <gaze>, estimate the most recent gaze center and FOV radius from the green dot
and red circle, or output "unknown" if not visible.
5) In <objects>, ground the most relevant objects inside or closest to the
gaze/FOV region; for each object provide: object_name, bbox_x1, bbox_y1, bbox_x2, bbox_y2
normalized to [0,1], or output "none" if no relevant object appears.
6) In <think>, use the gaze and grounded objects to perform step-by-step reasoning
and explain exclusion of irrelevant options.
7) In <answer>, provide exactly ONE final label (A/B/C/D or Yes/No)
with a short justification.

{Question}
```

Figure 21. **Prompt for visual-based reasoning.**

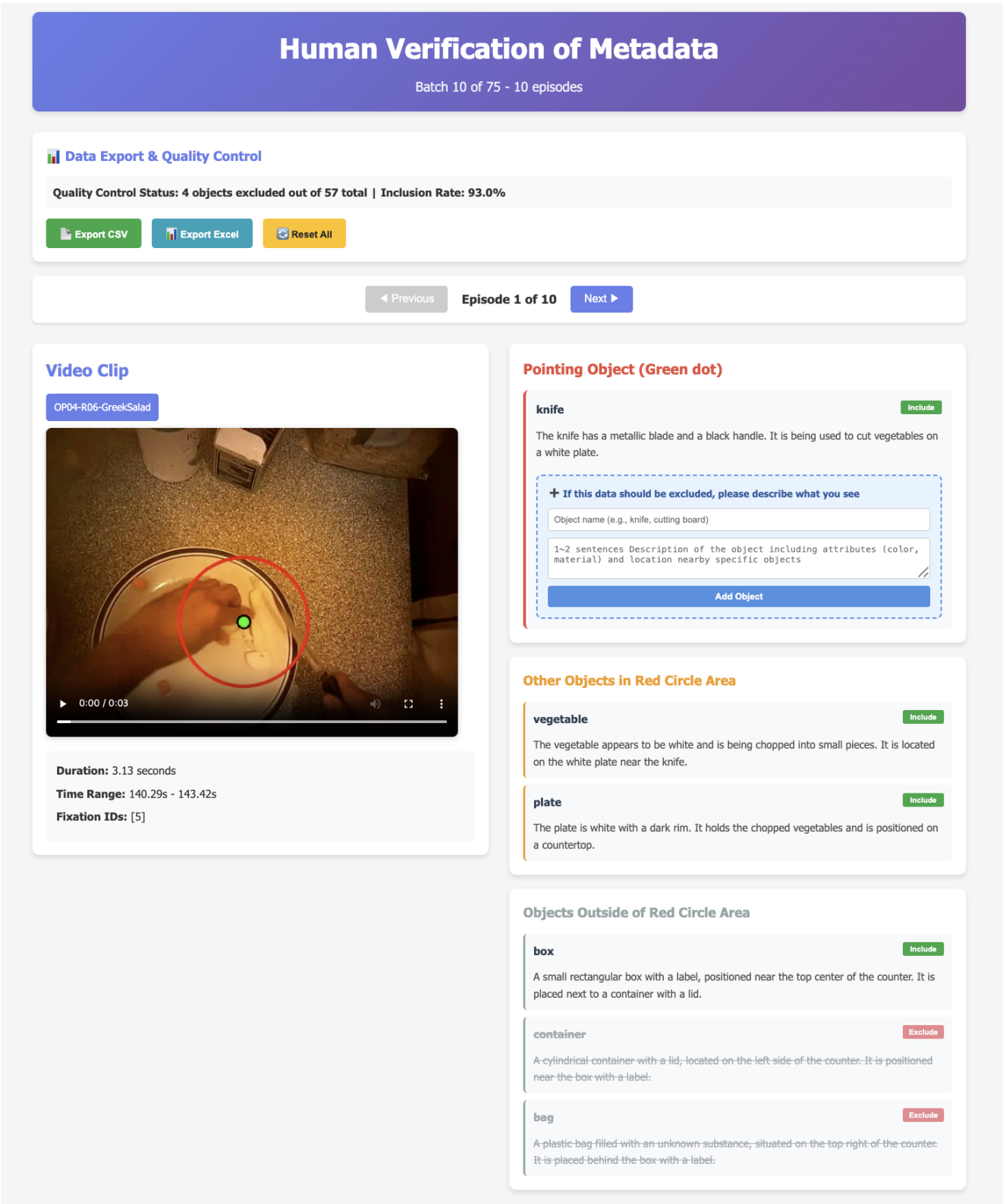


Figure 22. **HTML for human verification of StreamGaze data construction.**

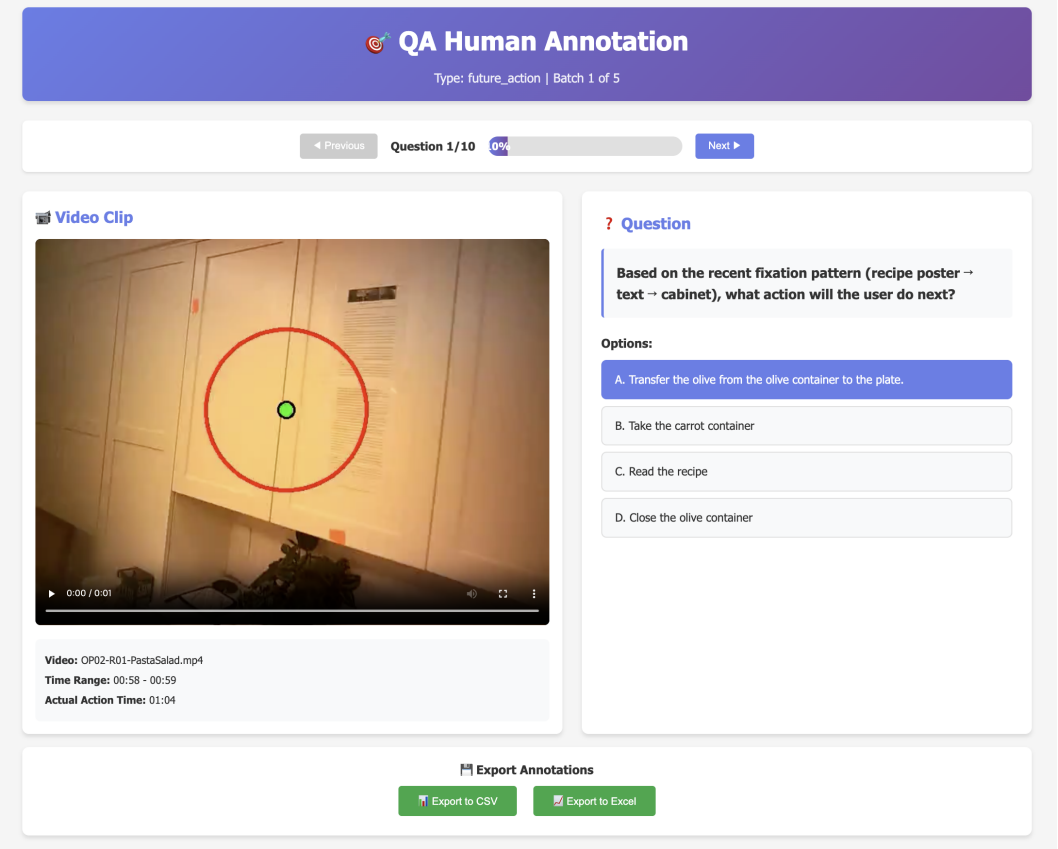
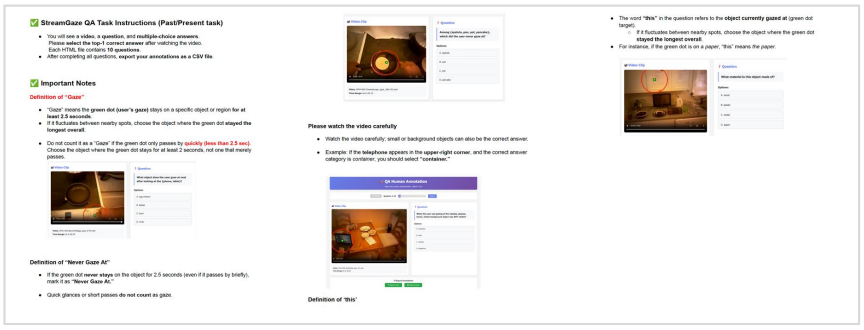


Figure 23. **HTML for human oracle evaluation of StreamGaze.**



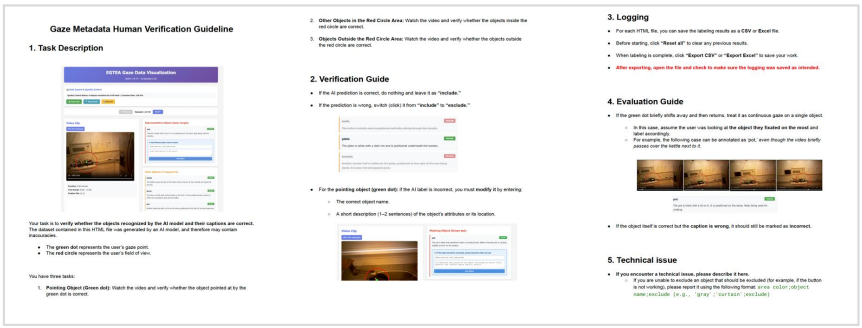


Figure 24. **Instructions provided to human annotators.**