ActiveEye: Enabling Continuous and Responsive Video Understanding for Smart Eyewear Systems

ZHENYU XU* and TIANLIN LU*, School of Computer Science, Fudan University, China
YINGYING ZHAO, Department of Computer and Information Sciences, University of Strathclyde, United Kingdom
YUJIANG WANG, Oxford Suzhou Centre for Advanced Research, China
MINGZHI DONG, Department of Computer Science, University of Bath, United Kingdom
YUHU CHANG, School of Computer Science, Fudan University, China
QIN LV, Department of Computer Science, University of Colorado Boulder, United States
ROBERT P. DICK, Department of Electrical Engineering and Computer Science, University of Michigan, United States
FAN YANG, School of Microelectronics, Fudan University, China
TUN LU, NING GU, and LI SHANG†, School of Computer Science, Fudan University, China

Integrating vision-language models (VLMs) with wearable devices offers great potential for continuous and responsive video understanding, a key capability for applications such as smart eyewear-based conversational assistants. However, achieving this on resource-constrained devices is challenging due to the high energy demands of continuous spatial-temporal sampling and transmission. We propose *ActiveEye*, a VLM designed for energy-efficient and responsive video understanding. *ActiveEye* separates visual and motion semantic representations and incorporates an active perception-based feedback path to adaptively adjust spatial-temporal sampling and transmission rates. Implemented as a wearable-mobile-cloud system, *ActiveEye* is evaluated for energy efficiency, real-time semantic change detection, and video understanding in both laboratory and field studies. Using the EgoSchema dataset, *ActiveEye* reduces the front-end energy consumption by 49.14%, supporting 8.37 hours of continuous operation on a 2.1 Wh battery. It achieves the highest F1 score (0.80) and the lowest average time difference (1.30 s) compared with heuristic-based event detection algorithms, validating its timely semantic detection. Furthermore, *ActiveEye* achieves a visual question answering (VQA) accuracy of 61.6%, which is comparable to state-of-the-art VLM agents, despite their reliance on larger language decoders and more computationally intensive frame selection strategies. Two rounds

*Equal contribution.
†Corresponding author.

Authors' Contact Information: Zhenyu Xu, zyxu22@m.fudan.edu.cn; Tianlin Lu, 23210240250@m.fudan.edu.cn, School of Computer Science, Fudan University, Shanghai, China; Yingying Zhao, yingying.zhao@strath.ac.uk, Department of Computer and Information Sciences, University of Strathclyde, Glasgow, United Kingdom; Yujiang Wang, yujiang.wang@oscar.ox.ac.uk, Oxford Suzhou Centre for Advanced Research, Suzhou, China; Mingzhi Dong, mingzhidong@gmail.com, Department of Computer Science, University of Bath, Bath, United Kingdom; Yuhu Chang, yhchang@fudan.edu.cn, School of Computer Science, Fudan University, Shanghai, China; Qin Lv, qin.lv@colorado.edu, Department of Computer Science, University of Colorado Boulder, Boulder, Colorado, United States; Robert P. Dick, dickrp@umich.edu, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan, United States; Fan Yang, yangfan@fudan.edu.cn, School of Microelectronics, Fudan University, Shanghai, China; Tun Lu, lutun@fudan.edu.cn; Ning Gu, ninggu@fudan.edu.cn; Li Shang, lishang@fudan.edu.cn, School of Computer Science, Fudan University, Shanghai, China.

of in-field user evaluations further confirm its effectiveness in real-world settings, demonstrating its practical viability as a continuous and responsive video understanding system, conversational assistant, and wearable companion.

CCS Concepts: • **Human-centered computing → Mobile devices**.

Additional Key Words and Phrases: Smart Eyewear, Video Understanding, Energy-efficient, Responsive

ACM Reference Format:
Zhenyu Xu, Tianlin Lu, Yingying Zhao, Yujiang Wang, Mingzhi Dong, Yuhu Chang, Qin Lv, Robert P. Dick, Fan Yang, Tun Lu, Ning Gu, and Li Shang. 2025. ActiveEye: Enabling Continuous and Responsive Video Understanding for Smart Eyewear Systems. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 4, Article 228 (December 2025), 33 pages. https://doi.org/10.1145/3770641

1 Introduction

The integration of vision-language models (VLMs) with wearable devices can significantly advance the capabilities of wearable technologies, improving user experience in numerous human-computer interaction (HCI) scenarios [23, 29, 40, 74]. A notable example is smart eyewear equipped with a scene camera that captures user's field of view, enabling visual context awareness. Acting as the perceptual front end, the camera records the user's perspective, while VLMs function as the processing back end, analyzing visual input to interpret scenes and understand the surrounding environment. This capability allows the system not only to recognize visual context but also to support context-aware interactions. Such advancements can support applications such as personalized assistance and companionship, expanding the potential of smart eyewear in everyday scenarios [14, 37, 65, 74].

For smart eyewear to function effectively, it must maintain continuous and responsive visual understanding. This enables the device to perceive the user's visual surroundings in real time, track changes in the visual context, and respond promptly when needed. For instance, in a conversational assistant setting, the system could engage with the user about a movie scene as it unfolds or acknowledge a successful basketball shot when it occurs. By offering such interactions in a timely fashion, the smart eyewear system evolves from a passive tool into an active companion, creating a more immersive and interactive user experience.

However, **the key challenge is balancing the device's limited battery capacity with the high demands of video data sampling and transmission required for VLM-based video understanding**, particularly in dynamic scenarios. To illustrate this issue, consider user activities such as watching sports or shopping, where capturing critical moments (e.g., scoring a goal or picking up an item requires high spatial and/or temporal scene-sensing resolution. Existing smart eyewear products and research efforts, such as Ray-Ban Meta Smart Glasses[1] and OS-1 [74], address power constraints by capturing images only during user interactions or at long intervals. Although energy-efficient, these approaches limit contextual awareness by omitting prior scene information or introducing delays in perception, which restricts their usability in the real world. For example, the Ray-Ban Meta Smart Glasses have a limited runtime due to thermal conditions, and the company officially estimates a 30-minute run time, which also depends on the ambient temperature being at least 5 degrees Celsius[2].

We argue that the challenge above stems from a fundamental contradiction between the data processing methods employed by existing VLMs and the nature of visual semantics. Existing VLMs process the visual and motion representations simultaneously to preserve the video semantics, requiring the system to capture and transmit uniformly high spatial-temporal resolution video data for processing. However, the inherent characteristics of visual and motion information necessitate variant resolutions. The visual world we navigate is both rich in detailed textures and dynamic in motion [4, 33], yet these two aspects impose different spatial-temporal sampling requirements for effective video understanding [24]. Visual semantics, such as object categories and scene context, rely on high spatial resolution to capture fine-grained details such as colors and

[1] https://www.meta.com/ai-glasses/
[2] https://moorinsightsstrategy.com/research-notes/ray-ban-meta-smart-glasses-review-better-cooler-and-more-useful-than-ever

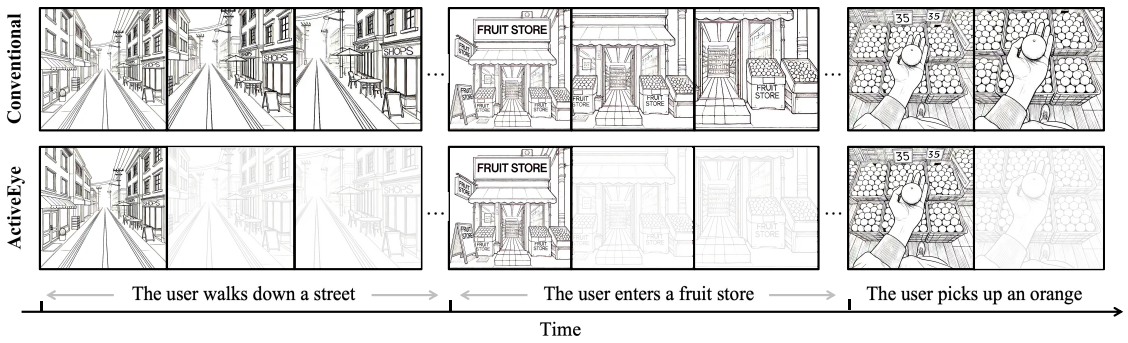

Fig. 1. Comparison of *ActiveEye*'s operation with conventional VLM-based smart eyewear systems.

textures. However, these semantics typically remain stable over time, allowing for lower temporal sampling rates. In contrast, motion semantics, which describes changes in object positions, requires higher temporal sampling rates to capture rapid transitions but can be extracted from lower spatial resolutions, as motion patterns tend to be spatially coherent across neighboring pixels [21]. **The coupling of visual and motion representations forces the video perception front end to meet both demands, leading to uniformly high spatial-temporal sampling rates for sensing and transmission**. This hinders the achievement of an effective balance between video understanding and energy efficiency for smart eyewear systems, leading to excessive energy consumption and poor on-device performance.

To this end, we propose *ActiveEye*, a vision-language model (VLM) designed for energy-efficient, continuous, and accurate video understanding in wearable systems. The core of *ActiveEye* lies in its decoupling of visual and motion semantics, allowing each to be processed at different spatial and temporal resolutions. This design enables adaptive resource usage and significantly reduces energy consumption. To preserve accuracy while conserving power, *ActiveEye* is grounded in active perception theory [3, 7, 8], which suggests that intelligent agents should acquire sensory data only when it deviates from prior knowledge or expectations. Accordingly, *ActiveEye* continuously monitors the decoupled visual and motion inputs to refine its understanding of the environment. A feedback loop further guides a Mixture-of-Experts (MoE) model to predict motion dynamics and determine whether the system should update its interpretation of the current visual scene. Based on this prediction, the system either enters a high-sampling-rate, energy-intensive mode or continues operating in a low-power state when changes are minimal. As shown in Figure 1, *ActiveEye* alternates between two stages—activating visual updates only when semantic changes are detected. This approach effectively filters out redundant visual information, reducing unnecessary sensing and computation while maintaining continuous and accurate video understanding.

We implement *ActiveEye* as a wearable-mobile-cloud system designed for real-time conversational assistant and companion applications using smart eyewear. In an in-lab study, we evaluate its energy efficiency, semantic change detection accuracy, and video understanding capability using the EgoSchema dataset [52]. Results demonstrate that *ActiveEye* significantly reduces front-end energy consumption, supports up to 8.37 hours daily usage, and outperforms heuristic baselines in detecting activity transitions. It also achieves video understanding accuracy comparable with recent state-of-the-art VLM agents, despite using smaller language decoders and less computationally intensive frame selection strategies. To further evaluate *ActiveEye* in real-world settings, we conduct a 3-month in-field user study with 25 participants. The user study is carried out in two rounds, each with distinct objectives. The first round focuses on system deployment validation and real-time performance testing, while the second examines user experience and system usability. The study produces several key insights, and

all quantitative results and findings from both in-lab and in-field evaluations are summarized in the following contributions.

In summary, this work makes the following contributions:

(1) We propose *ActiveEye*, a vision-language model (VLM) designed for continuous and responsive video understanding for smart eyewear systems. It decouples visual and motion representations and incorporates a feedback-driven adaptive spatial-temporal sampling mechanism to optimize energy efficiency while maintaining semantic awareness.
(2) We implement *ActiveEye* within a wearable-mobile-cloud system and evaluate its performance using the EgoSchema dataset. Results demonstrate that *ActiveEye* reduces front-end energy consumption by 49.14%, enabling 8.37 hours of continuous operation on a 2.1 Wh battery—supporting a once-daily charging requirement. Despite its efficiency, it maintains timely semantic detection, achieving the highest F1 score (0.80) and the lowest average detection delay (1.30 s) among baseline methods.
(3) *ActiveEye* achieves VQA accuracy comparable to state-of-the-art baselines while significantly reducing power consumption and processing latency.
(4) Pilot studies demonstrate the real-world usability of *ActiveEye*, achieving 90.70% relevance in captured scene content and generating contextually appropriate captions across two downstream applications: conversational assistant and companion. Users reported positive user experience (UX) and good usability, with easier adoption by participants with computer science (CS) backgrounds.

2 Related Work

2.1 Vision Language Models

With the rapid advancement of LLMs, many studies have focused on integrating visual encoders into these models, giving rise to VLMs, also referred to as Multimodal Large Language Models (MLLMs) [78]. A typical VLM uses a vision encoder (e.g., CLIP [58], SigLIP [81]) to encode images into patch tokens, which are aligned with the LLM via a connector such as an MLP [17, 48, 67], cross-attention module [2, 6, 76], or Q-Former [20, 44, 86, 87], enabling tasks like VQA and image captioning.

Building upon single-image processing, researchers extend VLMs to video understanding. A common approach involves uniformly sampling frames from videos, encoding them with a vision encoder, and feeding the representations into the LLM [5, 16, 43]. Although effective, this approach significantly increases memory usage and computation cost, limiting its applicability to long-form video analysis and real-time video understanding. Various techniques have been proposed to mitigate these issues, including using LLM agents to select semantically informative keyframes [22, 54, 68], applying token pooling [51, 72, 73, 75], merging [35, 60], or pruning [15, 77, 79, 83] to reduce the sequence length of patch tokens, and designing lightweight LLM architectures to lower computational demands [18, 31, 46, 80, 88]. However, existing efforts primarily reduce back-end costs while overlooking the front-end burden of video perception, which typically occurs directly on energy-constrained devices such as smart eyewear. This work addresses this challenge by significantly reducing front-end visual data perception through a dynamically adjusted spatial–temporal sampling and transmission method, while still enabling continuous and comprehensive video understanding.

2.2 Smart Eyewear with Visual Understanding Capabilities

Smart eyewear, equipped with first-person vision, has emerged as a promising platform for integrating computer vision techniques such as object detection and VLMs to interpret the wearer's environment and activities. This capability enables immersive and context-aware user experiences [82]. For instance, VisionARy [40] enhances language learning by detecting objects in the environment and incorporating them into LLM-generated dialogues. MemX [12] combines eye tracking with visual scene understanding to track human visual attention, enabling

applications like life-logging and travel experience recording. EMOShip [84] combines eye tracking with VLMs for accurate emotion recognition and semantic analysis of emotional triggers, supporting applications such as emotional self-reflection and life-logging. GazeGPT [37] integrates eye tracking with GPT-4V to generate personalized and contextually relevant responses based on the user's visual attention. OS-1 [74] explores the integration of smart eyewear with LLMs, VLMs, and vector databases, enabling long-term memory and continuous user adaptation. In this work, we focus on continuous and responsive video understanding on smart eyewear, with the goal of supporting applications such as conversational assistant and companion.

2.3 Decomposing Vision and Motion Representations

Two-stream networks [25] were first introduced to separately extract visual and motion representations from single-frame images and multi-frame optical flow for action recognition. Later, SlowFast networks [24] introduced a dual-branch design, where a heavy-weight encoder extracts visual semantics from low frame rate sequences, while a lightweight encoder captures motion semantics from high frame rate sequences. With the emergence of vision encoders trained via contrastive learning on large-scale image-text pairs, researchers integrated optical flow-based motion representations with visual features for zero-shot action recognition [57]. More recently, as VLMs advance, studies extend the concept of decomposing visual and motion representations to video-language tasks. Video-LaVIT [34], for instance, extracts keyframes and motion vectors from MPEG-encoded videos, encoding them separately for video understanding and generation. SlowFast-LLaVA [73] applies different temporal sampling and spatial pooling to visual features to separate visual and motion semantics. Unlike prior work, our approach employs adaptive spatial-temporal sampling to reduce data consumption and transmission, while still maintaining accurate and continuous video understanding.

2.4 Energy-efficient Computer Vision

Prior works have explored adaptive spatial-temporal resolution and data transmission to reduce energy consumption in vision perception while preserving performance on various video analysis tasks. A common approach involves extracting object locations from low-resolution images and selectively capturing high-resolution images for semantic details. SensEye [38] uses low-power cameras for motion detection, triggering high-resolution cameras only when necessary for resource-intensive tasks such as object recognition and tracking. Digital Foveation [50] designs a multi-round sensing strategy that transmits and processes only the scene regions relevant to the final task. Wang *et al.* [69] further introduced a feedback mechanism, where recognition results guide detection to improve accuracy and efficiency.

Other approaches optimize hardware and algorithms to reduce sensing and transmission costs. LiKamWa *et al.* [45] propose achieving mobile computer vision tasks more energy-efficiently through energy-proportional mechanisms, such as optimizing the clock frequency of sensors and leveraging low-power standby modes between frames. Lubana *et al.* [49] dynamically determine pixel importance based on task requirements and allocate bit depths accordingly to reduce data volume and communication latency. Zhao *et al.* [85] developed a reinforcement learning approach that adaptively selects video frame resolutions while applying temporal feature fusion to mitigate performance loss, reducing energy consumption in video instance segmentation. These approaches inspire our method, which extracts visual and motion semantics from video frames sampled at varying spatial–temporal resolutions, guided by a feedback mechanism for data acquisition. Unlike prior work, we focus on general video semantic understanding rather than specific computer vision tasks, enabling a broader range of interactive applications.

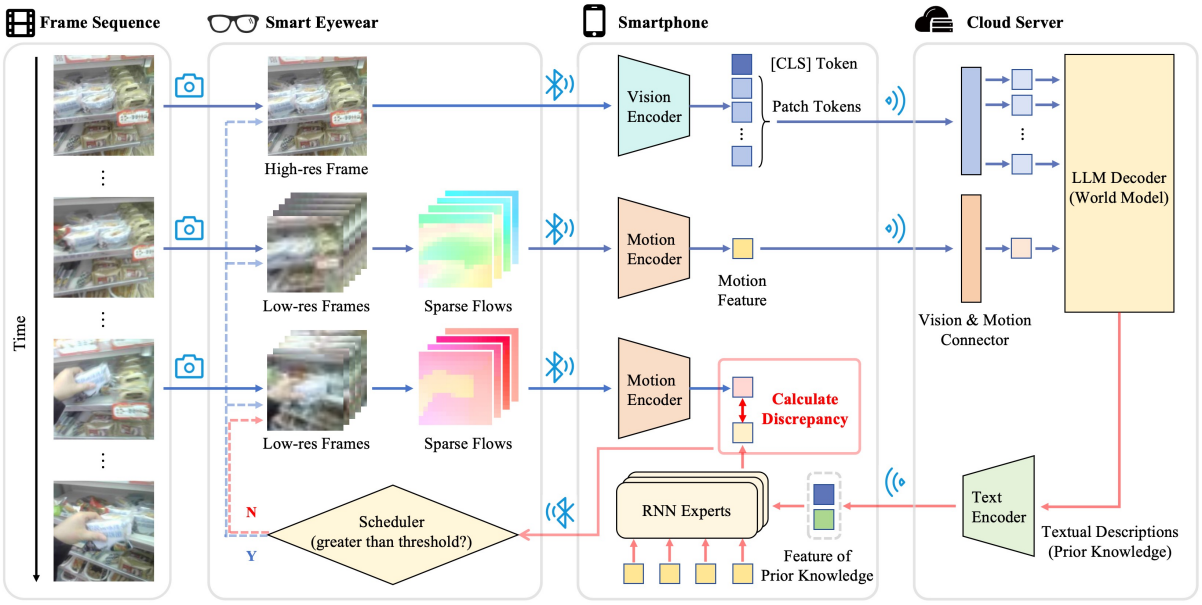

Fig. 2. Workflow of *ActiveEye* and its deployment as an integrated wearable-mobile-cloud system. Blue arrows denote the feed-forward path, while red arrows indicate the feedback path.

3 System Design

This section begins by describing the workflow of *ActiveEye*, including its feed-forward and feedback paths. It then details the algorithmic implementation, hardware design, and the integration of the system across wearable, mobile, and cloud platforms. Given that the primary focus of *ActiveEye* is the energy efficiency of the video perception front end, this section concludes with a description of its energy model and power measurement methodology.

3.1 *ActiveEye* Network

The workflow of *ActiveEye* and its deployment as an integrated wearable-mobile-cloud system are illustrated in Figure 2. *ActiveEye* operates in two alternating stages: *perception-to-understanding* and *understanding-to-perception*, corresponding to its feed-forward path (denoted by blue arrows) and feedback path (denoted by red arrows), respectively. In the *perception-to-understanding* stage, the feed-forward path captures one high-resolution frame and several low-resolution frames, from which it extracts visual features and motion features, respectively. These features are processed by the LLM decoder to update the system's prior knowledge about the scene, and the recent motion features are retained for future comparisons. In the *understanding-to-perception* stage, the feedback path predicts the next motion features based on prior knowledge and compares them with those extracted from recently captured low-resolution frames. If the discrepancy exceeds a predefined threshold, the system reactivates the feed-forward path to update the prior knowledge. Otherwise, it continues capturing low-resolution frames, extracting motion features, and performing prediction–comparison cycles until the next update is triggered.

3.1.1 Feed-forward Path. The feed-forward path workflow consists of three components: visual feature extraction, motion feature extraction, and prior knowledge generation. It takes as input one high-resolution frame and a

sequence of low-resolution frames. The first five low-resolution frames are used together with the high-resolution frame to update the system's prior knowledge—textual descriptions of the current scene, ongoing activities, and anticipated future events. The remaining low-resolution frames are processed to retain recent motion information for future comparison.

(1) Visual Feature Extraction. The vision encoder is a vision transformer (ViT) model pre-trained on a large set of image-text pairs [81]. It encodes the high-resolution input video frame at time t into semantic tokens, as follows:

$$[v_t^{[CLS]}, v_t^{patch}] = f_{\text{vision}}(x_t^h) \tag{1}$$

Here, $x_t^h \in \mathbb{R}^{H\times W\times 3}$ denotes the high-resolution input video frame, $v_t^{[CLS]} \in \mathbb{R}^C$ represents the global semantic information of x_t, and $v_t^{patch} \in \mathbb{R}^{h\times w\times C}$ encodes the fine-grained semantic information of individual image patches, where $h = H/P$, $w = W/P$, and P is the patch size.

(2) Motion Feature Extraction. Motion features are extracted from sparse optical flow computed on low-resolution video frames, as optical flow has been shown to effectively capture motion semantics in prior action recognition tasks [25]. To ensure real-time computation on resource-constrained edge devices, we employ the pyramid Lucas-Kanade (LK) method [10] to calculate sparse optical flow for $h' \times w'$ uniformly sampled points in each frame:

$$fl_t = f_{\text{LK}}(x_t^l, x_{t+1}^l) \tag{2}$$

where $x_t^l, x_{t+1}^l \in \mathbb{R}^{H'\times W'\times 3}$ are the low-resolution input images, and $fl_t \in \mathbb{R}^{h'\times w'\times 2}$ represents the sparse optical flow between these two frames. Following this step, we extract motion features from four consecutive flow maps, corresponding to a 1 second window under a 4 fps setting. This design is motivated by prior studies showing that short temporal windows (around 1 second) are sufficient to capture semantically meaningful and discriminative motion patterns [11, 63]. These flow maps are concatenated along the channel dimension and then passed through a lightweight CNN encoder to extract motion features:

$$m_t = f_{\text{motion}}(fl_t, fl_{t+1}, fl_{t+2}, fl_{t+3}) \tag{3}$$

where $m_t \in \mathbb{R}^{C'}$ encodes the motion semantics represented by the sparse optical flow.

(3) Prior Knowledge Generation. After extracting visual and motion semantic features, the visual patch tokens v_t^{patch} and motion features m_t, which represent fine-grained visual semantics and coarse-grained motion semantics respectively, are independently passed through a two-layer MLP, denoted as $f_{connector}^v$ and $f_{connector}^m$, to align their dimensions with those of the LLM decoder. The aligned representations, along with a text prompt T^{prompt}, are then used to generate textual descriptions of the current scene, the ongoing events, and predictions of future events. These outputs, representing *ActiveEye*'s prior knowledge of the current scene, are denoted as T_t^{prior} and are formulated as follows:

$$T_t^{\text{prior}} = f_{\text{LLM}}\left(f_{\text{connector}}^v(v_t^{\text{patch}}),\ f_{\text{connector}}^m(m_t),\ T^{\text{prompt}}\right) \tag{4}$$

In this formulation, $f_{connector}^v$ and $f_{connector}^m$ map the visual and motion features into the representation space of the LLM decoder. During the subsequent activation of the feedback path, T_t^{prior} guides the data acquisition process until the feed-forward path is reactivated.

3.1.2 Feedback Path. The feedback path operates in a top-down manner, translating *ActiveEye*'s prior knowledge into predictions for future motion features and controlling the perception process. It employs a mixture of recurrent neural network (RNN) experts, gated by $v_t^{[CLS]}$ and T_t^{prior}, to enable each expert to specialize in modeling specific scene dynamics. This mechanism allows the system to detect semantic changes in the video by predicting future motion features. When the prediction error exceeds a predefined threshold, the feed-forward path is reactivated to update the prior knowledge.

(1) Motion Feature Prediction. We employ a mixture of RNN experts to predict future motion features based on those extracted from observed low-resolution frames. The [CLS] token $v_{t'}^{[CLS]}$ and the prior knowledge $T_{t'}^{prior}$, representing the semantic information of the visual scene and events, are used as inputs to the gating network, where t' denotes the time when the prior knowledge was last updated. Specifically, $T_{t'}^{prior}$ is passed through a text encoder to compress it into a dense vector, which is then concatenated with $v_{t'}^{[CLS]}$ and fed into the gating network to compute scores for each RNN expert:

$$s_i = f_{\text{gating}}\left(\text{concat}\left(f_{\text{text}}(T_{t'}^{\text{prior}}),\ v_{t'}^{[\text{CLS}]}\right)\right), \quad i = 1, \ldots, N \tag{5}$$

Here, N denotes the number of RNN experts, and s_i is the score assigned to the i-th expert. The expert with the highest score is selected to predict the next motion feature:

$$\hat{m}_{t+1} = f_{RNN,i^*}(m_{t'}, m_{t'+1}, \ldots, m_t) \tag{6}$$

where $i^* = \arg\max_i s_i$, and $\hat{m}_{t+1}$ represents the predicted motion feature for the next step.

By combining $T_{t'}^{prior}$ with $v_{t'}^{[CLS]}$, the gating network enables each RNN expert to specialize in modeling motion features for specific scenes and events. This design ensures that a large prediction error reflects significant semantic changes in the scene or motion dynamics. Furthermore, this architecture activates only one RNN expert at any given time, enabling efficient operation while maintaining robust prediction performance.

(2) Discrepancy Detection. Inspired by previous work on event segmentation [1], we compute the prediction error using L1 loss and preprocess it with a low-pass filter to remove noise caused by camera motion. The prediction error is calculated as:

$$E_p(t) = \|m_{t+1} - \hat{m}_{t+1}\|_1 \tag{7}$$

where $\hat{m}_{t+1}$ is the predicted motion feature and m_{t+1} is the observed motion feature. To reduce noise, a low-pass filter is applied to smooth the prediction error over the last n steps, resulting in the perceptual quality metric $P_q(t)$:

$$P_q(t) = P_q(t-1) + \frac{1}{n}(E_p(t) - P_q(t-1)) \tag{8}$$

When the prediction error exceeds the perceptual quality metric $P_q(t-1)$ by a certain threshold ψ_e, the feed-forward path is reactivated to capture one high-resolution frame and several low-resolution frames, thereby triggering an update of the prior knowledge. Otherwise, the feedback path continues by sampling five additional low-resolution frames, extracting their motion features, and comparing them with the predicted ones. The gating decision is formulated as:

$$G(t) = \begin{cases} 1, & \frac{E_p(t)}{P_q(t-1)} > \psi_e \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Here, $G(t) = 1$ indicates that the feed-forward path is reactivated.

3.2 Algorithmic Implementation

We adopt LLaVA-OV-7B [43] as the world model to generate the system's prior knowledge about the scene by using the decoupled motion and visual information and converting them into textual descriptions. We select LLaVA-OV-7B because it is relatively lightweight, achieves competitive performance across a wide range of video understanding tasks, and, most importantly, is open-source. This allows us to access internal weights and perform gradient-based training, which is essential for optimizing our motion connector. These components enable the decoder of LLaVA-OV-7B to interpret structured motion features and align them with the decoupled visual representations. For the feedback path, we employ ResNet-18 [30] as the motion feature extractor f_{motion}, expanding its first convolutional layer to 8 channels to accommodate the multi-frame optical flow input, following common practices in action recognition for extracting motion features from optical flow sequences [25, 66]. This lightweight architecture was chosen to ensure efficient inference on our smartphone. The number of RNN experts, N, is set to 8, with a hidden size of 512. The gating network is a single-layer feed-forward network (FFN), and the text encoder f_{text} is derived from SigLIP [81]. The resolution of high-resolution video frames is set to 384×384, matching the standard input size for the vision encoder f_{vision}, while low-resolution video frames are set to 128×128. Sparse optical flow is estimated by sampling one point for every 8×8 grid, resulting in a sparse optical flow size of 16×16. For prediction error gating, we set the hyperparameters $n = 2$ and $\psi_e = 1.2$. We discuss the rationale for key hyperparameter settings in Section 4.

We train *ActiveEye* on the EgoClip dataset [47], an egocentric video-text dataset containing 129 daily-life scenarios. Videos shorter than 5 seconds are filtered out, leaving 19K video-text pairs for training. The training process runs on 8 NVIDIA GeForce RTX 3090 GPUs, and it consists of three stages:

(1) In the first stage, we train f_{motion} independently using the InfoNCE loss [56] to ensure it can effectively capture motion semantics from optical flow sequences.
(2) In the second stage, we train the feed-forward path. During this phase, only the parameters of $f^{m}_{connector}$ are unfrozen to align the motion features with the LLM's representation space, facilitating the generation of textual descriptions of egocentric videos. To preserve general visual understanding, we also include 12K video-text pairs from the LLaVA-OV-7B pretraining corpus. We use cross-entropy loss for this stage.
(3) In the third stage, we train the feedback path by unfreezing the parameters of f_{gating} and f_{RNN} to learn how to utilize prior knowledge to predict future motion semantics. We employ F1 loss during this stage.

To ensure seamless integration, we develop a smartphone app to manage communication between the smart eyewear and the cloud server. The app incorporates speech recognition [59] and text-to-speech (TTS) [19] capabilities, enabling real-time conversational interactions with the user based on the visual scene. To generate dialogue responses based on the visual scene, we select GPT-4o mini, using the most recent prior knowledge generated by *ActiveEye* before the user initiates the conversation as the context.

3.3 Hardware Design

1) Smart Eyewear Prototype. The wearable component, shown in Figure 3, consists of a 3D-printed frame housing an ESP32-CAM board and a 2.1 Wh lithium battery, connected via an XH 2.54 connector. The ESP32-CAM board is based on the ESP32-WROVER-E module (dual-core 240 MHz Xtensa LX6, 8 MB PSRAM), chosen for low power consumption and ability to perform on-device streamlined optical flow estimation within tight energy constraints. Also, the ESP32 supports Bluetooth 4.2; we leverage its Bluetooth Low Energy (BLE) functionality to transmit the sparse motion features and JPEG-compressed images to the mobile device. This avoids raw video streaming, which would incur higher power consumption and bandwidth costs.

An OV2640 camera, connected to the ESP32-CAM board via an FPC ribbon cable and mounted at the front of the frame, provides adaptive visual sensing as scheduled by the *ActiveEye* system. Specifically:

- Low-resolution mode (128×128 pixels at 4 fps) is used continuously for lightweight motion tracking with minimal energy consumption.
- High-resolution captures (640×480 pixels) are intermittently triggered by system-level scheduling to provide detailed scene understanding when needed.

2) Mobile and Cloud Integration. We use an iPhone 16 Pro as the intermediate mobile device, chosen for its hardware-accelerated machine learning cores (supporting `.mlmodel`) and robust BLE support. On the mobile device, we deploy:

- Vision encoder f_{vision}, a transformer-based module converting high-resolution frames into patch tokens v_t^{patch}. Running this module on the phone (instead of the eyewear) minimizes energy and thermal load on the wearable.
- Motion encoder f_{motion}, a lightweight CNN processing the optical flows received from the eyewear into compact motion features m_t suitable for subsequent gating and RNN processing.
- Gating network f_{gating} and RNN experts f_{RNN}, which together decide whether to invoke the vision encoder and generate predictions.

Importantly, f_{vision} is only activated when the prediction error of motion features exceeds a threshold ψ_e, and remains inactive for most of the time during system operation. This selective activation strategy reduces unnecessary computation and conserves energy.

After on-device processing, the processed tokens and motion features m_t (not raw images) are uploaded to the cloud, which returns a scene context vector T_t^{prior}, summarizing the visual environment and predicted future events. This design reduces both bandwidth usage and privacy risk, as no raw imagery is uploaded.

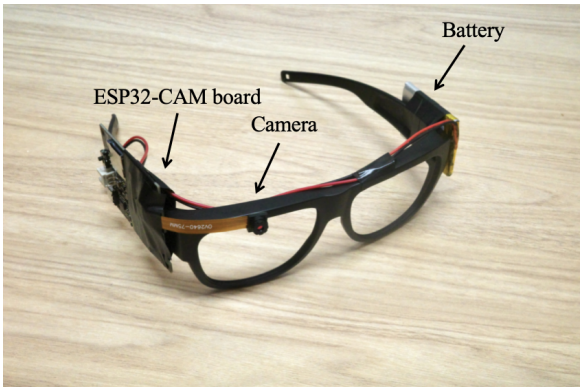

Fig. 3. Prototype smart eyewear.

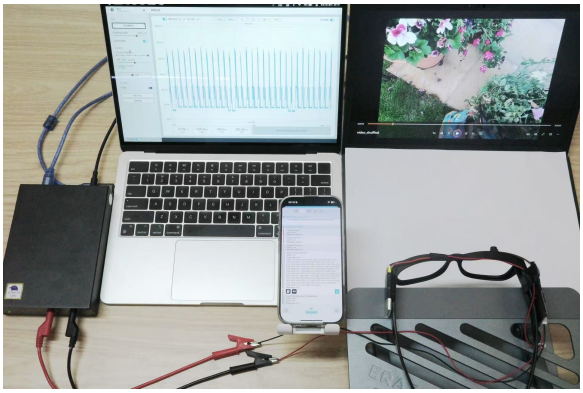

Fig. 4. Power measurement setup.

3.4 Energy Model And Power Measurement Methodology

We first explain the energy model on the eyewear side and theoretically demonstrate how our system reduces energy consumption compared to a conventional framework without *ActiveEye*. We then describe the methodology used for direct power measurements.

3.4.1 Energy Model. The total energy consumption on the eyewear side, denoted as E_{total}, is calculated as

$$E_{\text{total}} = E_{\text{sense}} + E_{\text{compute}} + E_{\text{transmit}} + E_{\text{base}},$$

where E_{base} accounts for background system-level energy consumption. The other three terms – E_{sense}, E_{compute}, and E_{transmit} – are attributed to two distinct tasks introduced by *ActiveEye*:

- Task_{h}: captures high-resolution frames at a scheduled frame rate set by *ActiveEye*, performs JPEG compression, and transmits compressed images. The corresponding power consumptions are denoted as P^h_{sense}, P^h_{compute}, and P^h_{transmit}, respectively.
- Task_{l}: continuously captures low-resolution frames, estimate optical flow, and transmit flow data during the intervals between Task_{h}. The corresponding power consumptions are denoted as P^l_{sense}, P^l_{compute}, and P^l_{transmit}, respectively.

The total energy consumption is then given by:

$$E_{\text{total}} = (P^h_{\text{sense}} + P^h_{\text{compute}} + P^h_{\text{transmit}}) \times T_h + (P^l_{\text{sense}} + P^l_{\text{compute}} + P^l_{\text{transmit}}) \times T_{\text{l}} + P_{\text{base}} \times (T_h + T_l) \tag{10}$$

According to empirical measurements described in Section 4, the duration of Task_{l} T_l is approximately 6.71× longer than that of T_h. Moreover, the power consumption values during Task_{h}—namely P^h_{sense}, P^h_{compute}, and P^h_{transmit}—are consistently higher than their Task_{l} counterparts (typically 4.02×, 2.84×, and 1.13× greater). This design ensures that the system operates efficiently, with the majority of the time spent in the low-power mode.

3.4.2 *Power Measurement Methodology.*

(1) Eyewear side. We use a BlueBird BLU939 power meter to measure energy consumption of the ESP32-CAM-based eyewear. This power meter is selected for its high sampling resolution and wide dynamic range, which are essential to capture power fluctuations caused by intermittent high-resolution frame capture, BLE communication bursts, and continuous low-power consumption sensing. To reflect realistic usage, we randomly shuffle and replay the video sequences from the EgoSchema dataset, feeding each frame directly into the eyewear's camera input, as shown in the measurement setup (Figure 4). The video sequence was continuously looped until the battery was fully depleted, enabling continuous measurement of energy consumption under sustained real-world usage. This setup offers several benefits. First, EgoSchema provides structured ground-truth semantic annotations labeling environmental and activity changes in a narrative format, ensuring consistent and reproducible evaluation of both power consumption and system effectiveness across varied contexts. Secondly, EgoSchema covers a wide range of real-world, head-worn scenarios, allowing us to assess power usage under representative conditions. Thirdly, the shuffled full-set playback captures a broad spectrum of environments (e.g., indoor-outdoor transitions) and activity dynamics (e.g., fast-changing and low-motion segments), enabling robust evaluation over long-term use. Most importantly, this video-driven power testing setup provides a direct, controlled, and ethical alternative to human subject testing, which is intrusive and difficult to repeat under consistent conditions. We repeat this measurement process three times and report the average results to ensure reliability.

(2) Mobile side. We use Apple's official `sysdiagnose`[3] tool to measure the power consumption on the mobile device. The smartphone is operated under conditions identical to those during the eyewear measurements, running our system to process incoming data streams. This setup allows for simultaneous power measurement on both wearable and mobile components, enabling a synchronized characterization of the system-wide power profile. As with the eyewear measurements, the process is also repeated three times, and average values are reported to ensure reliability.

4 Evaluation

This section begins with an evaluation of the system's performance in controlled laboratory environments, and then proceeds to assess its practical performance, along with real-world user experience and usability.

[3] https://it-training.apple.com/tutorials/support/sup075/

4.1 In-lab Evaluation

4.1.1 Experimental Setup. We evaluate *ActiveEye* on the EgoSchema [52] dataset to assess its energy efficiency, real-time visual and motion semantic change detection, and video understanding capabilities. EgoSchema is an egocentric long video QA dataset comprising over 5,000 three-minute video clips sampled from the large-scale Ego4D dataset [28]. It spans a wide range of everyday scenarios, such as cooking, cleaning, eating, and gardening. This diversity ensures that evaluations based on this dataset are comprehensive and not biased toward specific scenarios. Additionally, the Ego4D dataset includes scene annotations for each video, allowing us to analyze *ActiveEye*'s workload across different scenarios.

To evaluate *ActiveEye*'s real-time detection of visual and motion semantic changes, we utilize the dense narrations provided in the Ego4D dataset. These narrations annotate the camera wearer's activities with precise timestamps, such as "232.93 seconds, C walks out of the bedroom" and "240.16 seconds, C climbs down the stairs," where C refers to the camera wearer. Since semantic changes in egocentric videos are predominantly driven by the wearer's actions, such as transitioning between scenes or performing new activities, we use these activity timestamps as ground truth for evaluation. While other factors, such as lighting variations or independent object motion, can also induce semantic changes, they are less frequent and not explicitly annotated in large-scale egocentric datasets. Therefore, we adopt camera wearer activity timestamps as a practical approximation for evaluating real-time semantic change detection.

Lastly, EgoSchema associates each video clip with a question and five answer choices, one of which is correct. Accurately answering these questions requires models to comprehend diverse actions and visual scenes while performing complex temporal reasoning [52]. Thus, the dataset is well-suited for evaluating *ActiveEye*'s video understanding capabilities. Due to GPU memory constraints, it is not feasible to encode all high-resolution frames corresponding to the visual and motion semantic changes detected by *ActiveEye* into the context window of the LLM decoder. Instead, we select the top N frames with the largest prediction errors, as these frames are expected to represent the most significant semantic changes and provide the greatest incremental information relative to *ActiveEye*'s prior knowledge. This approach also allows for a fair comparison with other VLM agents that utilize different numbers of frames as input by adjusting the value of N.

4.1.2 Metrics. We simulate *ActiveEye*'s input using video clips from the EgoSchema dataset and evaluate its energy efficiency based on the average power usage of sensing, computation, and transmission.

For real-time detection of visual and motion semantic changes, we evaluate *ActiveEye* using time difference and average F1 score as metrics. The time difference quantifies system responsiveness by measuring the mean distance between each annotated timestamp and the nearest detected timestamp. The F1 score, balancing precision and recall, assesses the system's ability to accurately and comprehensively detect semantic changes. Following prior event boundary detection methods [61, 62], we calculate the relative distance (*Rel.Dis.*), which normalizes the timestamp difference by the duration of the corresponding action or event. A match is considered correct if the *Rel.Dis.* is within thresholds ranging from 0.05 to 0.5, accounting for variability in human annotations. Precision, recall, and F1 score are computed at each threshold, and their averages are reported.

For the VQA task, performance is assessed using both accuracy and processing latency. Here, processing latency is defined as the elapsed time from the reception of the input (visual data and query) to the generation of the VQA output. This metric isolates the computational cost of the VQA reasoning pipeline, enabling reproducible, lab-based comparisons.

4.1.3 Baselines.

(1) **Uniform Sampling.** OS-1 [74] delivers a function similar to our system – responding to users' conversations while taking into account the users' immediate visual surroundings. However, a direct comparison with OS-1 is not feasible, as the two systems have different objectives and input modalities. To enable a

fair comparison, in this work, we slightly modify the OS-1 method: we adopt its fixed frame sampling rate mechanism and disabled its speech input. We refer to this modified baseline as Uniform Sampling.

(2) **Heuristic Event Boundary Detection Methods.** Our approach leverages both ongoing motion semantics and historical visual semantics to identify semantic changes. To evaluate this design, we compare it against two heuristic event detection methods for consecutive frames, both based on optical flow from FlowGEBD [27]: (1) Pixel tracking with sparse optical flow (FlowGEBD-PT), which detects event boudaries when the ratio of tracked pixel points falls below a threshold θ_1, indicating a semantic change in scene. (2) Monitoring frame-by-frame dense optical flow maxima (FlowGEBD-FN), which detects an event when the normailized maximum flow value across frames exceed a threshold θ_2. For evaluation, we set θ_1 to 0.4 and θ_2 to 0.06 to achieve an optimal balance between precision and recall.

(3) **VLM Agents with Frame Selection Mechanism.** We also compare *ActiveEye*'s video understanding capabilities on the VQA task with several state-of-the-art VLM agents: VideoAgent [68], VideoAgent [22], MoReVQA [54], and VideoTree [70]. It is important to note that these baseline methods are primarily designed for offline VQA, as they operate on pre-captured video clips. Therefore, they are excluded from our evaluation of real-time semantic detection.

4.1.4 *Results.*

(1) Overall Performance. Table 1 reports the energy efficiency and real-time semantic detection metrics of *ActiveEye* and the baseline methods. As shown, *ActiveEye* achieves a 49.14% energy saving compared to the VLM system that uniformly samples video frames at 384×384@4fps. This efficiency is attributed to the adaptive spatial-temporal sampling rates for extracting visual and motion semantics and transmitting high-resolution frames only when semantic changes are detected, reducing the average power consumption for sensing, computation and transmission by 59.70%, 72.91% and 85.83%, respectively. Additionally, *ActiveEye* achieves the highest F1 score of 0.80 and the lowest time difference of 1.30 s compared to heuristic event boundary detection algorithms. This advantage arises from its ability to predict motion features, allowing it to model high-level motion semantics over time while being less susceptible to disturbances from egomotion. In contrast, heuristic optical flow-based methods are highly sensitive to egomotion, leading to frequent false detections when the wearer is in motion and failing to capture subtle hand movements when the wearer remains still.

Table 1. Energy efficiency and real-time performance comparison between *ActiveEye* and baseline methods.

	Real-time semantic detection metrics				**Energy efficiency metrics**			
Method	Precision	Recall	F1	TimeDiff (second)	P_{sense} (mW)	$P_{compute}$ (mW)	$P_{transmit}$ (mW)	P_{total} (mW)
Uniform sampling	0.13	**1.00**	0.23	**0.06**	66.50	48.94	176.89	516.25
FlowGEBD-PT	0.62	0.57	0.59	6.08	25.18	14.40	**21.75**	262.83
FlowGEBD-FN	0.60	0.54	0.56	4.34	25.21	29.95	21.76	278.15
Ours	**0.81**	0.80	**0.80**	1.30	**22.79**	**13.26**	25.07	**262.56**

To illustrate this, Figure 5 presents a case study from an egocentric video where the camera wearer is painting. The figure visualizes a sequence of video frames, annotated narrations with timestamps, and the timestamps detected by *ActiveEye* and baseline methods. In this scenario, the primary source of motion stems from the camera wearer's hand movements, such as "puts art on table" and "stirs the paint brush in the watercolor pan." FlowGEBD-PT fails to detect these changes, as it relies on tracking sparse points and monitoring their

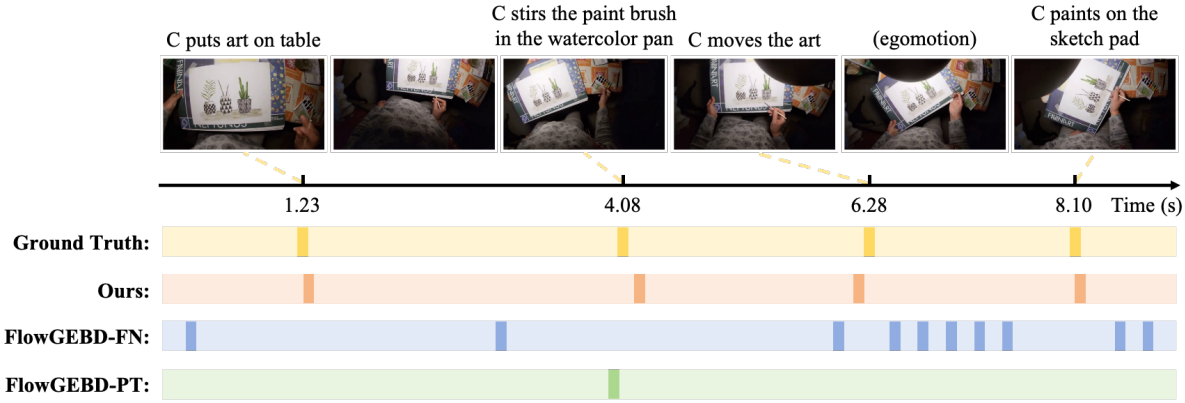

Fig. 5. Case study of semantic change detection during a painting activity.

disappearance. Meanwhile, FlowGEBD-FN, being sensitive to egomotion, misinterprets slight head movements between "moves the art" and "paints on the sketch pad" as significant changes, leading to consecutive false detections. In contrast, *ActiveEye* effectively captures motion semantics through feature extraction and prediction, allowing it to model subtle motion variations in relatively static scenes while remaining robust to low-level disturbances such as egomotion.

(2) Battery Performance. We also provide energy consumption profiles for both the eyewear and smartphone sides, as shown in Figure 6. In this figure, we visualize the voltage of the eyewear and the remaining smartphone battery percentage (on a fully charged 3582 mAh iPhone 16 Pro battery) during the continuous operation while replaying the EgoSchema dataset. The voltage profile demonstrates a near-linear decline, consistent with typical lithium-ion discharge profiles, which indicates stable performance without erratic power fluctuations. Specifically, the voltage of the eyewear drops from approximately 4.11 V to 2.81 V, indicating complete battery depletion at around 8.37 hours of continuous use. It is necessary to note that during the final phase of discharge (approximately from 8 to 8.37 hours), the rate of voltage drop accelerates. This is due to the depletion of internal charge, which prevents the system from maintaining a stable current. As a result, the voltage drops sharply from 3.33 V to 2.81 V, and the system ceases operation due to insufficient power supply. On the other hand, the smartphone's battery percentage decreases only modestly from 100% to approximately 82.7% over the same duration. This indicates that the proposed system imposes a minimal power burden on the smartphone, allowing it to comfortably sustain a full day of active use even with continuous interaction with the eyewear. We also provide a concrete example in Appendix A1 to illustrate how power consumption fluctuates during task execution in a typical working-at-desk scenario.

To further analyze *ActiveEye*'s battery performance across different scenarios, we summarize *ActiveEye*'s battery performance across 10 usage scenarios. These scenarios are selected from EgoSchema dataset based on the highest number of video samples, ensuring they represent typical daily activities. Figure 7 shows the average power consumption of the eyewear and the frequency of Task_h activations for each scenario. As expected, there is a clear correlation: scenarios with more frequent Task_h activations tend to consume more power. Dynamic activities (i.e., fast-changing scenarios) such as *Farming* and *Construction* involve frequent visual changes, resulting in higher energy use. In contrast, more static activities, such as *Phone Use*, allow the system to remain mostly in Task_l, thereby conserving energy.

To better illustrate contextual differences, we present two representative case studies:

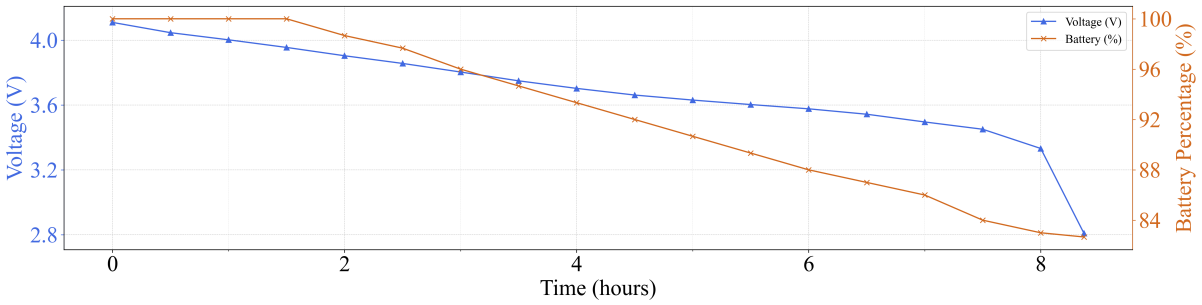

Fig. 6. Voltage depletion of eyewear and remaining smartphone battery percentage during continuous operation of the EgoSchema dataset replay.

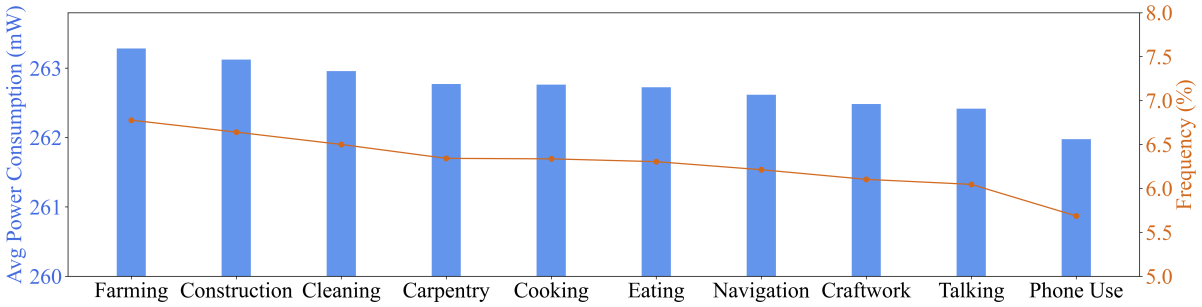

Fig. 7. Average power consumption and Task_h activation frequency of *ActiveEye* across ten daily scenarios.

- Playing tennis (dynamic): Frequent semantic changes lead to sustained Task_h. Average power is 273.88 mW, yielding 7.67 hours of battery life (see Figure 8).
- Desk work (static): Minimal scene change triggers Task_h only once, resulting in a lower average power draw of 256.65 mW and extending battery life of 8.18 hours (see Figure 9).

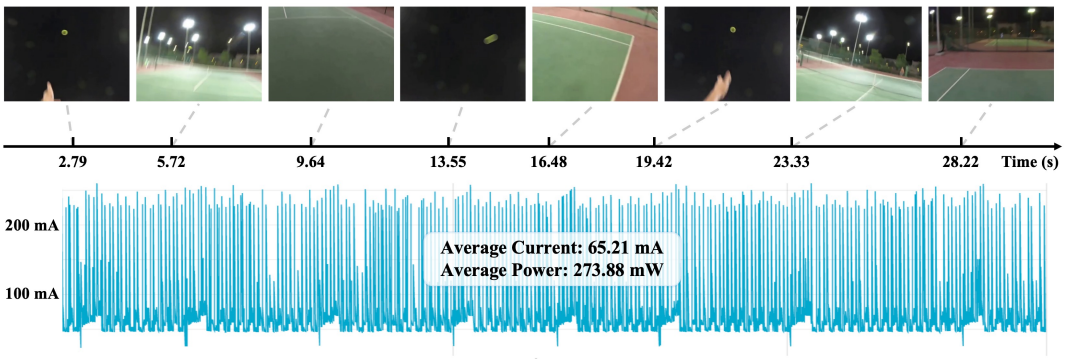

Fig. 8. Active energy trace in a fast-changing scene (playing tennis).

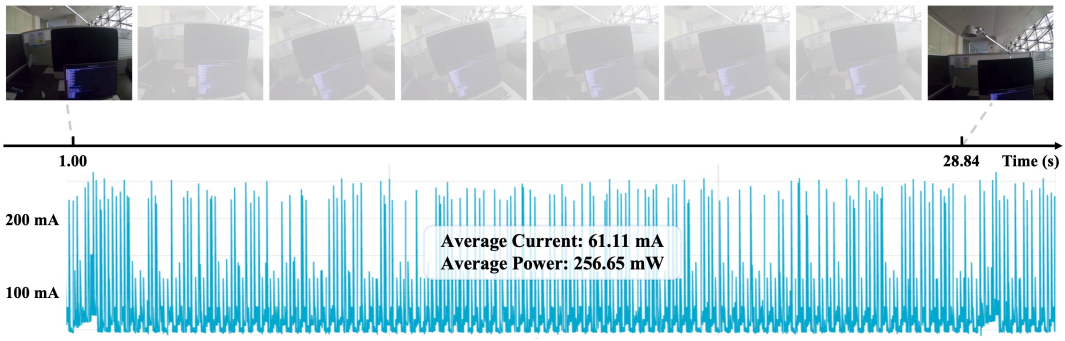

Fig. 9. Active energy trace in a slow-changing scene (desk work).

Table 2. VQA Accuracy Comparison for *ActiveEye* and Baselines.

Method	VLM Backbone	Num of Frames	Acc (Subset[4])	Acc (Full Set[5])
Uniform Sampling	LLaVA-OV-7B	32	62.4	60.3
VideoAgent (Wang *et al.*, 2024)	GPT-4	8.4	60.2	54.1
VideoAgent (Fan *et al.*, 2024)	GPT-4	360	62.8	60.2
MoReVQA (Min *et al.*, 2024)	PaLM-2	30	–	51.7
VideoTree (Wang *et al.*, 2024)	GPT-4	62.4	**66.2**	61.1
Ours	LLaVA-OV-7B	8	62.6	59.3
Ours	LLaVA-OV-7B	32	65.4	**61.6**

(3) Video Understanding Capability. We first evaluate *ActiveEye*'s video understanding capabilities under a controlled setting where both *ActiveEye* and baseline methods are given the same number of high-resolution frames. Table 2 presents the results on the EgoSchema VQA task. As shown, *ActiveEye* achieves higher accuracy than uniform sampling under this constraint, demonstrating that the use of motion features and feedback-driven adaptive frame selection enables the VLM to extract richer semantic information from video inputs. In addition, *ActiveEye* achieves comparable or better VQA accuracy than several VLM agents while using the same or even fewer high-resolution frames, validating its effectiveness in video understanding.

In the second experiment, we adopt a more realistic setting where the total number of frames is fixed, but *ActiveEye* adaptively selects only a small portion of high-resolution frames, while the rest are low-resolution, whereas the uniform sampling baseline uses all high-resolution frames at 4 fps. This setting better reflects real-world deployment constraints where energy and latency are critical. To further distinguish the effects of the *ActiveEye* from those of the underlying VLM backbones, we retrain both *ActiveEye* and a representative baseline method—Uniform Sampling—on *EgoGPT-7B*, allowing for a fair comparison under the same backbone and training conditions. This is because directly comparing our approach on the exact same VLM backbone used by other baselines (e.g., GPT-4, PaLM-2) (Table 2) is not feasible. These models are closed-source and accessible

[4]The subset contains 500 questions with publicly available ground-truth answers for offline evaluation in EgoSchema.
[5]The full set includes all questions in EgoSchema, including those in the subset. The remaining ones require online submission for scoring.

only through restricted APIs. Consequently, they do not support feature-level customization or gradient-based learning, which are both essential for training components such as the motion connector.

Furthermore, due to GPU memory constraints when feeding inputs to the LLM decoder, we apply a uniform spatial downsampling to the visual features in both systems, and to the motion features in *ActiveEye* only. This adjustment ensures comparability without favoring either method. We note that this downsampling is only introduced for benchmarking in a controlled setting; *ActiveEye* does not require such downsampling in typical deployment.

Table 3 shows the results, indicating that under the same VLM backbone, our method significantly reduces power consumption—by 49.1% for *LLaVA-OV-7B* and 48.7% for *EgoGPT-7B*—while maintaining competitive VQA accuracy, with only a slight decrease of 1.0% and 1.9%, respectively. Also, *ActiveEye* achieves competitive accuracy while reducing processing latency by 24.6% and 25.7% across different LLM backbones, thereby validating our design goal of balancing accuracy with efficiency.

Table 3. Comparison of VQA performance and energy savings across different VLM backbones.

VLM Backbone	Method	Acc (Subset)	Acc (Full set)	Avg. Power (mW)	Processing Latency (s)
LLaVA-OV-7B	Uniform Sampling	61.4	59.5	516.25	3.21
	ActiveEye	63.6 ↑3.6%	58.9 ↓1.0%	262.56 ↓49.1%	2.42 ↓24.6%
EgoGPT-7B	Uniform Sampling	68.8	72.8	516.25	4.04
	ActiveEye	68.0 ↓1.2%	71.4 ↓1.9%	264.97 ↓48.7%	3.00 ↓25.7%

(4) Ablation Study. We further conduct ablation studies to evaluate the effectiveness of the two key components of *ActiveEye*: (1) decoupling of visual and motion representations, and (2) feedback-driven adaptive sampling. To ensure a fair comparison, all methods use the same input resolutions: high-resolution frames are captured at 384×384 pixels, and low-resolution frames at 128×128 pixels. All experiments are performed on the EgoSchema dataset for the VQA task, using the LLaVA-OV-7B model as the VLM backbone.

We compare *ActiveEye* against the following three ablation baselines.

- **w/o DE+FE**. This baseline removes both components—'decoupling of visual and motion features' (DE) and 'feedback-driven adaptive sampling' (FE). It uniformly samples high-resolution video frames at a fixed rate, which is the maximum rate supported by the cloud server, and feeds these frames directly into the VLM for the VQA task. Neither specific motion features nor adaptive frame selection is used in this baseline.
- **w/o DE**. This baseline removes only the 'decoupling of visual and motion features' mechanism while retaining 'feedback-driven adaptive sampling'. In this setup, it adaptively adjusts the rate of high-resolution video frame transmission based on motion prediction errors, similar to *ActiveEye*. Specifically, it first predicts future motion features based on the evolving video understanding (using the same prediction module as in *ActiveEye*), and then compares the prediction with the actual observed motion to compute the prediction error. Based on the error, we dynamically select a subset of high-resolution frames to transmit, within the capacity of the cloud server. Unlike *ActiveEye*, however, this baseline does not explicitly separate the visual and motion feature streams.
- **w/o FE**. This baseline removes only 'feedback-driven adaptive sampling' (FE) while retaining 'decoupling of visual and motion features' (DE). As in *ActiveEye*, this setup also processes the low-resolution frames at 4 fps to extract motion features, while high-resolution video frames are sampled and transmitted at the maximum frame rate supported by the cloud server. Thus, although motion and visual features remain

decoupled in this baseline, the high-resolution visual stream operates at a constant frame rate and does not adapt based on feedback from the motion stream.

Table 4. Ablation study of *ActiveEye* on the EgoSchema dataset for the VQA task.

Method	Module			Acc	
	VLM backbone (LLaVA-OV-7B)	FE	DE	Subset	Full Set
w/o DE+FE	✓	–	–	62.4	60.3
w/o DE	✓	✓	–	62.8	60.8
w/o FE	✓	–	✓	64.0	60.8
Ours	✓	✓	✓	**65.4** ↑2.2%	**61.6** ↑1.3%

Table 4 presents the ablation study results, demonstrating the effectiveness of the proposed two components—decoupling of visual and motion features (DE) and feedback-driven adaptive sampling (FE). We can see that our method achieves the best performance on both the subset and the full set of the EgoSchema for the VQA task, with accuracy improvements of 1.4 and 0.8 (corresponding to relative gains of 2.2% and 1.3%, respectively), compared to the strongest baselines. These results prove the indispensable role of both DE and FE in enhancing system performance. Removing either DE or FE individually (w/o DE and w/o FE) still improves performance compared to removing both components (w/o DE+FE), demonstrating that each component independently contributes to better video understanding.

These results highlight the combined effect of motion-visual decoupling and adaptive sampling in improving wearable video-based VQA.

4.1.5 Parameters Tuning and Trade-Offs. To offer deeper insights, we report the key parameter tuning choices and discuss the resulting trade-offs.

(1) Impact of ψ_e (Trade-off of energy vs. accuracy under varying temporal resolution.) A key parameter of *ActiveEye* is the threshold ψ_e, which quantifies the difference between the observed motion and predicted motion. This threshold determines when the system triggers the most recent visual and motion information capture. In other words, ψ_e governs the temporal resolution of the system: A smaller ψ_e results in more frequent high-resolution sampling, thereby increasing both power consumption and task performance (measured by VQA accuracy).

Figure 10 illustrates the trade-off between power consumption and accuracy as ψ_e varies from 1.0 to 1.6 (step size 0.1), constrained by the 24 GB memory of the NVIDIA GeForce RTX 3090 GPU used in the cloud. On the EgoSchema dataset, we observe that average power consumption rises due to more frequent high-resolution captures as ψ_e decreases, while accuracy also improves accordingly (with minor fluctuations). Notably, the accuracy increases up to about $\psi_e \approx 1.2$, beyond which it plateaus or slightly declines—showing a classic diminishing-returns effect. These results suggest that lowering ψ_e allows the system to incorporate more high-resolution frames, enriching semantic information and improving VQA accuracy. However, once the essential semantic content is adequately captured, further lowering ψ_e yields little to no benefit while incurring additional energy cost. Therefore, in this work, we empirically set ψ_e as 1.2, balancing accuracy and energy efficiency. In Figure 10, this point is highlighted by the orange star.

(2) Trade-off of energy vs. accuracy under varying spatial resolutions. We further investigate the energy–accuracy trade-off induced by spatial resolution for two tasks: Task_l and Task_h, which exhibit distinct

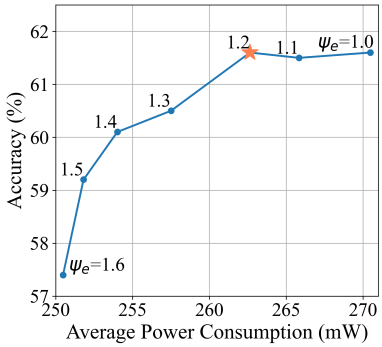

Fig. 10. Impact of ψ_e on VQA accuracy and average power consumption.

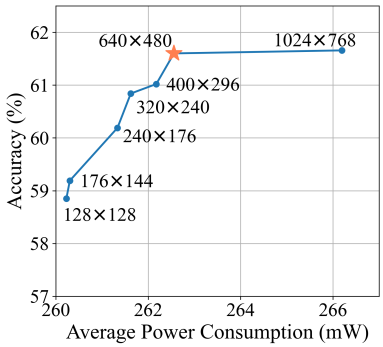

Fig. 11. Energy–accuracy trade-off for Task_h at different spatial resolutions.

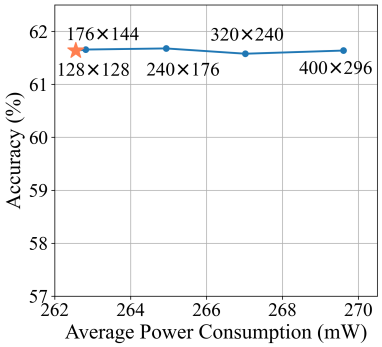

Fig. 12. Energy–accuracy trade-off for Task_l at different spatial resolutions.

resolution requirements. While higher resolutions enhance semantic richness, practical constraints of model input size and camera hardware impose limits.

We first fix the resolution of Task_l at 128×128, as indicated by widely-used optical flow models [64], and vary Task_h across 7 candidate resolutions supported by the ESP32-CAM: 128×128, 176×144, 240×176, 320×240, 400×296, 640×480 and 1024×768. Prior to inference, all Task_h frames are resized to 384×384, the vision encoder's required input. As expected, increasing the capture resolution improves accuracy but raises energy usage, with diminishing returns beyond 640×480 in Figure 11. This suggests 640×480 as a balanced resolution for Task_h.

We then examine the optimal resolution for Task_l while keeping Task_h fixed at 640×480. Task_l's resolution is varied among the above candidate sizes, lower than 640×480. Results in Figure 12 show that increasing Task_l's resolution beyond 128×128 causes a significant increase in average power consumption with minimal gains in VQA accuracy improvements. This is because *ActiveEye* extracts sparse optical flow from the low-resolution stream, and since motion within local regions is typically spatially coherent [21], high-resolution inputs offer limited additional benefits. Based on these observations, we adopt 128×128 and 640×480 as the resolutions for Task_l and Task_h, respectively, in previous experiments.

4.2 In-field Study

4.2.1 Experimental Setup. To evaluate the system performance, usability, and user experience in practice, we conduct two rounds of in-field user studies, each designed with distinct objectives. The first-round study focuses on system deployment validation and real-time performance testing. Participants are recruited from a small, lab-level population to allow rapid feedback and iterative refinement. While this sample is homogeneous, it serves the purpose of early-stage system evaluation, and we only use the data for evaluating system performance from a perspective independent of individual participant traits. The second-round study aims to evaluate system usability and user experience in various real-world contexts. We recruited the participants from a broader university-level population to enhance diversity in gender, age, and disciplinary background (including both Computer Science (CS) and Non-CS[6] fields).

A summary of participant demographics is provided in Table 5. Participants from the first-round study are labeled P1 to P10, and those from the second round are labeled P11 to P25.

[6]Non-CS fields in this work include Industrial Design, Atmospheric and Oceanic Sciences, Physics, Art Conservation, Business Administration, Microelectronics, and Electronic Information.

Table 5. Summary of Participant Demographics Across Two User Study Rounds.

Round	No. of Participants	Gender	Age Range	Background
1	10	10 Males	21–24	All Computer Science (CS)
2	15	8 Males, 7 Females	22–36	7 CS, 8 Non-CS

Each study begins with an introductory session, where participants were briefed on system usage, data review procedures, and the evaluation interface for submitting feedback. In Round 1, we predefined eight daily scenarios—commuting, shopping, exercising, housework, working, walking, dining, casual chat—to ensure comparability across participants. Each participant was instructed to engage in a minimum of 10 minutes of conversation with *ActiveEye* per scenario, completing at least five dialogue rounds, with the dialogue content grounded in the ongoing visual context. This controlled setup minimized variations due to personal background or context. In Round 2, participants were asked to use the system in self-selected naturalistic scenarios encountered in their daily lives. They were also encouraged to include both indoor and outdoor settings, as well as a mix of dynamic environments (with rapidly changing visual semantics) and static environments (with stable or slowly evolving visual content). This more flexible setup enabled the collection of richer, context-aware usability data and supported a broader, more inclusive evaluation of user experience.

4.2.2 *Evaluation Method.*

1) System Performance Evaluation Method. We evaluated *ActiveEye* in terms of real-time capturing accuracy, real-time responsiveness, and energy efficiency across different usage scenarios. To assess real-time capturing accuracy, we developed a web-based interface that allowed participants to review their interactions with *ActiveEye* alongside the high-resolution images captured during predicted semantic changes. Participants were asked whether *ActiveEye* correctly identified semantic changes relevant to the ongoing conversation and captured corresponding high-resolution frames, providing binary ("Yes" or "No") feedback. We define the Hit Rate as the percentage of correctly detected semantic changes and use it to quantify real-time capturing performance. For real-time responsiveness, we measured user-perceived latency, defined as the time interval between the end of the user's speech input and the onset of the system's audio feedback. Finally, we assessed energy efficiency using the average power consumption of *ActiveEye* during each conversational scenario.

2) User Experience Evaluation Method. In this work, we define user experience (UX) as user's perceptions and responses arising from their interaction with the system in real-world scenarios [39], including aspects such as perceived usefulness, conversational quality, emotional connection, interaction design, and future adoption intent. To evaluate these dimensions, we conducted a comparative study involving *ActiveEye* and two baseline systems: (1) Uniform Sampling, which uses a fixed 10-second image capture interval from prior work [74], and (2) Reactive Sampling, which captures images only when the user initiates a dialogue. These two baselines were selected because they represent different approaches to interactive perception and share comparable capabilities for vision-language dialogue, making them suitable for comparative assessment. Participants interacted with all three systems in self-selected real-world scenarios. To mitigate order effects, we used a Latin square design[7], ensuring that each system appeared in different positions across participants. For each scenario, participants were asked to engage in at least five rounds of dialogue lasting a minimum of 10 minutes, with conversations grounded in their surrounding visual contexts. Following the interactions, participants took part in a semi-structured interview (around 30 minutes), which captured qualitative insights into their experience across systems.

[7] https://en.wikipedia.org/wiki/Latin_square

3) System Usability Evaluation Method. While the UX evaluation addressed broader experiential aspects across the three systems, we also conducted a separate assessment focusing on perceived usability, specifically for our system. To this end, we employed System Usability Scale (SUS), a widely-adopted, industry-standard tool for evaluating perceived usability [42], which is particularly well-suited for assessing early-stage prototypes [41]. We administered the SUS questionnaire to all 15 participants in Round 2, following their interaction with *ActiveEye*. The SUS comprises 10 standard items—such as whether the system is unnecessarily complex, requires technical support, or feels cumbersome to use—each rated on a 5-point Likert scale (1 = "strongly disagree", 5 = "strongly agree"). These responses are converted into a composite SUS score ranging from 0 to 100, with a score of 68 being generally regarded as the benchmark for average usability [42].

4.2.3 Performance of ActiveEye in Practice. We present the conversation duration, number of valid rounds, Hit Rate, average power consumption, and real-time responsiveness for each scenario in the first-round in-field user study, as shown in Table 6.

Table 6. Evaluation of *ActiveEye* performance across eight daily scenarios in the first-round in-field user study.

Scenario	Duration (hours)	Rounds	Hit Rate (%)	Avg. Power Consumption (mW)	Responsiveness (seconds)
Commuting	0.95	73	95.89	262.17	1.78
Shopping	0.95	145	88.27	260.55	1.69
Exercising	0.44	70	91.42	261.19	1.67
Housework	0.94	110	88.18	260.22	1.57
Working	1.17	58	93.10	256.72	1.55
Walking	1.77	143	92.30	260.55	1.71
Dining	1.21	76	96.05	259.58	1.64
Casual Chat	1.36	153	86.93	257.04	1.35
Total	**8.80**	**828**	**90.70**	**259.58**	**1.60**

From the experimental results, we draw the following conclusions:

(1) ActiveEye Enables Accurate and Continuous Video Perception with Dynamic Adaptation to Scene Changes. Across all scenarios in the in-field study, *ActiveEye* achieves an average Hit Rate of 90.70%, indicating a strong capability in detecting semantic changes relevant to ongoing conversations. It also maintains an average user-perceived latency of 1.60 seconds, which falls within the empirically established range (0–2 seconds) considered acceptable for natural interactions in conversational systems [26]. Furthermore, *ActiveEye* sustains an average power consumption of 259.58 mW, with higher usage observed in dynamic scenarios (e.g., commuting, shopping, exercising) and lower in more static ones (e.g., working, dining, casual chatting), consistent with patterns from the controlled lab study.

(2) Real-Time Responsiveness vs. Energy Consumption. Using data from the first-round user study, we examined the trade-off between real-time responsiveness and energy consumption in the conversational application. As illustrated in Figure 13, we observe a clear positive correlation between latency and energy consumption. Higher user-perceived latency—indicating slower system responsiveness—was associated with higher energy usage. This relationship is intuitive, as higher energy consumption typically reflects a greater number of processed Task_h

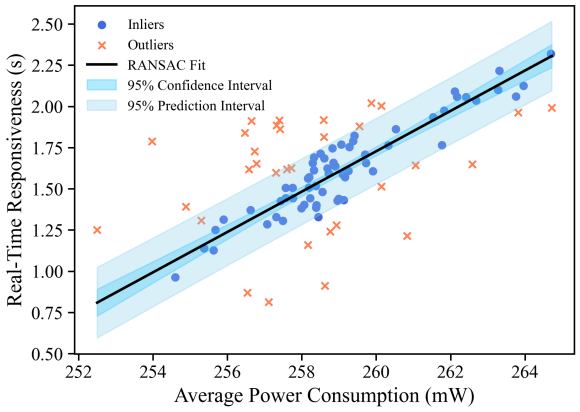

Fig. 13. Relationship between average power consumption and latency. RANSAC regression [53] identified around 66% of data points as inliers (blue circles), showing a strong linear trend (R^2 = 0.878). Outliers (orange crosses) were excluded from model fitting.

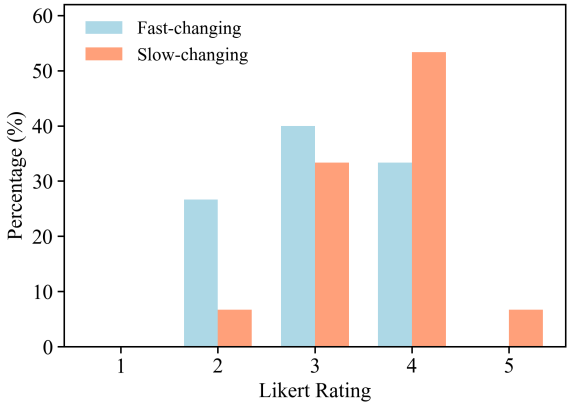

Fig. 14. Distribution of latency acceptability ratings. Users rated slow-changing tasks as more tolerable, with 60.0% assigning a rating of 4 or 5. In contrast, fast-changing tasks were rated less favorably, with the highest proportion (40.0%) selecting a mid-range rating of 3.

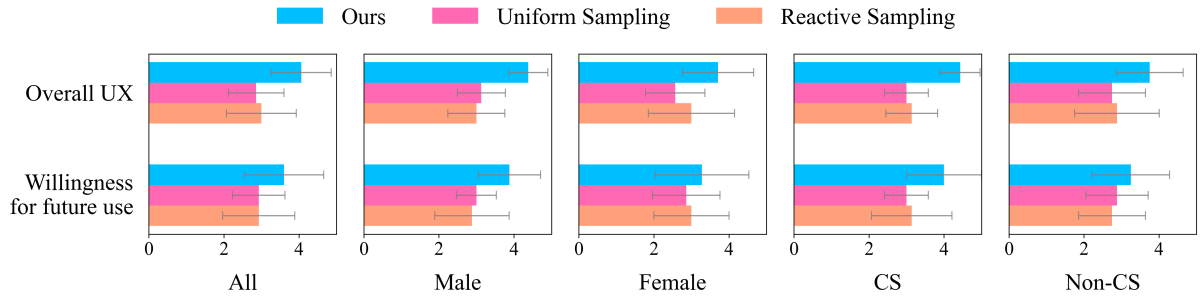

Fig. 15. UX and willingness ratings across user groups.

events. These involve more frequent image sensing, JPEG compression, and bursts of BLE communication, all of which contribute to increased processing time and, consequently, longer perceived latency.

We also examine outliers to gain further insight. In cases where low power consumption was paired with high latency, this typically occurs when a user initiates speech while the scene semantics were simultaneously changing or just after the change—such as when the user was working or walking and turned their head to engage in a conversation with the system. In these situations, the system has already started processing updated visual and motion features or was doing so concurrently with the speech input. Although this leads to a longer wait time for the user (i.e., higher latency), the processing workload is spread over time, and high-resolution updates are not triggered excessively. As a result, energy usage remain relatively low, even though the response was delayed.

4.2.4 Findings on User Experience. **Good overall user experience and willingness for future use across gender and professional background.** As shown in Figure 15, all 15 participants reported a positive overall user experience with *ActiveEye* (4.07 ± 0.80). Meanwhile, they also agreed that *ActiveEye* offered a superior experience

compared to both baselines (2.87 ± 0.74 for Uniform Sampling and 3.00 ± 0.93 for Reactive Sampling). These trends were consistent across genders and professional backgrounds, suggesting that *ActiveEye* offered a broadly positive experience regardless of user demographics. Participants attributed *ActiveEye*'s superior experience to its ability to promptly detect changes in scene semantics, allowing it to adapt quickly to evolving visual contexts and generate more relevant responses. When asked to elaborate, 11 out of 15 participants specifically highlighted *ActiveEye*'s enhanced responsiveness to dynamic scene changes. One participant noted, *"ActiveEye was quicker to respond when the scene changed, making the conversation feel natural. With the other two systems, I had to manually guide the conversation, but ActiveEye just followed along."* (P19). Interestingly, participants' perceptions of system intelligence were also influenced by responsiveness—even though all systems were backed by the same LLM. As one participant put it: *"ActiveEye felt smarter because it picked up quickly on what I wanted it to look at"* (P15), while another commented, *"The uniform sampling system lagged behind in dynamic scenes—it was still talking about the previous one when I had already moved on, which made it feel dumb"* (P14).

With potential for future use but requiring further personalization and task support. Participants also reported a positive intention to use our system in the future (3.60 ± 1.06), exceeding both baselines (2.93 ± 0.70 and 2.93 ± 0.96). Qualitative feedback linked this adoption intent to our system's ability to proactively initiate topics based on its timely and continuous perception of the environment. This capability fostered a stronger sense of engagement during interactions, distinguishing it from question-answer-based systems that rely solely on user prompts. As one participant noted, *"When ActiveEye saw me jogging in the hallway, it would suggest going for a run outside. Other systems typically only respond after I ask something."* (P16). These patterns are also consistent across genders and professional backgrounds, suggesting early but promising adoption potential. However, participants also expressed a desire for more customized (or personalized) styles and tighter integration with downstream tools, indicating that long-term use still depends on future improvements, such as personalization and task support.

Participants valued *ActiveEye* as a responsive conversational assistant and companion due to its continuous visual perception. Ten out of 15 participants reported that *ActiveEye* provided more helpful, contextually relevant responses by promptly detecting scene changes. For instance, one participant shared, *"It gave me suggestions based on what I was currently looking at in the store. One moment I was browsing snacks, and when I turned to the yogurt aisle, it recommended having yogurt in the afternoon"* (P25). Additionally, three participants noted that *ActiveEye*'s persistent awareness of scene dynamics served as a useful supplement to their own visual perception and memory—a benefit not provided by the other two baseline systems. We illustrated such a case in Appendix Figure A2. Furthermore, 8 out of 15 participants felt a stronger sense of connection with our system compared to baselines. This was attributed to the system's ability to "see what users see,", which enhanced the feeling of co-presence. (*"It gave me a sense of 'being there with me.'... felt like it was experiencing things alongside me"* (P23). An example of such interaction is visualized in Appendix Figure A3.

Participants also provided suggestions for its assistant and companion roles. For assistance, integration with tools (e.g., navigation, reminders) (9 out of 15) and extended long-term memory (4 out of 15) were requested (*"I wish it could connect to apps like navigation, ..., reminders, and even integrate with wearables like fitness bands"* (P22)). For companionship, participants emphasized the need for more personalized interaction styles (6 out of 15) and emotional support capabilities (3 out of 15). For example, *"Chatting with ActiveEye felt like being with a boyfriend who's physically present but emotionally unavailable."* (P23).

4.2.5 Findings on System Usability. **Good usability.** Participants generally found *ActiveEye* easy to use. *ActiveEye* achieved an average SUS score of 77.17 ± 10.26 (out of 100), as shown in Figure 16, exceeding the standard benchmark of 68 and indicating good usability. Across all 10 SUS items, participant responses were consistently positive. For example, they agreed that the system was easy to use (Item A3, 4.20 ± 0.41), quick to learn (Item A7, 4.67 ± 0.62), and not unnecessarily complex (Item A2, 2.20 ± 0.94). Additionally, participants expressed confidence

in using the system without external support (Item A4, 1.87 ± 1.25). As one participant noted: *"I didn't need much external help while using it—just briefly skimming the manual was enough, and the interface wasn't complicated."* (P12). Another echoed this sentiment: *"I found it fairly easy to get started with the help of the instructions—the interface was also quite clear."* (P14).

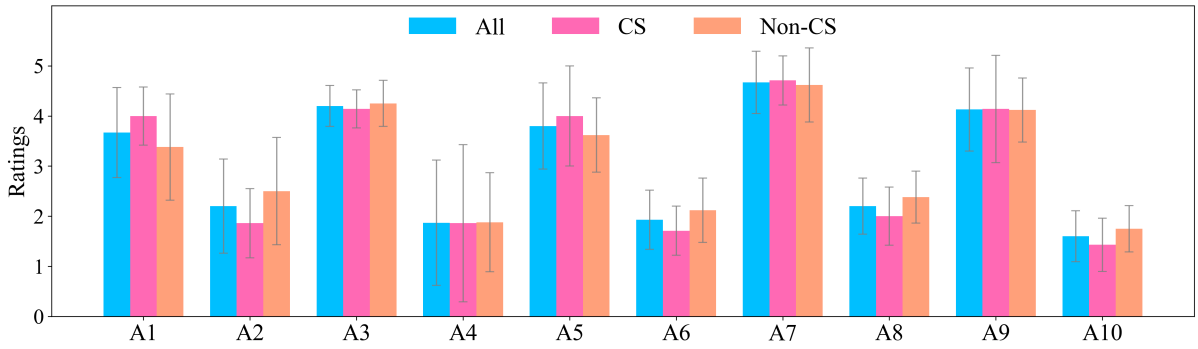

Fig. 16. SUS item ratings by participants. A1: willing to reuse; A2: too complex; A3: Easy to use; A4: needs support; A5: well integrated; A6: too inconsistent; A7: quick to learn; A8: cumbersome; A9: confidence in use; A10: steep learning curve. Higher scores reflect stronger agreement with positive usability for A1, A3, A5, A7, and A9, while lower scores reflect positive usability for reverse-worded items A2, A4, A6, A8, and A10.

Good usability across professional backgrounds, with higher ease of adoption among CS participants. As shown in Figure 16, *ActiveEye* was perceived as usable by both CS and non-CS participants, with both SUS scores exceeding the standard benchmark of 68. Specifically, CS participants reported a higher average score (81.43 ± 11.17) compared to the non-CS participants (73.44 ± 8.34). Moreover, CS participants scored more positively than non-CS participants on nine out of ten SUS items, indicating consistently greater ease of use and confidence. These trends suggest that while the system is generally usable, users with technical backgrounds found it easier to adopt—likely due to greater familiarity with digital tools and interaction patterns. As one CS participant noted: *"I have worked on similar systems before, so it was quite intuitive for me."* (P13). Another from an art conservation background remarked, *"The system felt a bit complex at first and wasn't easy to get started with, but it became fairly straightforward once I got used to it."* (P22).

This discrepancy may also be explained by differences in expectations. Non-CS participants may hold higher expectations regarding interface intuitiveness and device usability, especially for wearable systems. As one participant from an industrial design background noted: *"I think the mobile app interface could be simpler. Ideally, users shouldn't need to be told what to do."* (P18). Another participant from a business administration background commented: *"I felt the glasses were still a bit heavy and bulky for me—they seemed cumbersome to use"* (P23).

4.2.6 Latency Acceptability Across Contexts Discussion. We further investigated user latency tolerance by focusing on real-time responsiveness as experienced across diverse real-world scenarios, using data from the second-round user study. Different from the earlier discussion of isolated processing latency, real-time responsiveness captures the holistic user experience, shaped by factors such as network variability, device heterogeneity, and context-specific constraints. These real-world conditions can significantly influence how latency is perceived and tolerated in practice.

Participants (n=15) rated latency acceptability on a 5-point Likert scale, where 5 indicates strongly agree, 1 indicates strongly disagree that latency is acceptable. To access contextual effects, we categorized scenarios as fast-changing (e.g., shopping with rapidly evolving visual scenes) and slow-changing (e.g., desk working in a

stationary environment), reflecting differences in semantic update rates. Figure 14 shows the percentage of each rating across the two scenario types. **In slow-changing contexts, participants rated latency favorably**, with 60.0% giving scores 4 and 5. In contrast, **participants in fast-changing scenarios were more critical**: with 33.3% reporting a positive score of 4, 40.0% scored 3, and 26.7% scored 2, indicating lower satisfaction compared to the slow-changing scenario.

These findings align with our expectations and highlight how context dynamics influence user experience regarding latency perception, with fast-paced interactions presenting a greater challenge for real-time systems. Examples of supporting qualitative feedback include the following: *"When I was running... it responded about the stairs only after I had already gone down. That's too late to be functionally useful."* (P16) By contrast, in slow-changing scenarios, latency was less noticeable and thus more tolerable. *"While working or studying, I barely noticed any delay."* (P11)

5 Limitations

Participant sample size and diversity limitations. Recruiting participants for in-situ user studies is quite challenging and resource-intensive. In this work, we recruited 25 participants over a period of 3 months. While the study provided several informative findings, we acknowledge limitations related to both the scale and diversity of the participant sample. Although no substantial bias was identified during the study, this was not formally assessed, and further analysis would be necessary to draw more generalizable conclusions. *(1) Gender-related variation in user experience (UX).* Both male and female participants reported generally positive UX; however, some differences were observed. Specifically, female participants gave a lower average UX score (3.71 ± 0.95) than male participants (4.38 ± 0.52), as shown in Figure 15. Their willingness-to-continue ratings were also slightly lower. Follow-up open-ended responses revealed that some female users desired additional forms of support, such as more emotionally responsive interactions or integrated functional features (e.g., reminders or navigation). While these insights are informative, the limited number of female participants prevented us from determining whether such preferences are broadly representative. Further work is needed to explore whether this reflects a gender-specific expectation or a more general design consideration. *(2) Differences in perceived system usability.*Both groups gave favorable usability ratings; however, male participants rated the system marginally higher than female participants. This difference was most evident in SUS Item A1 ("I would like to use this system frequently") and Item A5 ("I found the system well integrated"). The lower score on Item A1 aligns with the UX responses and may suggest a lower intention to adopt the system over time among female users. Regarding Item A5, prior literature in design psychology suggests that male users may prefer structured layouts and visible feature sets, whereas female users may value clarity and simplicity in design [13]. Although these findings are preliminary, they raise important considerations for the design of inclusive systems. Nevertheless, the current sample size limits the extent to which we can draw conclusions about these observed differences.

Future work will focus on expanding the participant pool in both size and diversity to better understand how user characteristics influence UX and usability. This includes formally assessing potential gender-related preferences and examining whether observed differences reflect broader trends.

Contextual and Environmental Limitations. Our user study was conducted under typical daily-life conditions, including scenarios such as walking, shopping, and casual conversation. These scenarios represent common use cases for our system and generally allow for moderate tolerance to latency, as the interactions do not require immediate system responsiveness. However, they do not reflect time-critical or high-stakes contexts in which rapid visual understanding and low-latency response matter, e.g., navigating, reacting to safety hazards, or performing complex tasks in fast-changing environments. In such settings, abrupt semantic shifts in the visual scene may demand more immediate processing capabilities than were evaluated in this study.

Furthermore, our evaluation was carried out in moderate indoor and outdoor conditions, with limited variation in lighting and temperature. For lighting, we compared system power usage under daylight (12:00 noon) and reduced-light (6:00 pm) conditions, observing only a small increase in power consumption (0.72%) and no significant battery degradation. However, we do not examine extreme lighting environments (e.g., low-light at night or strong artificial lighting), which could affect sensor performance and semantic accuracy. Similarly, temperature conditions were stable (approx. 28.7°C), within the typical operating range for wearable devices. We do not investigate performance in extreme temperatures, such as cold winter environments, where lithium-ion battery efficiency and sensor behavior may vary. These contextual and environmental constraints highlight areas for future work to evaluate system performance in more demanding or diverse real-world conditions.

6 Privacy Considerations

Enabling smart eyewear with continuous visual perception introduces privacy concerns—not only for the wearer (e.g., private activities or locations) but also for nearby individuals [32, 55]. In our in-field study, we adopted the following measures to mitigate privacy risks: (1) Informed consent: Participants were clearly informed that semantic information (e.g., activities, objects, scene categories) would be processed. All participation is voluntary, with the right to withdraw at any time during the study. (2) Minimal raw data transmission: Raw video frames were transmitted only via Bluetooth to the paired smartphone. All feature extraction occurred locally, and only dense vectors were sent to the cloud—avoiding raw video upload over the internet. (3) User-controlled data management: Participants could review and choose whether to upload raw frames for analysis, and were given the option to delete all data after evaluation.

While these measures offer partial protection, risks remain—such as incidental capture of bystanders and the transmission of semantically rich features. Future work will explore more robust privacy-preserving strategies, including recording indicators [9], user-controlled capture via gestures or voice [36], and compression of features into codebook indices before transmission [71] to minimize semantic leakage.

7 Conclusions

To the best of our knowledge, *ActiveEye* is the first work to enable continuous and responsive video understanding on energy-constrained smart eyewear systems. Through an innovative VLM design that decouples visual and motion representations, *ActiveEye* extracts separate semantic information from video frames with varying spatial-temporal resolutions. It also introduces a feedback path that adaptively regulates front-end sampling and transmission rates based on the VLM's evolving scene understanding. We have implemented *ActiveEye* as an integrated wearable-mobile-cloud system and evaluated its performance through an in-lab study, demonstrating its effectiveness in improving energy efficiency, real-time semantic change detection, and video understanding. Furthermore, we have deployed *ActiveEye* in a smart eyewear-based conversational assistant and companion application, and conducted an in-field study to assess its real-world usability. The results demonstrate *ActiveEye*'s capability to enable continuous and responsive experiences in these applications while maintaining accurate video semantic understanding and high energy efficiency.

Acknowledgments

This work was supported in part by the Department for Science, Innovation and Technology under Grant K250071-101. Yujiang Wang was supported by a Basic Research Program of Jiangsu (BK20240414) and a Leadership Talent Program (Science and Education) of Suzhou Industrial Park (KJQ2024204).

References

[1] Sathyanarayanan N Aakur and Sudeep Sarkar. 2019. A perceptual prediction framework for self supervised event segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1197–1206.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems* 35 (2022), 23716–23736.

[3] Yiannis Aloimonos. 2013. *Active perception.* Psychology Press.

[4] Russell L Andersson. 1986. Living in a dynamic world. In *Proceedings of 1986 ACM Fall joint computer conference.* 97–104.

[5] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. 2024. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413* (2024).

[6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966* (2023).

[7] Ruzena Bajcsy. 1988. Active perception. *Proc. IEEE* 76, 8 (1988), 966–1005.

[8] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. 2018. Revisiting active perception. *Autonomous Robots* 42 (2018), 177–196.

[9] Taryn Bipat, Maarten Willem Bos, Rajan Vaish, and Andrés Monroy-Hernández. 2019. Analyzing the Use of Camera Glasses in the Wild. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–8. doi:10.1145/3290605.3300651

[10] Jean-Yves Bouguet et al. 2001. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel corporation* 5, 1-10 (2001), 4.

[11] Johanna Carvajal, Conrad Sanderson, Chris McCool, and Brian C Lovell. 2014. Multi-action recognition via stochastic modelling of optical flow and gradients. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis.* 19–24.

[12] Yuhu Chang, Yingying Zhao, Mingzhi Dong, Yujiang Wang, Yutian Lu, Qin Lv, Robert P. Dick, Tun Lu, Ning Gu, and Li Shang. 2021. MemX: An Attention-Aware Smart Eyewear System for Personalized Moment Auto-capture. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 56 (June 2021), 23 pages. doi:10.1145/3463509

[13] Chien-Hsiung Chen and Yi-Ting Zhai. 2023. The Effects of Information Layout, Display Mode, and Gender Difference on the User Interface Design of Mobile Shopping Applications. *IEEE Access* 11 (2023), 47024–47039. doi:10.1109/ACCESS.2023.3274575

[14] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 2024. VideoLLM-online: Online Video Large Language Model for Streaming Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 18407–18418.

[15] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2025. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision.* 19–35.

[16] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271* (2024).

[17] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. 2024. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences* 67, 12 (2024), 220101.

[18] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766* (2024).

[19] Alibaba Cloud. 2023. Intelligent Speech Interaction for Human-Computer Interaction - Alibaba Cloud — alibabacloud.com. https://www.alibabacloud.com/product/intelligent-speech-interaction. [Accessed 10-08-2023].

[20] Wenliang Dai, Junnan Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500* 2 (2023).

[21] Weisheng Dong, Guangming Shi, Xiaocheng Hu, and Yi Ma. 2014. Nonlocal sparse and low-rank regularization for optical flow estimation. *IEEE transactions on image processing* 23, 10 (2014), 4527–4538.

[22] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2025. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision.* Springer, 75–92.

[23] Cathy Mengying Fang, Patrick Chwalek, Quincy Kuang, and Pattie Maes. 2024. WatchThis: A Wearable Point-and-Ask Interface powered by Vision-Language Models for Contextual Queries. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology.* 1–4.

[24] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision.* 6202–6211.

[25] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1933–1941.

[26] Markus Funk, Carie Cunningham, Duygu Kanver, Christopher Saikalis, and Rohan Pansare. 2020. Usable and acceptable response delays of conversational agents in automotive user interfaces. In *12th international conference on automotive user interfaces and interactive vehicular applications.* 262–269.

[27] Sourabh Vasant Gothe, Vibhav Agarwal, Sourav Ghosh, Jayesh Rajkumar Vachhani, Pranay Kashyap, and Barath Raj Kandur Raja. 2024. What's in the Flow? Exploiting Temporal Motion Cues for Unsupervised Generic Event Boundary Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 6941–6950.

[28] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18995–19012.

[29] Yu Hao, Alexey Magay, Hao Huang, Shuaihang Yuan, Congcong Wen, and Yi Fang. 2024. ChatMap: A Wearable Platform Based on the Multi-modal Foundation Model to Augment Spatial Cognition for People with Blindness and Low Vision. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 129–134.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[31] Musashi Hinck, Matthew L Olson, David Cobbley, Shao-Yen Tseng, and Vasudev Lal. 2024. LLaVA-Gemma: Accelerating Multimodal Foundation Models with a Compact Language Model. *arXiv preprint arXiv:2404.01331* (2024).

[32] Roberto Hoyle, Robert Templeman, Steven Armes, Denise Anthony, David Crandall, and Apu Kapadia. 2014. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. 571–582.

[33] Akshay V Jagadeesh and Justin L Gardner. 2022. Texture-like representation of objects in human visual cortex. *Proceedings of the National Academy of Sciences* 119, 17 (2022), e2115302119.

[34] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, Kun Gai, and Yadong Mu. 2024. Video-LaVIT: Unified Video-Language Pre-training with Decoupled Visual-Motional Tokenization. In *International Conference on Machine Learning*. 22185–22209.

[35] Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Yadong Mu, et al. 2024. Unified Language-Vision Pretraining in LLM with Dynamic Discrete Visual Tokenization. In *International Conference on Learning Representations*.

[36] Marion Koelle, Swamy Ananthanarayan, Simon Czupalla, Wilko Heuten, and Susanne Boll. 2018. Your smart glasses' camera bothers me! exploring opt-in and opt-out gestures for privacy mediation. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction* (Oslo, Norway) *(NordiCHI '18)*. Association for Computing Machinery, New York, NY, USA, 473–481. doi:10.1145/3240167.3240174

[37] Robert Konrad, Nitish Padmanaban, J Gabriel Buckmaster, Kevin C Boyle, and Gordon Wetzstein. 2024. Gazegpt: Augmenting human capabilities using gaze-contingent contextual ai for smart eyewear. *arXiv preprint arXiv:2401.17217* (2024).

[38] Purushottam Kulkarni, Deepak Ganesan, Prashant Shenoy, and Qifeng Lu. 2005. SensEye: a multi-tier camera sensor network. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 229–238.

[39] Lai Chong Law, Virpi Roto, Marc Hassenzahl, Arnold P. O. S. Vermeeren, and Joke Kort. 2009. Understanding, scoping and defining user experience: a survey approach. In *Sigchi Conference on Human Factors in Computing Systems*.

[40] Hyungmin Lee, Chen-Chun Hsia, Aleksandr Tsoy, Sungmin Choi, Hanchao Hou, and Shiguang Ni. 2023. VisionARy: Exploratory research on Contextual Language Learning using AR glasses with ChatGPT. In *Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter*. 1–6.

[41] James R. Lewis. 2018. The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.

[42] James R Lewis and Jeff Sauro. 2018. Item benchmarks for the system usability scale. *Journal of Usability studies* 13, 3 (2018).

[43] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326* (2024).

[44] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[45] Robert LiKamWa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. 2013. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. 69–82.

[46] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. 2024. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947* (2024).

[47] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. 2022. Egocentric video-language pretraining. *Advances in Neural Information Processing Systems* 35 (2022), 7575–7586.

[48] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26296–26306.

[49] Ekdeep Singh Lubana, Vinayak Aggarwal, and Robert P Dick. 2019. Machine Foveation: An application-aware compressive sensing framework. In *2019 Data Compression Conference (DCC)*. IEEE, 478–487.

[50] Ekdeep Singh Lubana and Robert P Dick. 2018. Digital foveation: An energy-aware machine vision framework. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 11 (2018), 2371–2380.

[51] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424* (2023).

[52] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems* 36 (2023), 46212–46244.

[53] Martin, A., Fischler, Robert, C., and Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* (1981).

[54] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. 2024. MoReVQA: Exploring Modular Reasoning Models for Video Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13235–13245.

[55] Vivian Genaro Motti and Kelly Caine. 2015. Users' privacy concerns about wearables: impact of form factor, sensors and type of data collected. In *Financial Cryptography And Data Security: FC 2015 International Workshops, BITCOIN, WAHC, And Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*. Springer, 231–244.

[56] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[57] Rui Qian, Yeqing Li, Zheng Xu, Ming-Hsuan Yang, Serge Belongie, and Yin Cui. 2022. Multimodal open-vocabulary video classification via pre-trained vision and language models. *arXiv preprint arXiv:2207.07646* (2022).

[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[59] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*. PMLR, 28492–28518.

[60] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388* (2024).

[61] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. 2020. Intra-and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 730–739.

[62] Mike Zheng Shou, Stan Weixian Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. 2021. Generic event boundary detection: A benchmark for event segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 8075–8084.

[63] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems* 27 (2014).

[64] Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*. Springer, 402–419.

[65] Ethan Waisberg, Joshua Ong, Mouayad Masalkhi, Nasif Zaman, Prithul Sarker, Andrew G Lee, and Alireza Tavakkoli. 2024. Meta smart glasses—large language models and the future for assistive glasses for individuals with vision impairments. *Eye* 38, 6 (2024), 1036–1038.

[66] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. Springer, 20–36.

[67] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079* (2023).

[68] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2025. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*. Springer, 58–76.

[69] Zihan Wang, Qun Hao, Fanghua Zhang, Yao Hu, and Jie Cao. 2018. A variable resolution feedback improving the performances of object detection and recognition. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 232, 4 (2018), 417–427.

[70] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2024. VideoTree: Adaptive Tree-based Video Representation for LLM Reasoning on Long Videos. *arXiv preprint arXiv:2405.19209* (2024).

[71] Rongchang Xie, Chen Du, Ping Song, and Chang Liu. 2024. MUSE-VL: Modeling Unified VLM through Semantic Discrete Encoding. *arXiv preprint arXiv:2411.17762* (2024).

[72] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994* (2024).

[73] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. 2024. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841* (2024).

[74] Zhenyu Xu, Hailin Xu, Zhouyang Lu, Yingying Zhao, Rui Zhu, Yujiang Wang, Mingzhi Dong, Yuhu Chang, Qin Lv, Robert P. Dick, Fan Yang, Tun Lu, Ning Gu, and Li Shang. 2024. Can Large Language Models Be Good Companions? An LLM-Based Eyewear System with Conversational Common Ground. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 2, Article 87 (May 2024), 41 pages. doi:10.1145/3659600

[75] Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. 2024. DeCo: Decoupling Token Compression from Semantic Abstraction in Multimodal Large Language Models. *arXiv preprint arXiv:2405.20985* (2024).

[76] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178* (2023).

[77] Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2024. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. *arXiv preprint arXiv:2409.10197* (2024).

[78] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549* (2023).

[79] Gaotong Yu, Yi Chen, and Jian Xu. 2024. Balancing Performance and Efficiency: A Multimodal Large Language Model Pruning Method based Image Text Interaction. *arXiv preprint arXiv:2409.01162* (2024).

[80] Zhengqing Yuan, Zhaoxu Li, and Lichao Sun. 2023. Tinygpt-v: Efficient multimodal large language model via small backbones. *arXiv preprint arXiv:2312.16862* (2023).

[81] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11975–11986.

[82] Dell Zhang, Yongxiang Li, Zhongjiang He, and Xuelong Li. 2024. Empowering Smart Glasses with Large Language Models: Towards Ubiquitous AGI. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 631–633.

[83] Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2024. [CLS] Attention is All You Need for Training-Free Visual Token Pruning: Make VLM Inference Faster. *arXiv preprint arXiv:2412.01818* (2024).

[84] Yingying Zhao, Yuhu Chang, Yutian Lu, Yujiang Wang, Mingzhi Dong, Qin Lv, Robert P Dick, Fan Yang, Tun Lu, Ning Gu, et al. 2022. Do smart glasses dream of sentimental visions? Deep emotionship analysis for eyewear devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–29.

[85] Yingying Zhao, Mingzhi Dong, Yujiang Wang, Da Feng, Qin Lv, Robert P Dick, Dongsheng Li, Tun Lu, Ning Gu, and Li Shang. 2021. A reinforcement-learning-based energy-efficient framework for multi-task video analytics pipeline. *IEEE Transactions on Multimedia* 24 (2021), 2150–2163.

[86] Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. 2023. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581* (2023).

[87] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).

[88] Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. 2024. LLaVA-phi: Efficient Multi-Modal Assistant with Small Language Model. *arXiv preprint arXiv:2401.02330* (2024).

Appendix

A1 Task-Level Power Consumption Fluctuations: A Case Study

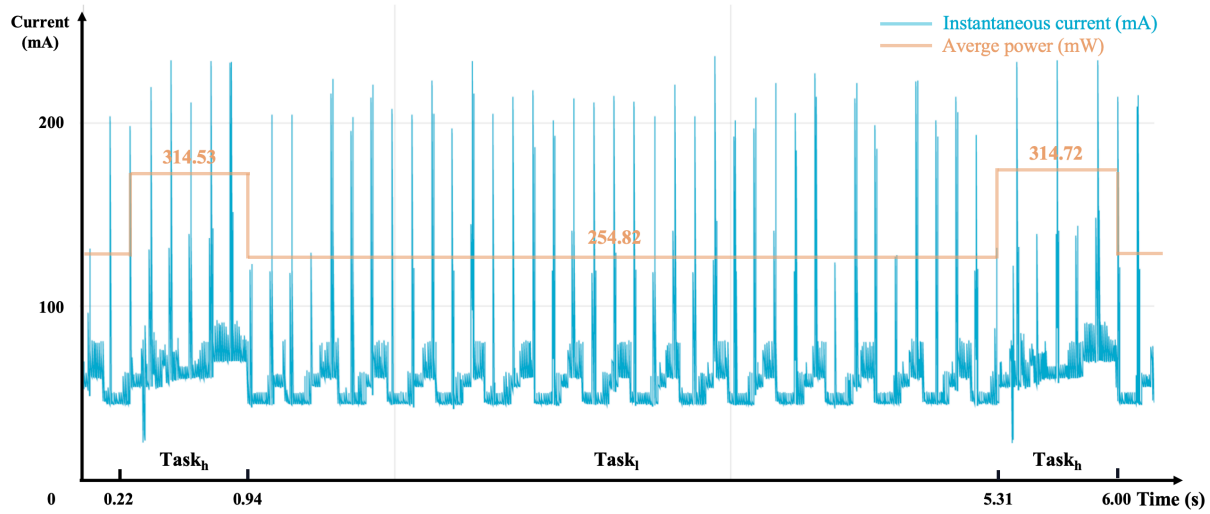

Fig. A1. A case study of active power usage trace.

As shown in Figure A1, we visualize a 6.42-second energy trace during a working-at-desk scenario. Task$_h$ is triggered twice: from 0.22–0.94 seconds and 5.31–6.00 seconds, accounting for 22.3% of the total duration. During these interval, Task$_h$ incurs an average power of 314.53 mW and 314.72 mW, respectively. In the remaining 4.99 seconds (77.7% of the duration), Task$_l$ operates at a lower average power of 254.82 mW. From waveform decomposition, we obtain the following detailed breakdowns:

- Task$_h$: sensing: 63.86 mW; compression: 29.56 mW; transmission: 27.05 mW.
- Task$_l$: sensing: 15.89 mW; computation: 10.40 mW; transmission: 24.04 mW.

These results validate that Task$_l$ consistently consumes significantly less power than Task$_h$ across all phases (sensing, computation, and transmission). It is worth noting that, in the EgoSchema-based experiments, Task$_l$ dominates the runtime (approximately 87.04%), while Task$_h$ accounts for the remaining 12.96%. This task alternation reflects the system's design focus on primarily running the energy-efficient Task$_l$, with the more power-intensive Task$_h$ invoked only when necessary. This design ensures that the system operates efficiently, with the majority of the time spent in the low-power mode.

A2 Case Studies: Perceptual Support and Companionship

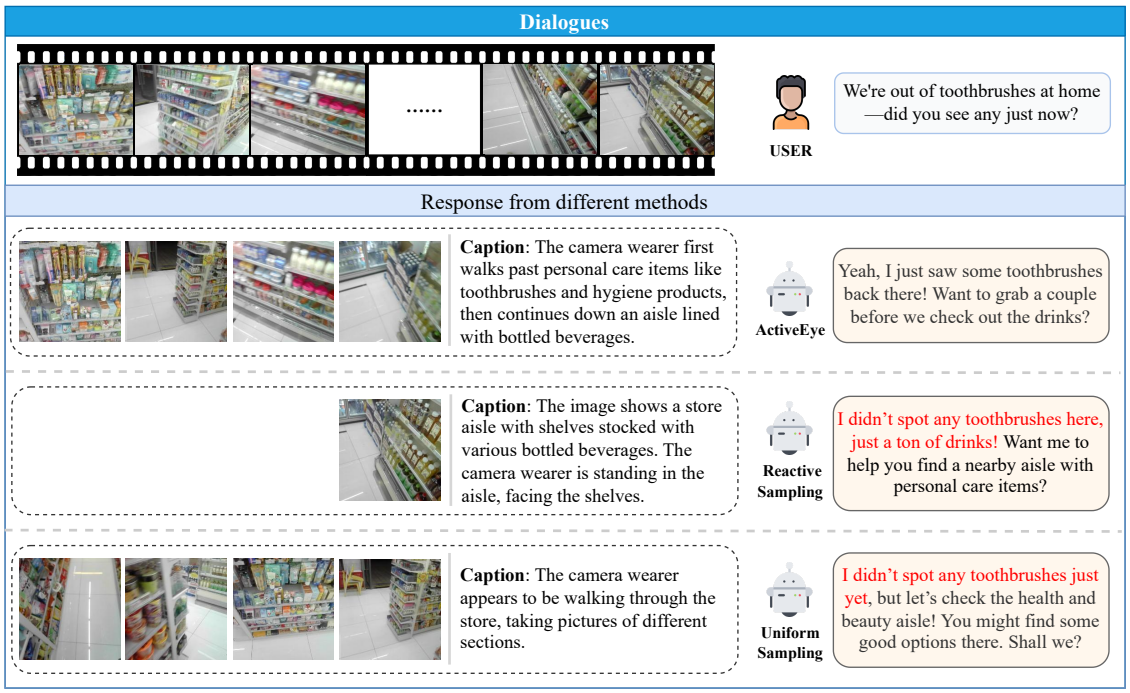

Fig. A2. Case 1: *ActiveEye* identifies a toothbrush shelf while the participant turns their gaze, enabling relevant guidance. Baselines either lacked sufficient visual input (Reactive Sampling) or missed the timing of the visual context (Uniform Sampling).

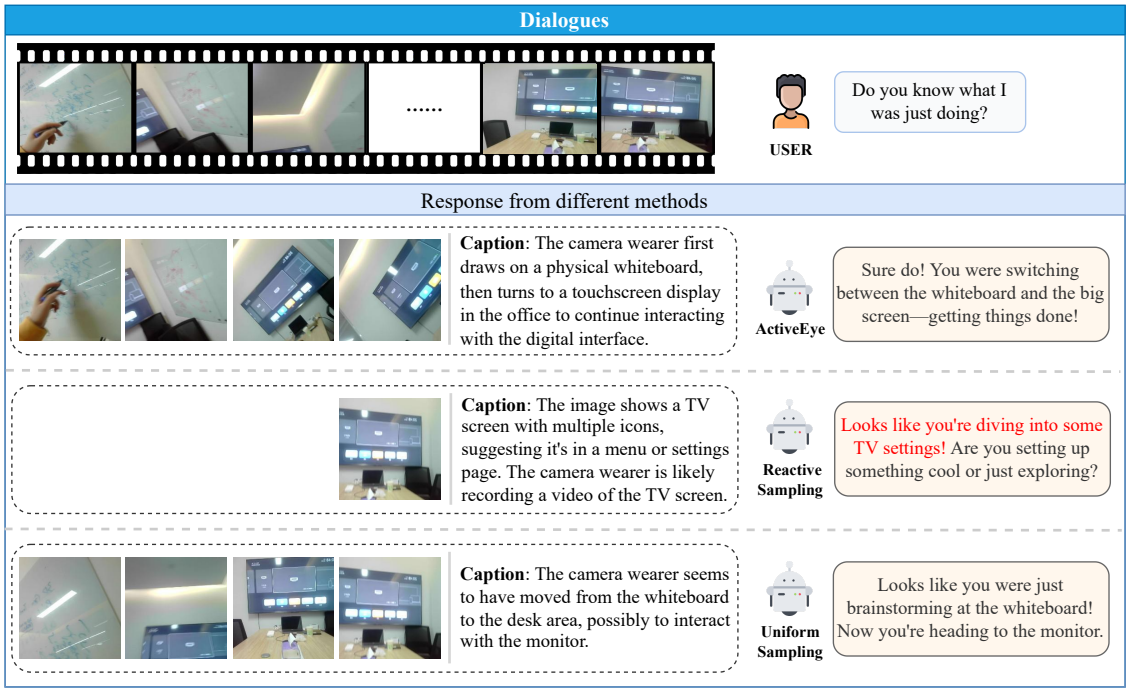

Fig. A3. Case 2: *ActiveEye* interprets user behavior even in the absence of explicit dialogue, enhancing its role as a proactive and attentive companion.