scientific reports

OPEN

I-MPN: inductive message passing network for efficient human-in-the-loop annotation of mobile eye tracking data

Hoang H. Le[1,2,3,7]✉, Duy M. H. Nguyen[1,4,5,7]✉, Omair Shahzad Bhatti[1], László Kopácsi[1], Thinh P. Ngo[2], Binh T. Nguyen[2], Michael Barz[1,6] & Daniel Sonntag[1,6]

Comprehending how humans process visual information in dynamic settings is crucial for psychology and designing user-centered interactions. While mobile eye-tracking systems combining egocentric video and gaze signals can offer valuable insights, manual analysis of these recordings is time-intensive. In this work, we present a novel *human-centered learning algorithm* designed for automated object recognition within mobile eye-tracking settings. Our approach seamlessly integrates an object detector with a spatial relation-aware inductive message-passing network (I-MPN), harnessing node profile information and capturing object correlations. Such mechanisms enable us to learn embedding functions capable of generalizing to new object angle views, facilitating rapid adaptation and efficient reasoning in dynamic contexts as users navigate their environment. Through experiments conducted on three distinct video sequences, our *interactive-based method* showcases significant performance improvements over fixed training/testing algorithms, even when trained on considerably smaller annotated samples collected through user feedback. Furthermore, we demonstrate exceptional efficiency in data annotation processes and surpass prior interactive methods that use complete object detectors, combine detectors with convolutional networks, or employ interactive video segmentation.

Keywords Human-centered AI, Scene Recognition

The advent of mobile eye-tracking technology has significantly expanded the horizons of research in fields such as psychology, marketing, and user interface design by providing a granular view of user visual attention in naturalistic settings[1,2]. By capturing intricate details of eye movement, this technology provides real-time insights into cognitive processes and user behavior during interactions with physical products or mobile devices. For instance, in educational research, mobile eye-tracking enables the exploration of learners' gaze behavior in interactive, real-world environments like classrooms and science laboratories[3,4]. Insights into where students focus their visual attention, therefore, can guide the design of instructional strategies and foster improved learning outcomes[5]. In this study, we investigate a new approach aimed at enhancing object recognition under *interactive mobile eye-tracking*, specifically optimizing data annotation efficiency and advancing human-in-the-loop learning models (Fig. 2). Equipped with eye-tracking devices, users generate video streams alongside fixation points, providing visual focus as they navigate through their environment. Our primary aim lies in recognizing specific objects, such as tablet-left, tablet-right, book, device-left, and device-right, with all other elements considered background, as demonstrated in Fig. 1.

However, the manual analysis of these eye-tracking data is challenging due to the extensive volume of data generated and the complexity of dynamic visual environments, where target objects may overlap and be affected by environmental noise[6,7]. In clinical, real-world research contexts or educational, the variability of gaze patterns across participants further complicates the extraction of meaningful insights. Additionally, the dynamic nature

[1]Interactive Machine Learning Department, German Research Center for Artificial Intelligence (DFKI), 66123 Saarbrücken, Germany. [2]Mathematics and Computer Science Department, University of Science, VNU-HCM, Ho Chi Minh City, Vietnam. [3]Quy Nhon AI Research and Development Center, FPT Software, Quy Nhon, Vietnam. [4]Max Planck Research School for Intelligent Systems (IMPRS-IS), 70569 Stuttgart, Germany. [5]Machine Learning and Simulation Science Department, University of Stuttgart, 70569 Stuttgart, Germany. [6]Applied Artificial Intelligence Department, University of Oldenburg, 26129 Oldenburg, Germany. [7]Hoang H. Le and Duy M. H. Nguyen: These authors contributed equally to this work. ✉email: lehuyhoang08032001@gmail.com; ho_minh_duy.nguyen@dfki.de

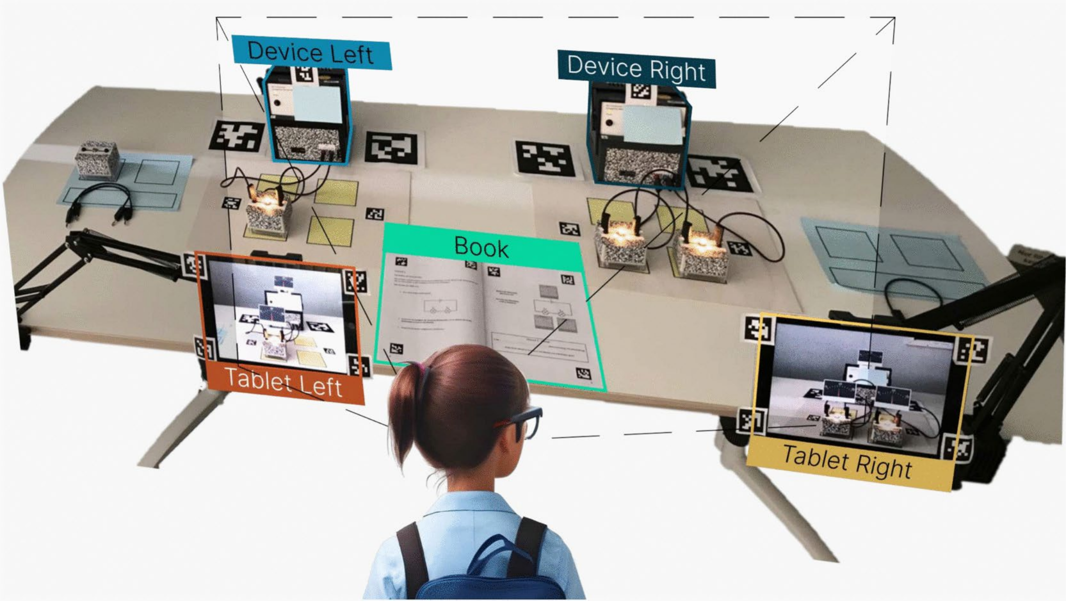


Fig. 1. Our setup for dataset collection is designed in a laboratory environment, featuring various Areas of Interest (AOIs), such as tablets and experiment stations for electrical circuits. A child equipped with a mobile eye tracker interacts with these stations, allowing for the collection of gaze data essential for research in educational contexts. The study aims to analyze children's learning processes under both AR-supported and non-AR conditions. To achieve this, our algorithm processes the input data from the eye tracker in a backend service, predicting the real-time objects the user is focusing on and tracking their attention over time.

of visual scenes demands precise object identification and segmentation, often requiring extensive manual annotations to account for factors such as occlusion, shifting lighting conditions, and rapid scene changes. These challenges highlight the pressing need for autonomous analytical strategies that can leverage advanced computational techniques to streamline data processing, improve accuracy, and reduce the burden of human intervention. Among these strategies, techniques such as gaze-based clustering[8,9], fixation heatmaps[10,11], and predictive modeling[12] are particularly promising, as they can not only enhance data interpretation but also facilitate real-time applications, such as adaptive learning systems or assistive technologies for individuals with visual or cognitive impairments.

The algorithms beyond those methods are largely powered by machine learning, with state-of-the-art architectures such as convolutional neural networks (CNNs)[13,14] and recurrent neural networks (RNNs)[15,16] achieving remarkable accuracy in predicting gaze trajectories and identifying areas of interest (AOIs) across both static and dynamic environments[17,18]. Building on these successes, object detection models, particularly those employing multi-scale feature extraction techniques like YOLO[19] and Faster R-CNN[20], have further enhanced the efficiency of visual attention detection in complex scenes[21]. Other important directions involve graph neural networks (GNNs)[22–24], which utilize graph structures to capture and model the spatial and semantic relationships among objects or regions in images, enabling robust object recognition in dynamic environments. Overall, by automating traditionally manual and time-intensive tasks, these models provide a scalable and robust approach to analyzing eye-tracking data, unlocking broader applications in dynamic and visually intricate environments.

Nevertheless, these methods face notable challenges, many of which arise from the inherent variability in human eye movement patterns and contextual dependencies[25,26]. For example, gaze behavior is highly dynamic, varying across users, tasks, and environmental factors such as occlusions and changes in lighting conditions. This complex interplay of factors often compromises model robustness, particularly in real-world scenarios where such variability is prevalent. To overcome these obstacles, large-scale training datasets are essential for ensuring effective generalization; however, the process of acquiring such datasets is both labor-intensive and time-consuming, posing an additional hurdle for advancing these methods. Moreover, integrating user feedback with individual preferences and situational contexts into machine learning workflows remains a significant bottleneck[27]. These personalized adaptations are crucial for improving the usability and accuracy of mobile eye-tracking systems, yet they often conflict with the need for computational efficiency and real-time responsiveness. Bridging this gap thus requires innovative approaches that balance adaptability with resource constraints, paving the way for models that can seamlessly customize to individual differences while remaining practical for real-world deployment.

In this work, we design a new method for interactive mobile eye-tracking as demonstrated in Fig. 2. The training process starts with initial data annotations by leveraging video object segmentation (VoS) techniques[28,29]. Users are prompted to provide weak scribbles denoting areas of interest and assign corresponding labels in initial frames. Subsequently, the VoS tool autonomously extrapolates segmentation boundaries closest to the scribbled regions, thereby generating predictions for later frames. During a period of time, users interact with the interface, reviewing and refining results by manipulating scribbles or area-of-effect (AoE) labels if they reveal error annotations.

In the next phase, we collect segmentation masks and correspondence annotations provided by the VoS tool to define bounding boxes encompassing AoI and their corresponding labels to train recognition algorithms. Our approach, named I-MPN, consists of two primary components: (i) an object detector tasked with generating proposal candidates within environmental setups and (ii) an Inductive Message-Passing Network[30–32] designed to discern object relationships and spatial configurations, thereby determining the labels of objects present in the current frame based on their correlations. It is crucial to highlight that identical objects may bear different labels contingent upon their spatial orientations (e.g., left, right) in our settings (Fig. 1, device left and right). This characteristic often poses challenges for methods reliant on local feature discrimination, such as object detection or convolutional neural networks, due to their inherent lack of global spatial context. I-MPN, instead, can overcome this issue by dynamically formulating graph structures at different frames whose node features are represented by bounding box coordinates and semantic feature representations inside detected boxes derived from the object detector. Nodes then exchange information with their local neighborhoods through a set of trainable aggregator functions, which remain invariant to input permutations and are adaptable to unseen nodes in subsequent frames. Through this mechanism, I-MPN plausibly captures the intricate relationships between objects, thus augmenting its representational capacity to dynamic environmental shifts induced by user movement.

Given the initial trained models, we integrate them into a human-in-the-loop phase to predict outcomes for each frame in a video. If users identify erroneous predictions, they have the ability to refine the models by providing feedback through drawing scribbles on the current frame using VoS tools, as shown in Fig. 3. This feedback triggers the generation of updated annotations for subsequent frames, facilitating a rapid refinement process similar to the initial annotation stage but with a reduced timeframe. The new annotations are then gathered and used to retrain both the object detector and message-passing network in the backend before being deployed for continued inference. If errors persist, the iterative process continues until the models converge to produce satisfactory results. We illustrate such an iterative loop in Fig. 2.

In summary, we make the following contributions:

- Firstly, we introduce I-MPN, an efficient object recognition framework specifically designed to integrate with mobile eye-tracking systems for analyzing gaze behavior in dynamic environments.
- Secondly, I-MPN demonstrates exceptional adaptability to user feedback within mobile eye-tracking applications. By leveraging only a small fraction of user feedback data (20%-30%), it achieves performance levels comparable to or surpassing conventional methods that rely on fixed data splits (e.g., 70% training data).

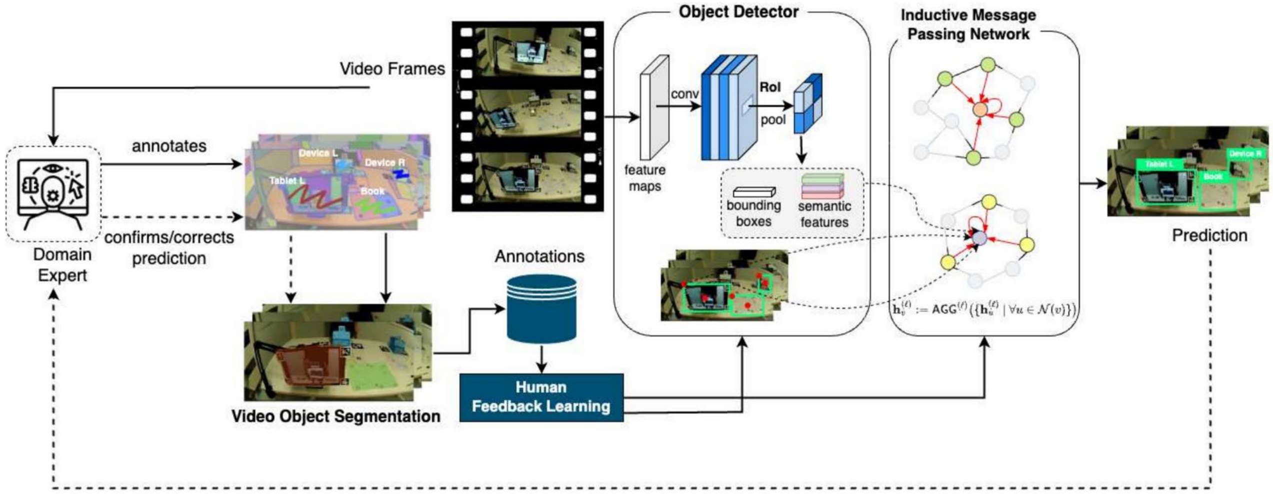

Video Frames
annotates
confirms/corrects prediction
Domain Expert
Device L
Device R
Tablet L
Book
Video Object Segmentation
Annotations
Human Feedback Learning
Object Detector
feature maps
conv
RoI pool
bounding boxes
semantic features
Inductive Message Passing Network
$\mathbf{h}_v^{(\ell)} := \mathsf{AGG}^{(\ell)}(\{\mathbf{h}_u^{(\ell)} \mid \forall u \in \mathcal{N}(v)\})$
Prediction

Fig. 2. Overview of our human-in-the-loop I-MPN approach. Video frames are processed by an object detector to produce feature maps, bounding boxes, and semantic features, which are then analyzed by the Inductive Message Passing Network for object prediction. A domain expert annotates initial frames, and the video object segmentation module propagates these annotations, reducing manual effort. Users confirm or correct predictions, engaging in a feedback loop (dashed arrow) that updates the model iteratively until it achieves a predefined level of accuracy or reliability performance assessed by the domain expert.

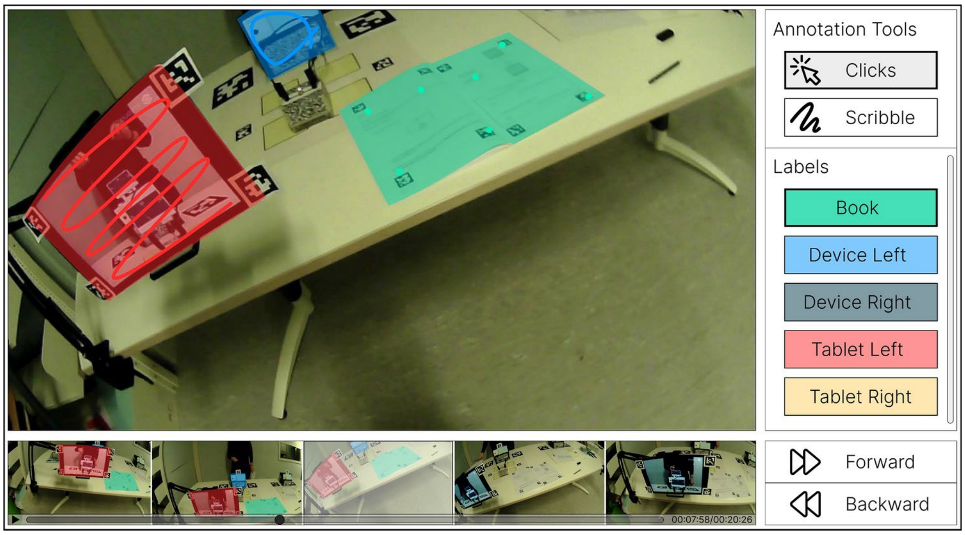


Fig. 3. The video object segmentation-based interface allows users to annotate frames using weak prompts like clicks and scribbles, then propagate these annotations to subsequent frames.

- Thirdly, a comparative analysis with other human-learning approaches, such as object detectors and interactive segmentation methods, highlights the superior performance of I-MPN, especially in dynamic environments influenced by user movement. This underscores I-MPN's capability to comprehend object relationships in challenging conditions.
- Finally, we measure the average user engagement time needed for initial model training data provision and subsequent feedback updates. Through empirical evaluation of popular annotation tools in segmentation and object classification, we demonstrate I-MPN's time efficiency, reducing label generation time by $60\% - 70\%$. We also investigate factors influencing performance, such as message-passing models. Our findings confirm the adaptability of the proposed framework across diverse network architectures.

We outline the structure of the paper with related work presented in Section "Related work", followed by a detailed description of our methodology in Section "Methodology", experimental results in Section "Experiments & results", and, finally, the conclusion and future directions in Section "Conclusion and future work".

Related work

Eye tracking-related machine learning models

Many methods rely on pre-trained models to analyze *localized features* around fixation points. For instance, some map fixations to bounding boxes using object detection models[33,34], while others classify small image patches around fixation points with image classification models[21,35]. These approaches, however, are typically limited to controlled settings where the training data closely matches the target domain[21,36]. Studies highlight substantial discrepancies between manual and automatic annotations for areas of interest in benchmark datasets like COCO[37,38], emphasizing the challenges in applying pre-trained models to diverse real-world scenarios[34]. Some efforts to fine-tune object detection models for specific domains show promise[39,40], yet these lack interactivity and cannot dynamically update during annotation.

Global interaction-based methods focus on capturing and utilizing broader contextual information. Traditional semi-automatic annotation strategies often rely on non-learnable feature descriptors like color histograms or bag-of-SIFT features[41,42], limiting adaptability. More recently, Kurzhals et al.[43] proposed an interactive approach for annotating egocentric eye-tracking data by iteratively searching time sequences based on eye movements and visual features. Their method involves segmenting gaze-focused image patches, clustering them into representative thumbnails, and visualizing these clusters on a 2D plane. While innovative, such methods primarily operate on pre-segmented patches and lack the dynamic modeling capabilities needed for more complex, adaptive environments. Unlike these works, our I-MPN is designed to capture both detailed visual feature representations and broader relational interactions among objects through an inductive message-passing network, enhancing model robustness under occlusion or significant viewpoint changes.

Graph neural networks for object recognition

Graph neural networks (GNNs) are neural models designed for analyzing graph-structured data like social networks, biological networks, and knowledge graphs[44]. Beyond these domains, GNNs can be applied in object recognition to identify and locate objects in images or videos by leveraging graph structures to encode spatial and semantic relations among objects or regions. Through mechanisms like graph convolution[45] or attention mechanisms[22], GNNs efficiently aggregate and propagate information across the graph. Methods such as GCN[46], GAT [22,47], KGN[23], SGRN[48], and RGRN[24] demonstrate the ability of GNNs to incorporate contextual reasoning, spatial relationships, and real-time processing into object recognition workflows.

Other approaches extend GNNs to address specific challenges in spatial reasoning and dynamic scenarios. Relation Networks[49] and Scene Graph Generation[50] explicitly model object relationships and generate structured scene representations. Hierarchical GNNs, like HGRN[51], integrate low-level visual features with high-level semantics for improved interaction modeling. Additionally, dynamic frameworks such as Dynamic Graph Neural Networks (DyGNN)[52], Principal Neighbourhood Aggregation (G-PNA) [53], Gated Graph Sequence Neural Networks (GatedG) [54], and Graph Transformer (TransformerG) [55] capture both spatial and temporal dynamics for video-based object recognition.

However, in mobile eye-tracking scenarios, these methods face two significant challenges. Firstly, the message-passing mechanism typically operates on the entire graph structure, necessitating a fixed set of objects during both training and inference. This rigidity implies that the entire model must be updated to accommodate new, unseen objects that may arise later due to user interests. Secondly, certain methods, such as RGRN[24] or TransformerG[55], depend on estimating the co-occurrence of object pairs in scenes using large amounts of training data. However, in human-in-the-loop settings, where users typically provide only small annotated samples, this information is not readily available. As a result, the co-occurrence matrices between objects evolve dynamically over time as more annotations are provided by the user. I-MPN tackles these issues by performing message passing to aggregate information from neighboring nodes, enabling the model to maintain robustness to variability in the graph structure across different instances. While there exist works have exploited this idea for link predictions[30], recommendation systems[56], or video tracking[57], we the first propose a formulation for human interaction in eye-tracking setups.

Human-in-the-loop for eye tracking

Recent works on human-in-the-loop methods for mobile eye tracking have utilized CNNs for object detection and classification[35,58,59]. These methods incorporate user feedback to enhance model performance, making them more adaptive to real-world scenarios. However, they often face challenges such as high computational demands and the need for extensively annotated datasets[59]. Additionally, these models can struggle with environmental noise and varying object angles, which can reduce their accuracy[60]. In contrast, our I-MPN framework combines object detectors with inductive message-passing techniques, offering more robust performance in dynamic environments while being less resource-intensive than traditional CNN-based methods.

Methodology

Dataset

We begin by detailing our setup, including the process of dataset recording and the generation of video-ground-truth annotations used to evaluate our method. Figure 1 illustrates our experimental setup where we record three video sequences captured by different users, each occurring in two to three minutes (Table 2). The users wear an eye tracker on their forehead, which records what they observe over time while also providing fixation points, showing the user's focus points at each time frame. We are interested in detecting five objects: tables (left, right), books, and devices (left, right).

Video ground-truth annotations To generate data for model evaluation, we asked users to annotate objects in each video frame using the video object segmentation tool introduced in Section "User feedback as video object segmentation". Following the cross-entropy memory method as described in[29], we interacted with users by displaying segmentation results on a monitor. Users then labeled data and created ground truths by clicking the "Scribble" and "Adding Labels" functions for objects. Subsequently, by clicking the "Forward" button, the VoS tool automatically segmented the objects' masks in the next frames until the end of the video. If users encountered incorrectly generated annotations, they could click "Stop" to edit the results using the "Scribble" and "Adding Labels" functions again (Fig. 3). Further analysis of the VoS tool is provided in Table 2, which includes a runtime comparison against other methods based on object detection and semantic segmentation.

Overview systems

Figure 2 illustrates the main steps in our pipeline. Given a set of video frames: (i) the user generates annotations by scribbling or drawing boxes around objects of interest, which are then fed into the video object segmentation algorithm to generate segment masks over the time frames. (ii) The outputs are subsequently added to the database to train an object detector, perform spatial reasoning, and generate labels for appearing objects using inductive message-passing mechanisms. The trained models are then utilized to infer the next frames until the user interrupts upon encountering incorrect predictions. At this point, users provide feedback as in step (i) for these frames (Fig. 2 bottom dashed arrow). New annotations are then added to the database, and the models are retrained as in step (ii). This loop is repeated for several rounds until the model achieves satisfactory performance. In the following sections, we describe our efficient strategy for enabling users to quickly generate annotations for video frames (Section "User feedback as video object segmentation") and our robust machine learning models designed to quickly adapt from user feedback to recognize objects in dynamic environments (Section "Dynamic spatial-temporal object recognition").

User feedback as video object segmentation

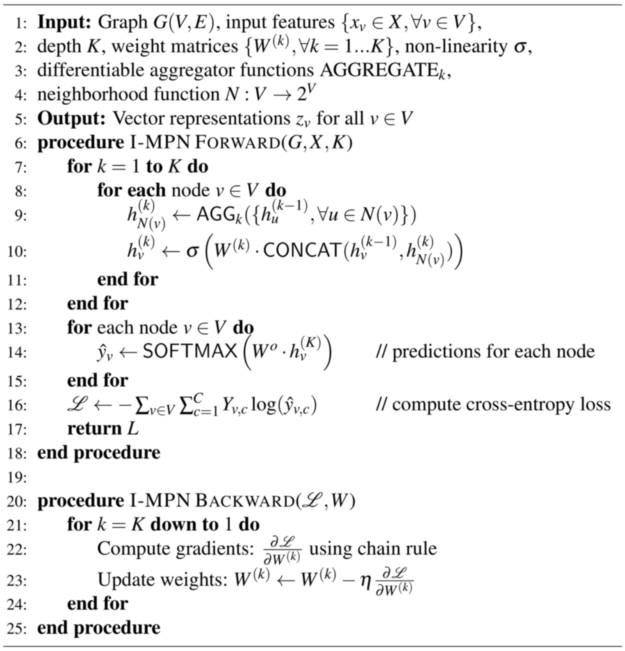
Annotating objects in video on a frame-by-frame level presents a considerable time and labor investment, particularly in lengthy videos containing numerous objects. To surmount these challenges, we utilize video object segmentation-based methods[61,62], significantly diminishing the manual workload. By using cross-video memory[29], this method achieved promising accuracy in various tasks ranging from video understanding[63], robotic manipulation[64], or neural rendering[65]. In this study, we harness this capability as an efficient tool for user interaction in annotation tasks, particularly within mobile eye-tracking, facilitating learning and model update phases. The advantages of using VoS over other prevalent annotation methods in segmentation are presented in Table 2.

Generally, with VoS, users simply mark points or scribble within the Area of Interest (AoI) along with their corresponding labels (Fig. 3). Subsequently, the VoS component infers segmentation masks for successive frames by leveraging spatial-temporal correlations (Fig. 2-left). These annotations are then subject to user verification and, if needed, adjustments, streamlining the process rather than starting from scratch each time. Formally, VoS aims to identify and segment objects across video frames ($\{F_1, F_2, \ldots, F_T\}$), producing a segmentation mask M_t for each frame F_t. In the first step, for each frame F_t, the model extracts a set of feature vectors $\mathrm{F}_t = \{f_{t1}, f_{t2}, \ldots, f_{tn}\}$, where each f_{ti} corresponds to a region proposal in the frame and n is the total number of proposals. Another *memory module* maintains a memory $\mathrm{M}_t = \{m_1, m_2, \ldots, m_k\}$ that stores aggregated feature representations of previously identified object instances, where k is the number of unique instances stored up to frame F_t. To generate correlation scores $\mathrm{C}_t = \{c_{t1}, c_{t2}, \ldots, c_{tn}\}$ among consecutive frames, a *memory reading function* $\mathrm{R}(\mathrm{F}_t, \mathrm{M}_{t-1}) \to \mathrm{C}_t$ is used. The scores in C_t estimate the likelihood of each region proposal in F_t matching an existing object instance in memory. The memory is then updated via a writing function $\mathrm{W}(\mathrm{F}_t, \mathrm{M}_{t-1}, \mathrm{C}_t) \to \mathrm{M}_t$, which modifies M_t based on the current observations and their correlations to stored instances. Finally, given the updated memory and correlation scores, the model assigns to each pixel in frame F_t a label and an instance ID, represented by $\mathrm{S}(\mathrm{F}_t, \mathrm{M}_t, \mathrm{C}_t) \to \{(l_{t1}, i_{t1}), (l_{t2}, i_{t2}), \ldots, (l_{tn}, i_{tn})\}$, where (l_{ti}, i_{ti}) indicates the class label and instance ID for the *i*-th proposal.

Dynamic spatial-temporal object recognition

Generating candidate proposals

Due to the powerful learning ability of deep convolutional neural networks, object detectors such as Faster R-CNN[66] and YOLO[19,67] offer high accuracy, end-to-end learning, adaptability to diverse scenes, scalability, and real-time performance. However, they still only propagate the visual features of the objects within the region proposal and ignore complex topologies between objects, leading to difficulties distinguishing difficult samples in complex spaces. Rather than purely using object detector outputs, we leverage their bounding boxes and corresponding semantic feature maps at each frame as candidate proposals, which are then inferred by another relational graph network. In particular, denoting f_θ as the detector, at the *i*-th frame F_i, we compute a set of *k* bonding boxes cover AoE regions by $\mathrm{B}_i = \{b_{i1}, b_{i2}, ..., b_{ik}\}$ and feature embeddings inside those ones $\mathrm{Z}_i = \{z_{i1}, z_{i2}, ..., z_{ik}\}$ while ignoring P_i denotes the set of class probabilities for each bounding boxes in B_i where $\{\mathrm{B}_i, \mathrm{Z}_i, \mathrm{P}_i\} \leftarrow \mathrm{f}_\theta(F_i)$. The f_θ is trained and updated with user feedback with annotations generated from the VoS tool.



```
Input: Graph $G(V,E)$, input features $\{x_v \in X, \forall v \in V\}$,
depth $K$, weight matrices $\{W^{(k)}, \forall k = 1...K\}$, non-linearity $\sigma$,
differentiable aggregator functions AGGREGATE$_k$,
neighborhood function $N : V \to 2^V$
Output: Vector representations $z_v$ for all $v \in V$
procedure I-MPN FORWARD$(G, X, K)$
    for $k = 1$ to $K$ do
        for each node $v \in V$ do
            $h^{(k)}_{N(v)} \leftarrow \mathsf{AGG}_k(\{h^{(k-1)}_u, \forall u \in N(v)\})$
            $h^{(k)}_v \leftarrow \sigma\left(W^{(k)} \cdot \mathsf{CONCAT}(h^{(k-1)}_v, h^{(k)}_{N(v)})\right)$
        end for
    end for
    for each node $v \in V$ do
        $\hat{y}_v \leftarrow \mathsf{SOFTMAX}\left(W^o \cdot h^{(K)}_v\right)$        // predictions for each node
    end for
    $\mathscr{L} \leftarrow -\sum_{v \in V} \sum_{c=1}^{C} Y_{v,c} \log(\hat{y}_{v,c})$        // compute cross-entropy loss
    return $L$
end procedure

procedure I-MPN BACKWARD$(\mathscr{L}, W)$
    for $k = K$ down to 1 do
        Compute gradients: $\frac{\partial \mathscr{L}}{\partial W^{(k)}}$ using chain rule
        Update weights: $W^{(k)} \leftarrow W^{(k)} - \eta \frac{\partial \mathscr{L}}{\partial W^{(k)}}$
    end for
end procedure
```

Algorithm 1. I-MPN forward and backward pass.

Inductive message passing network

We propose a graph neural network g_{ϵ} using inductive message-passing operations[30,31] for reasoning relations among objects detected within each frame in the video. Let $\mathrm{G}_i = (\mathrm{V}_i, \mathrm{E}_i)$ denote the graph at the *i*-th frame where V_i being nodes with each node $v_{ij} \leftarrow b_{ij} \in \mathrm{V}_i$ defined from bounding boxes B_i. E is the set of edges where we permit each node to be fully connected to the remaining nodes in the graph. We initialize node-feature matrix X_i, which associates for each $v_{ij} \in V_i$ a feature embedding $x_{v_{ij}}$. In our setting, we directly use $x_{v_{ij}} = z_{ij} \in Z_i$ taken from the output of the object detector. Most current GNN approaches for object recognition[24,48] use the following framework to compute feature embedding for each node in the input graph G (for the sake of simplicity, we ignore frame index):

$$\mathrm{H}^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{\mathrm{A}}\tilde{D}^{-\frac{1}{2}}\mathrm{H}^{(l)}\mathrm{W}^{(l)}) \tag{1}$$

where: $\mathrm{H}^{(l)}$ represents all node features at layer , $\tilde{\mathrm{A}}$ is the adjacency matrix of the graph G with added self-connections, $\tilde{D}$ is the degree matrix of $\tilde{\mathrm{A}}$, $\mathrm{W}^{(l)}$ is the learnable weight matrix at layer *l*, σ is the activation function, $\mathrm{H}^{(l+1)}$ is the output node features at layer $l+1$. To integrate prior knowledge, Zhao, Jianjun, et al.[24] further counted co-occurrence between objects as the adjacency matrix $\tilde{\mathrm{A}}$. However, because the adjacency matrix $\tilde{\mathrm{A}}$ is fixed during the training, *the message passing operation in* Eq (1) *cannot generate predictions for new nodes that were not part of the training data appear during inference*, i.e., the set of objects in the training and inference has to be identical. This obstacle makes the model unsuitable for the mobile eye-tracking setting, where users' areas of interest may vary over time. We address such problems by changing the way node features are updated, from being dependent on the entire graph structure $\tilde{\mathrm{A}}$ to neighboring nodes $\mathcal{N}(v)$ for each node *v*. In particular,

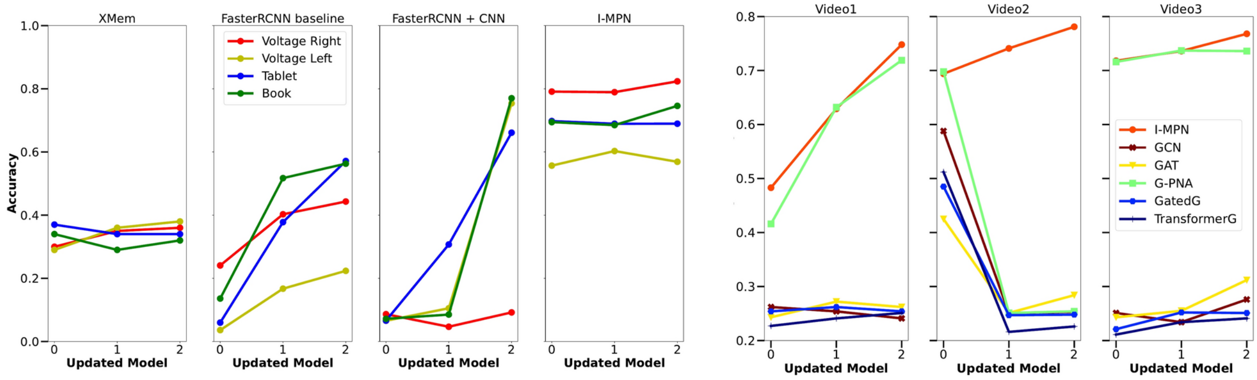
$$\mathrm{h}_{\mathcal{N}(v)}^{(l)} = \mathrm{AGG}^{(\ell)}(\{\mathrm{h}_u^{(l)}, \forall u \in \mathcal{N}(v)\}) \tag{2}$$

$$\mathrm{h}_v^{(l+1)} = \sigma\Big(\mathrm{W}^{(l)} \cdot \mathrm{CONCAT}\big(\mathrm{h}_v^{(l)}, \mathrm{h}_{\mathcal{N}(v)}^{(l)}\big)\Big) \tag{3}$$

where: $\mathrm{h}_v^{(l)}$ represents the feature vector of node *v* at layer *l*, AGG is an aggregation function (e.g., Pooling, LSTM), CONCAT be the concatenation operation, $\mathrm{h}_v^{(l+1)}$ is the updated feature vector of node *v* at layer $l+1$. In scenarios when a new unseen object v_{new} is added to track by the user, we can aggregate information from neighboring seen nodes $v_{seen} \in \mathcal{N}(v_{new})$ by:

$$\mathrm{h}_{v_{new}}^{(l+1)} = \sigma\Big(\mathrm{W}^{(l)} \cdot \mathrm{CONCAT}\big(\mathrm{h}_{v_{new}}^{(l)}, \mathrm{AGG}^{(\ell)}(\{\mathrm{h}_{v_{seen}}^{(l)}\})\tag{4}$$

and then update the trained model on this new sample rather than all nodes in training data as Eq.(1). The forward and backward pass of our message-passing algorithm is summarized in the Algorithm 1. We found that such operations obtained better results in experiments than other message-passing methods such as attention network[22], principled aggregation[53] or transformer[68] (Fig. 4b).



(a) Performance comparison between various human-in-the-loop baselines after each updated time across three video sequences. Results are measured for all objects using the average balanced accuracy metric.

(b) Our I-MPN method uses inductive graph performance compared to other GNNs. Performance is computed for all objects in the 30% test set using average accuracy.

Fig. 4. Comparative performance analysis.

```
    # f_theta: object detector
    # g_epsilon: inductive message passing network
    # max_update: maximum number of taking user feedback
    # VoS: video object segmentation model
    # t_initial: time for initial annotation step
    # t_update: time for updating with user feedback
    # F = [F_1, ..., F_t]: list of frames in video

## Stage 1. Training initial models
    # extract initial annotations by user (Alg. 3)
    D_init = interactive_func(F[0:t_initial], VoS)
    # train object detector and relational graph network
    f_theta.train(D_init); g_epsilon.train(D_init)

## Stage 2. Inference and User Feedback Update
    update_time = 0
    frame_index = t_initial
    while frame_index <= len(F) + 1:
       # generate object candidates by the detector
       candidate_objects, feature_maps = f_theta(F[frame_index])

       # build graph and inference labels
       G = construct_graph(candidate_objects, feature_maps)
       detected_objects, labels = g_epsilon(G)

       # show outputs to user
       display(detected_objects, labels)

       # user feedback if encountering wrong outputs
       if (update_time <= max_update) and (user.satisfy(detected_objects, label) is False):
           start_index = frame_index
           end_index = start_index + t_update + 1

           # using Alg. 3
           D_feedback = interactive_func(F[start_index, end_index], VoS)

           # updated model with user feedback
           f_theta.train(D_feedback);
           g_epsilon.train(D_feedback)

           # update counting numbers
           update_time += 1
           frame_index = end_index
       else:
           frame_index += 1
```

Algorithm 2. PyTorch-style I-MLE algorithm.

End-to-end learning from human feedback

In Algorithm 2, we present the proposed human-in-the-loop method for mobile eye-tracking object recognition. This approach integrates user feedback to jointly train the object detector f_θ and the graph neural network g_ϵ for spatial reasoning of object positions. Specifically, f_θ is trained to generate coordinates for proposal object bounding boxes, which are then used as inputs for g_ϵ (bounding box coordinates and feature embedding inside those regions). The graph neural network g_ϵ is, on the other hand, trained to generate labels for these objects by considering the correlations among them. Notably, our pipeline operates as an end-to-end framework, optimizing both the object detector and the graph neural network simultaneously rather than as separate components. This lessens the propagation of errors from the object detector to the GNN component, making the system be robust to noises in environment setups. The trained models are deployed afterward to infer the next frames and are then refined again at wrong predictions, giving user annotation feedback in a few loops till the model converges. In the experiment results, we found that such a human-in-the-loop scheme enhances the algorithm's adaptation ability and yields comparable or superior results to traditional learning methods with a set number of training and testing samples.

```
## User feedback functions
def interactive_func(list_frame, VoS):
   D = [] # store annotation data

   # generate initial segment masks
   init_mask = VoS(list_frames[0])
   display(init_mask)

   # user correct with scribbles
   ann_mask, label = user.annotate(init_mask)

   # propagate predictions for next frames
   for frame in sorted(list_frames[1:]):
       next_mask, label = VoS(frame, ann_mask, label)
       display(next_mask, label)

       # user update if persist errors
       if user.satisfy(updated_mask, label) is False:
            ann_mask, label = user.annotate(next_mask, label)
            D.append({ann_mask, label, frame})
        else:
            D.append({next_mask, label, frame})
    return D
```

Algorithm 3. User feedback propagation algorithm.

Experiments & results

Device Our hardware setup utilizes the Pupil Core eye-tracking device[1], which users wear to observe their surroundings during the video recording process. The device outputs both video data and fixation points, capturing the user's focal attention at each moment in time. The videos are displayed on a monitor, where a backend service powered by the VoS tool facilitates the annotation process.

Dataset statistics Our study utilizes three distinct video sequences, each recorded by a different user within our controlled environment. The first video spans 169 seconds, yielding 3873 extracted frames. The second video, slightly longer at 183 seconds, comprises 3422 frames. The third sequence, shorter in duration at 118 seconds, contains 2340 frames. This diverse range of video lengths and frame counts ensures a comprehensive dataset for evaluating the performance and adaptability of our method.

Metrics The experiment results are measured by the consistency of predicted bounding boxes and their labels with ground-truth ones. In most experiments except the fixation point cases, we evaluate performance for all objects in each video frame. We define $AP@\alpha$ as the Area Under the Precision-Recall Curve (AUC-PR) evaluated at α IoU threshold $AP@\alpha = \displaystyle\int_0^1 p\left(r\right) dr$ where $p(r)$ represents the precision at a given recall level r. The mean Average Precision[69] is computed at different α IoU ($mAP@\alpha$), which is the average of AP values over all classes, i.e., $mAP@\alpha = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(AP@\alpha)_i$. We provide results for $\alpha \in \{50, 75\}$. Furthermore, we report mAP as an average of different IoU ranging from $0.5 \rightarrow 0.95$ with a step of 0.05.

To assess the overall accuracy of object classification across the entire scene (Section "Further analysis"), we compute the average accuracy, which reflects the model's ability to correctly identify and classify objects within a given frame. Specifically, the average accuracy metric is calculated by averaging the classification accuracy for each object in all video frames. This measure is critical in tasks where not only the localization of objects but also their correct classification is important.

Model configurations We use the Faster-RCNN[66] as the network backbone for the object detector $\mathbf{f}_\theta$ and follow the same proposed training procedure by the authors. The message-passing component $\mathbf{g}_\epsilon$ uses the $\mathrm{MaxPooling}$ and LSTM aggregator functions to extract and learn embedding features for each node. We use output bounding boxes and feature embedding at the last layer in $\mathbf{f}_\theta$ as inputs for $\mathbf{g}_\epsilon$. The outputs of $\mathbf{g}_\epsilon$ are then fed into the $\mathrm{Softmax}$ and trained with cross-entropy loss using Adam optimizer[70].

Human-in-the-loop vs. conventional data splitting learning

We investigate I-MPN's abilities to interactively adapt to human feedback provided during the learning model and compare it with a conventional learning paradigm using the fixed train-test splitting rate.

Baselines setup In the *conventional machine learning* approach (CML), we employ a fixed partitioning strategy, where the first 70% of video frames, along with their corresponding labels, are utilized for training, while the remaining 30% are reserved for testing purposes. We use I-MPN to learn from these annotations. In the *human-in-the-loop* (HiL) setting, we still utilize I-MPN but with a different approach. Initially, only the first 10 seconds

[1] https://pupil-labs.com/products/core

of data are used for training. Subsequently, the model is then iteratively updated with 10 seconds of human feedback at each step, triggered whenever the user detects poor performance.

Performance evaluation in both settings is conducted under two scenarios: one where 30% of the frames from each video are used for testing, and another where the entire video is used for testing. The first scenario evaluates the model's ability to generalize to unseen samples, while the second scenario tests whether the model suffers from under-fitting by assessing its performance over the entire dataset.

Results Table 1 showcases our findings, highlighting two key observations. Firstly, I-MPN demonstrates its ability to learn from user feedback, as evidenced by the model's progressively improving performance with each update across various metrics and videos. For example, the mAP@50_w score for Video 1 significantly increases from 0.544 (at $k = 0$) to 0.822 (at $k = 2$), reflecting a 51% improvement. Similarly, Video 2 exhibits a 50% increase in performance, confirming this trend.

Secondly, human-in-the-loop (HiL) learning with I-MPN has demonstrated its ability to match or exceed the performance of conventional learning approaches with just a few updates, even when utilizing a small amount of training samples. For instance, in Videos 1 and 2, after initial training and two to three loops of feedback integration (equating to approximately $18 - 23\%$ of the total training data), HiL achieves a mAP@50_w of 0.835, while the CML counterpart achieves 0.814 (trained with 70% of the available data). We argue that such advantages come from user feedback on hard samples, enabling the model to adapt its decision boundaries to areas of ambiguity caused by similar objects or environmental conditions. Conversely, the CML approach treats all training samples equally, potentially resulting in over-fitting to simplistic cases often present in the training data and failing to explicitly learn from challenging samples.

Comparing with other interactive approaches

In our study, we aim to discriminate the positions of items in the same class, e.g., left and right devices (Fig. 1). This requires the employed model to be able to explicitly capture spatial relations among object proposals rather than just local region ones. We highlight this characteristic in I-MPN by comparing it with other human-in-the-loop algorithms.

Baselines setup (i) The first algorithm we used is the faster-RCNN, which learns from the same human user feedback as I-MPN and generates directly bounding boxes together with corresponding labels for objects in video frames. (ii) The second baseline adapts another deep convolutional neural network (CNN) on top of Faster-RCNN outputs to refine predictions using visual features inside local windows around the area of interest. (iii) Finally, we compare the VoS model used in I-MPN's user annotation collection with the X-mem method[29], but it is now used as an inference tool instead. Specifically, at each update time, X-mem re-initializes segmentation masks and labels, which are given user feedback; then, X-mem propagates these added annotations for subsequent frames.

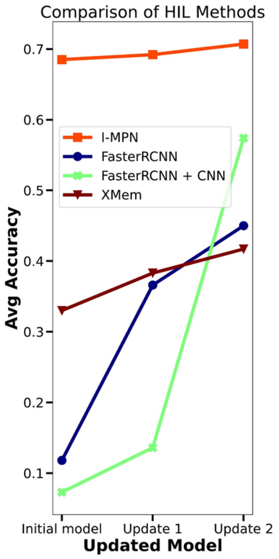
Results We report in Fig. 5b the performance of all methods in two classes, left and right devices that require spatial reasoning abilities. A balanced accuracy metric[71] is used to compute performance at video frames where

Data	Method	Feedback	%Data	Time$_w$ (s)↓	mAP ↑	mAP@50 ↑	mAP@75 ↑	Time$_t$ (s)↓	mAP$_t$↑	mAP@50$_t$↑	mAP@75$_t$↑
Video 1	CML	0	70%	401	0.66	0.814	0.771	402	0.671	0.803	0.761
	HiL	0	6%	48	0.330	0.544	0.332	32	0.300	0.504	0.307
		1	6%	46	0.600	0.799	0.693	29	0.541	0.732	0.656
		2	6%	46	**0.676**	**0.822**	**0.782**	29	0.574	0.782	0.741
		3	6%	46	**0.702**	**0.835**	**0.793**	28	**0.687**	**0.809**	**0.778**
Video 2	CML	0	70%	361	0.562	0.740	0.657	367	0.568	0.755	0.673
	HiL	0	5.8%	51	0.349	0.498	0.411	48	0.348	0.516	0.412
		1	5.8%	53	0.471	0.611	0.560	48	0.565	0.744	0.648
		2	5.8%	54	0.591	0.645	0.687	48	0.581	0.762	0.662
		3	5.8%	54	**0.622**	**0.747**	**0.683**	57	**0.622**	**0.800**	**0.683**
Video 3	CML	0	70%	143	0.758	0.962	0.878	252	0.758	0.957	0.878
	HiL	0	8.5%	47	0.558	0.829	0.656	58	0.558	0.829	0.656
		1	8.5%	45	0.625	0.901	0.713	46	0.625	0.901	0.713
		2	8.5%	48	**0.764**	**0.963**	**0.890**	57	**0.764**	**0.967**	**0.880**

Table 1. Performance comparison between conventional machine learning (**CML**) and human-in-the-loop (**HiL**) using I-MPN, evaluated on the *whole video* (**w**) and evaluated on a *fixed test set* (30%) (**t**). Feedback $= k$, where $k = 0$ indicates the initial training phase, $k > 0$ is the number of times the algorithm is updated. **Time (s)** is the training time. **Bold** and underline values mark results of **HiL**, which are higher than **CML** and represent the best performance overall. In the **HiL** setting, the model is interactively updated with user feedback whenever users identify poor performance (**Feedback**) column.

Video	Object	Initial	Update 1	Update 2
Video 1	Avg Acc	0.391	0.694	0.742
	Voltage	0.617	0.692	0.739
	Tablet	0.274	0.912	0.966
	Book	0.189	0.350	0.489
	Background	0.530	0.798	0.812
Video 2	Avg Acc	0.501	0.755	0.839
	Voltage Left	0.711	0.955	0.977
	Tablet	0.943	0.944	0.982
	Book	0.597	0.686	0.740
	Background	0.600	0.625	0.923
	Voltage Right	0.820	0.887	0.907
Video 3	Avg Acc	0.250	0.726	0.748
	Voltage	0.182	0.222	0.667
	Tablet	0.146	0.636	0.903
	Book	0.213	0.787	0.955
	Background	0.766	0.851	0.971

(a)



(b)

Fig. 5. (**a**) Eye Tracking Point Classification results are improved after upgrading the model with user feedback. Evaluation of different objects given fixation points. (**b**) Comparison between human-in-the-loop methods on classes requiring spatial object understanding. Results are on balanced accuracy. Higher is better.

one of these classes appears and average results across three video sequences. Furthermore, we present in Fig. 4b the case where all objects are measured.

It is evident that methods relying on human interaction have consistently improved their performance based on user feedback, except X-Mem, which only re-initializes labels at some time frames and uses them to propagate for the next ones. Among these, I-MPN stably achieved better performance. Furthermore, when examining classes such as left and right devices in detail, I-MPN demonstrates markedly superior performance, exhibiting a significant gap compared to alternative approaches. For instance, after two rounds of updates, we achieved an approximate accuracy of 70% with I-MPN, whereas X-mem lagged at only 41.7%. This discrepancy highlights the limitations of depending solely on local feature representations, such as those employed in Faster-RCNN or CNN, or on temporal dependencies among objects in sequential frames, like X-mem, for accurate object inference. Objects with similar appearances might have different labels based on their spatial positions. Therefore, utilizing message-passing operations, as done in I-MPN, provides a more effective method for predicting spatial object interactions.

Efficient user annotations

In this section, we demonstrate the benefits of using video object segmentation to generate video annotations from user feedback introduced in Section "User feedback as video object segmentation".

Baselines Setup (i) We first compare with the `CVAT` method[72], a tool developed by Intel and an open-source annotation tool for images and videos. CVAT offers diverse annotation options and formats, making it well-suited for many computer vision endeavors, spanning from object detection and instance segmentation to pose estimation tasks. (ii) The second software we evaluate is `Roboflow`[2], another popular platform that includes AI-assisted labeling for bounding boxes, smart polygons, and automatic segmentation.

Results Table 2 outlines the time demanded by each method to generate ground truth across all frames within three video sequences. Two distinct values are reported: (a) T_{tot}, representing the *total* time consumed by each method to produce annotations, encompassing both user-interaction phases and algorithm-supported steps; and (b) T_{eng}, indicating the time users *engage* on interactive tasks such as clicking, drawing scribbles or bounding boxes, etc. Notably, actions such as waiting for model inference on subsequent frames are excluded from these calculations.

Observed results show us that using the VoS tool is highly effective in saving annotation time compared to frame-by-frame methods. For instance, in Video 1, CVAT and Roboflow take longer 3 times than I-MPN on T_{tot}. Users also spend less time annotating with I-MPN than other ones, such as 43 seconds in Video 2 versus 1386 seconds with Roboflow. We argue that these advantages derive from the algorithm's ability to automatically

[2] https://roboflow.com/

Dataset	Time (s)	Frames	Our		CVAT		Roboflow	
			T_{tot} ↓	T_{eng} ↓	T_{tot} ↓	T_{eng} ↓	T_{tot} ↓	T_{eng} ↓
Video 1	169	3873	516	74	1638	1638	1722	1722
Video 2	183	3422	426	43	1476	1476	1386	1386
Video 3	118	2340	330	36	1032	1032	924	924

Table 2. Running time comparison of different methods to generate video annotations. T_{tot} denotes the time taken by each method to infer labels for all frames, while T_{eng} indicates the time users spend actively interacting with the tool through click-and-draw actions, excluding waiting time during mask generation. Smaller is better.

infer annotations across successive frames using short spatial-temporal correlations and its support for weak annotations like points or scribbles.

Further analysis

Inductive message passing network contribution

Each frame of the video captures a specific point of view, making the graphs based on these images dynamic. New items may appear, and some may disappear during the process of recognizing and distinguishing objects. This necessitates a spatial reasoning model that quickly adapts to unseen nodes and is robust under missing or occluded scenes. In this section, we demonstrate the advantages of the inductive message-passing network employed in I-MPN and compare it with other approaches.

Baselines setup We experiment with Graph Convolutional Network (`GCN`)[46], Graph Attention Network (`GAT`)[22,47], Principal Neighbourhood Aggregation (`G-PNA`)[53], Gated Graph Sequence Neural Networks (`GatedG`)[54], and Graph Transformer (`TransformerG`)[55]. Among these baselines, GCN and GAT employ different mechanisms to aggregate features but still depend on the entire graph structure. G-PNA, GatedG, and Transformer-G can be adapted to unseen nodes, using neighborhood correlation or treating input nodes in the graph as a sequence.

Results Figure 4b presents our observations on the averaged accuracy across all objects. We identified two key phenomena. First, methods that utilize the entire graph structure, such as GCN and GAT, struggle to update their model parameters effectively, resulting in minimal improvement or stagnation after the initial training phase. Second, approaches capable of handling arbitrary object sizes, like GatedG and transformers, also exhibit low performance. We attribute this to the necessity of large training datasets to adequately train these models. Additionally, while G-PNA shows promise as an inductive method, its performance is inconsistent across different datasets, likely due to the complex parameter tuning required for its multiple aggregation types. In summary, this ablation study highlights the superiority of our inductive mechanism, which proves to be stable and effective in adapting to new objects or changing environments, particularly in eye-tracking applications.

Fixation-point results

In eye-tracking experiments, researchers are generally more interested in identifying the specific areas of interest (AOIs) that users focus on at any given moment rather than determining the bounding boxes of all possible AOIs. Therefore, we have further examined the accuracy of our model in the fixation-to-AOI mapping task. Fortunately, this can be solved by leveraging outputs of I-MPN at each frame with bounding boxes and corresponding labels. In particular, we map the fixation point at each time frame to the bounding box and check if the fixation point intersects with the bounding box to determine if an AOI is fixated (Fig. 6). Similar to our previous experiment, we start with a 10-second annotation phase using the VoS tool after initial training. As soon as there is an incorrect prediction for fixation-to-AOI mapping, we perform an update with a 10-second correction.

Results Figure 5a presents the outcomes of the fixation-point classification accuracy following model updates based on user feedback. For Video 1, the average accuracy increased from 0.391 at the initial stage to 0.742 after the second update. The classification accuracy for tablets notably increased to 0.966, while books and background objects also exhibited improved accuracies by the second update. For Video 2, an increase in average accuracy from 0.501 to 0.839 was observed. The left voltage object's accuracy reached 0.977, and the right voltage improved to 0.907 by the second update. Tablets maintained high accuracy throughout the updates. For Video 3, the average accuracy enhanced from 0.250 to 0.748. Tablets and books showed substantial improvements, with final accuracies of 0.903 and 0.955, respectively. The background classification also improved. Overall, the results underscore the effectiveness of user feedback in refining the model's AOI classification, proving the model's adaptability and increased precision in identifying fixated AOIs within eye-tracking experiments.

Visualization results

The visualizations in Fig. 6 demonstrate the I-MPN approach's effectiveness in object detection and fixation-to-AOI mapping. Firstly, even if multiple identical objects are present in a frame, I-MPN is able to recognize and differentiate them and further reason about their spatial location. We see in Fig. 6 (bottom left) that both voltage devices are recognized and further differentiated by their spatial location. Additionally, if the objects are

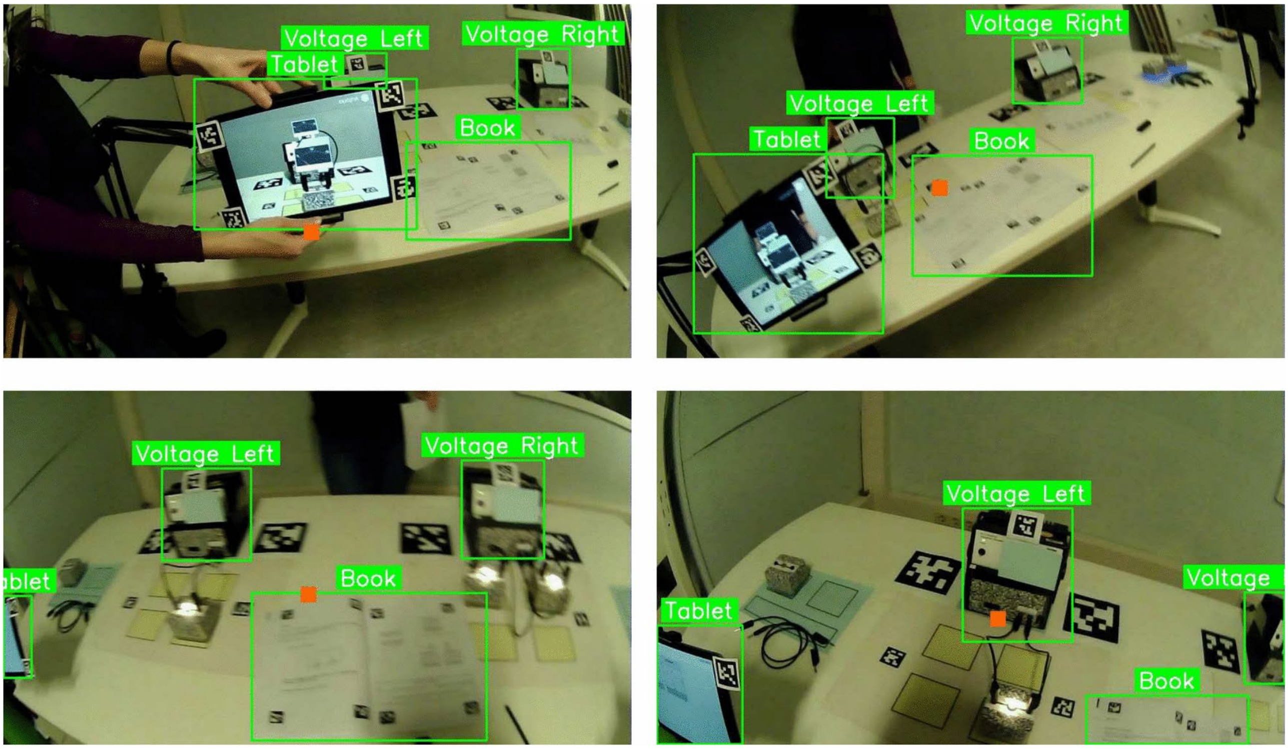


Fig. 6. Visualization results from our interactive-based model, showing fixation points (marked in red) across different video frames.

only partially in the frame or occluded by another object, I-MPN is still able to recognize the objects reliably. This is especially important in real-world conditions where the scene is very dynamic due to the movements of the person wearing the eye tracker. Lastly, traditional methods that rely only on local information around the fixation point, such as using a crop around the fixation point, can struggle with correctly detecting the fixated object. This is especially true when the fixation point is at the border of the object. This issue is evident in Fig. 6 (top/bottom left), where traditional methods fail to detect objects accurately. In contrast, our approach uses bounding box information, which allows us to reason more accurately about the fixated AOI. In summary, we argue that I-MPN provides a more comprehensive understanding of the scene, particularly in mobile eye-tracking applications where precise AOI identification is essential.

Limitations

While our proposed I-MPN framework demonstrates promising results in enhancing object recognition within interactive mobile eye-tracking applications, several limitations need to be acknowledged. Firstly, our experiments were conducted on a limited dataset of three video sequences recorded in controlled laboratory environments. This restricted setting may not fully capture the diversity of real-world scenarios, potentially affecting the generalizability of our findings. Expanding the dataset to include a wider variety of environmental conditions, user behaviors, and object types is necessary to validate the robustness of I-MPN in broader contexts. Secondly, the effectiveness of our approach depends on the initial annotations provided through the video object segmentation tool. Inaccurate or incomplete annotations can compromise the model's ability to learn effectively. Moreover, we assume perfect human feedback during the interactive annotation process, which may not reflect real-world scenarios where user input can be inconsistent or erroneous. Lastly, while our method shows improved robustness to occlusions and dynamic environmental shifts compared to traditional object detectors and convolutional neural networks, challenges may remain when dealing with severe occlusions or extremely rapid scene changes. The inductive message-passing network may not fully capture complex temporal dependencies over longer sequences or when objects abruptly enter or exit the field of view.

Conclusion and future work

In this paper, we introduce I-MPN, a machine-learning framework designed to enhance object recognition in interactive mobile eye-tracking applications. Our main objectives are to (i) develop an efficient object recognition system for dynamic environments and to (ii) optimize data annotation through human-in-the-loop learning, reducing manual effort in annotating large eye-tracking datasets.

Our experimental results show that I-MPN successfully meets these goals. By combining an object detector with a spatially aware inductive message-passing network, I-MPN improves object recognition in dynamic settings. The framework captures both local visual features and global interactions among objects, ensuring robust recognition even during occlusions or rapid shifts in the user's viewpoint. This achieves our first objective of providing an efficient object recognition system for mobile eye-tracking applications. Additionally, our approach employs a video segmentation-based annotation method that facilitates efficient user interaction and feedback. Using the VoS tools, we significantly reduce annotation time compared to traditional methods. The human-in-the-loop process, given this, allows the model to quickly adapt to user feedback with minimal labeled data, fulfilling our second objective of optimizing the data annotation process. In summary, our experiments demonstrate that I-MPN not only enhances object recognition in mobile eye-tracking scenarios but also streamlines the annotation process, confirming the effectiveness of our framework in achieving the research objectives.

While I-MPN achieved promising results on our real setups, we believe the following points are important to investigate:

- Firstly, conducting experiments on more complicated human-eye tracking, for example, with advanced driver-assistance systems (ADAS)[73,74] to improve safety by understanding the driver's focus and intentions. Such applications require state-of-the-art models, e.g., foundation models[75] trained on large-scale data, which can make robust recognition under domain shifts like day and night or different weather conditions. However, fine-tuning such a large model using a few user feedback remains a challenge[76].
- Secondly, while our simulations using the video object segmentation tool have demonstrated that I-MPN requires minimal user intervention to match or surpass the state-of-the-art performance, future research should prioritize a comprehensive human-centered design experiment. This entails a deeper investigation into how to best utilize the strengths of I-MPN and create an optimal interaction and user interface. The design should be intuitive, minimize errors by clearly highlighting interactive elements, and provide immediate feedback on user actions. These features are important to ensure that eye-tracking data is both accurate and reliable[77,78].
- Thirdly, extending I-MPN from user to multiple users has several important applications, for e.g., collaborative learning environments to understand how students engage with shared materials, helping educators to optimize group study sessions. Nonetheless, those situations pose challenges related to fairness learning[79,80], which aims to make the trained algorithm produce equitable decisions without introducing bias toward a group's behavior with several users sharing similar behaviors.
- Finally, enabling I-MPN interaction running on edge devices such as smartphones, wearables, and IoT devices is another interesting direction. This ensures that individuals with limited access to high-end technology can still benefit from the convenience and functionality offered by our systems. To tackle this challenge effectively, it is imperative to explore model compression techniques aimed at enhancing efficiency and reducing complexity without sacrificing performance[81–84].

Data availability

The datasets generated and/or analyzed during this study are not publicly available due to data privacy concerns related to the collection of information from children and the fact that publication was not an intended purpose. However, the data are available from the corresponding author on reasonable request.

Received: 7 July 2024; Accepted: 14 March 2025
Published online: 23 April 2025

References

1. Holmqvist, K. *et al. Eye tracking: A comprehensive guide to methods and measures* (OUP Oxford, 2011).
2. Duchowski, T. A. *Eye tracking: methodology theory and practice* (Springer, 2017).
3. Salminen-Saari, J. F. A. et al. Phases of collaborative mathematical problem solving and joint attention: a case study utilizing mobile gaze tracking. *ZDM - Mathematics Education* **53**, 771–784. https://doi.org/10.1007/s11858-021-01280-z (2021).
4. Fleischer, T. *et al.* Mobile Eye Tracking during Experimenting with Digital Scaffolding–Gaze Shifts between Augmented Reality and Experiment during Zinc Iodide Electrolysis Set-Up. *Education Sciences* **13**, https://doi.org/10.3390/educsci13020170 (2023).
5. Jarodzka, H., Holmqvist, K. & Gruber, H. Eye tracking in Educational Science: Theoretical frameworks and research agendas. *Journal of Eye Movement Research* **10**, https://doi.org/10.16910/jemr.10.1.3 (2017). Section: Articles.
6. Strandvall, T. Eye tracking in human-computer interaction and usability research. In *Human-Computer Interaction–INTERACT 2009: 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedings, Part II 12*, 936–937 (Springer, 2009).
7. Gardony, A. L., Lindeman, R. W. & Brunyé, T. T. Eye-tracking for human-centered mixed reality: promises and challenges. In *Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR)*, vol. 11310, 230–247 (SPIE, 2020).
8. Kao, C.-W., Chen, H.-H., Wu, S.-H., Hwang, B.-J. & Fan, K.-C. Cluster based gaze estimation and data visualization supporting diverse environments. In *Proceedings of the International Conference on Watermarking and Image Processing*, 37–41 (2017).
9. Wang, X., Zhang, J., Zhang, H., Zhao, S. & Liu, H. Vision-based gaze estimation: A review. *IEEE Transactions on Cognitive and Developmental Systems* **14**, 316–332 (2021).
10. Pfeiffer, T. & Memili, C. Model-based real-time visualization of realistic three-dimensional heat maps for mobile eye tracking and eye tracking in virtual reality. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 95–102 (2016).
11. Drusch, G., Bastien, J. & Paris, S. Analysing eye-tracking data: From scanpaths and heatmaps to the dynamic visualisation of areas of interest. *Advances in science, technology, higher education and society in the conceptual age: STHESCA* **20**, 25 (2014).
12. Krafka, K. *et al.* Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2176–2184 (2016).
13. Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* **33**, 6999–7019 (2021).
14. Nam, H. & Han, B. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4293–4302 (2016).
15. Schmidt, R. M. Recurrent neural networks (rnns): A gentle introduction and overview. *arXiv preprint* arXiv:1912.05911 (2019).

16. Wang, Q., Yuan, C., Wang, J. & Zeng, W. Learning attentional recurrent neural network for visual tracking. *IEEE Transactions on Multimedia* **21**, 930–942 (2018).
17. Zhang, X., Sugano, Y., Fritz, M. & Bulling, A. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence* **41**, 162–175 (2017).
18. Yang, K., He, Z., Zhou, Z. & Fan, N. Siamatt: Siamese attention network for visual tracking. *Knowledge-based systems* **203**, 106079 (2020).
19. Redmon, J. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016).
20. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* **39**, 1137–1149 (2016).
21. Barz, M. & Sonntag, D. Automatic visual attention detection for mobile eye tracking using pre-trained computer vision models and human gaze. *Sensors* **21**, 4143 (2021).
22. Veličković, P. *et al.* Graph attention networks. *6th International Conference on Learning Representations* (2017).
23. Liu, Z., Jiang, Z., Feng, W. & Feng, H. Od-gcn: Object detection boosted by knowledge gcn. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6 (IEEE, 2020).
24. Zhao, J., Chu, J., Leng, L., Pan, C. & Jia, T. Rgrn: Relation-aware graph reasoning network for object detection. *Neural Computing and Applications* 1–18 (2023).
25. Wei, P., Liu, Y., Shu, T., Zheng, N. & Zhu, S.-C. Where and why are they looking? jointly inferring human attention and intentions in complex tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6801–6809 (2018).
26. Hu, Z., Bulling, A., Li, S. & Wang, G. Ehtask: Recognizing user tasks from eye and head movements in immersive virtual reality. *IEEE Transactions on Visualization and Computer Graphics* (2021).
27. Wu, X. et al. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems* **135**, 364–381 (2022).
28. Wang, H., Jiang, X., Ren, H., Hu, Y. & Bai, S. Swiftnet: Real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1296–1305 (2021).
29. Cheng, H. K. & Schwing, A. G. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, 640–658 (Springer, 2022).
30. Hamilton, W., Ying, Z. & Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems* **30** (2017).
31. Ciano, G., Rossi, A., Bianchini, M. & Scarselli, F. On inductive-transductive learning with graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 758–769 (2021).
32. Qu, M., Cai, H. & Tang, J. Neural structured prediction for inductive node classification. In *International Conference on Learning Representations* (2021).
33. Venuprasad, P. *et al.* Analyzing Gaze Behavior Using Object Detection and Unsupervised Clustering. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Full Papers, https://doi.org/10.1145/3379155.3391316 (Association for Computing Machinery, New York, NY, USA, 2020). Event-place: Stuttgart, Germany.
34. Deane, O., Toth, E. & Yeo, S.-H. Deep-SAGA: a deep-learning-based system for automatic gaze annotation from eye-tracking data. *Behavior Research Methods* https://doi.org/10.3758/s13428-022-01833-4 (2022).
35. Barz, M., Bhatti, O. S., Alam, H. M. T., Nguyen, D. M. H. & Sonntag, D. Interactive fixation-to-aoi mapping for mobile eye tracking data based on few-shot image classification. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, 175–178 (2023).
36. Nguyen, V. D. et al. Deep domain adaptation: A sim2real neural approach for improving eye-tracking systems. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* **7**, 1–17 (2024).
37. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755 (Springer, 2014).
38. Deng, X., Yu, Q., Wang, P., Shen, X. & Chen, L.-C. Coconut: Modernizing coco segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21863–21873 (2024).
39. Batliner, M., Hess, S., Ehrlich-Adám, C., Lohmeyer, Q. & Meboldt, M. Automated areas of interest analysis for usability studies of tangible screen-based user interfaces using mobile eye tracking. *AI EDAM* **34**, 505–514 (2020).
40. Kumari, N. et al. Mobile eye-tracking data analysis using object detection via yolo v4. *Sensors* **21**, 7668 (2021).
41. Kurzhals, K., Hlawatsch, M., Seeger, C. & Weiskopf, D. Visual analytics for mobile eye tracking. *IEEE transactions on visualization and computer graphics* **23**, 301–310 (2016).
42. Panetta, K., Wan, Q., Kaszowska, A., Taylor, H. A. & Agaian, S. Software architecture for automating cognitive science eye-tracking data analysis and object annotation. *IEEE Transactions on Human-Machine Systems* **49**, 268–277 (2019).
43. Kurzhals, K. *et al.* Visual analytics and annotation of pervasive eye tracking video. In *ACM Symposium on Eye Tracking Research and Applications*, 1–9 (2020).
44. Zhou, J. et al. Graph neural networks: A review of methods and applications. *AI open* **1**, 57–81 (2020).
45. Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17 (2017).
46. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)* (2017).
47. Brody, S., Alon, U. & Yahav, E. How attentive are graph attention networks? In *ICLR* (OpenReview.net, 2022).
48. Xu, H., Jiang, C., Liang, X. & Li, Z. Spatial-aware graph relation network for large-scale object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9298–9307 (2019).
49. Santoro, A. *et al.* Relational recurrent neural networks. *Advances in neural information processing systems* **31** (2018).
50. Lin, X., Ding, C., Zhan, Y., Li, Z. & Tao, D. Hl-net: Heterophily learning network for scene graph generation. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19476–19485 (2022).
51. Qin, Z., Yang, S. & Zhong, Y. Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems* **36** (2024).
52. Zhang, M., Wu, S., Yu, X., Liu, Q. & Wang, L. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* **35**, 4741–4753 (2022).
53. Corso, G., Cavalleri, L., Beaini, D., Liò, P. & Veličković, P. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems* **33**, 13260–13271 (2020).
54. Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. S. Gated graph sequence neural networks. In Bengio, Y. & LeCun, Y. (eds.) *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (2016).
55. Shi, Y. *et al.* Masked label prediction: Unified message passing model for semi-supervised classification. In Zhou, Z. (ed.) *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, 1548–1554, https://doi.org/10.24963/IJCAI.2021/214 (ijcai.org, 2021).
56. Zeng, H., Zhou, H., Srivastava, A., Kannan, R. & Prasanna, V. GraphSAINT: Graph sampling based inductive learning method. In *International Conference on Learning Representations* (2020).
57. Prummel, W., Giraldo, J. H., Zakharova, A. & Bouwmans, T. Inductive graph neural networks for moving object segmentation. *arXiv preprint* arXiv:2305.09585 (2023).

58. Trajkovska, K., Kljun, M. & Pucihar, K. Č. Gaze2aoi: Open source deep-learning based system for automatic area of interest annotation with eye tracking data. *arXiv preprint* arXiv:2411.13346 (2024).
59. Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J. & Fernández-Leal, Á. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review* **56**, 3005–3054 (2023).
60. Saeed, A., Spathis, D., Oh, J., Choi, E. & Etemad, A. Learning under label noise through few-shot human-in-the-loop refinement. *arXiv preprint* arXiv:2401.14107 (2024).
61. Yao, R., Lin, G., Xia, S., Zhao, J. & Zhou, Y. Video object segmentation and tracking: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* **11**, 1–47 (2020).
62. Zhou, T., Porikli, F., Crandall, D. J., Van Gool, L. & Wang, W. A survey on deep learning technique for video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**, 7099–7122 (2022).
63. Song, E. *et al.* Moviechat: From dense token to sparse memory for long video understanding. *arXiv preprint* arXiv:2307.16449 (2023).
64. Huang, W. *et al.* Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint* arXiv:2307.05973 (2023).
65. Tschernezki, V. *et al.* Epic fields: Marrying 3d geometry and video understanding. *Advances in Neural Information Processing Systems* **36** (2024).
66. Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448 (2015).
67. Jiang, P., Ergu, D., Liu, F., Cai, Y. & Ma, B. A review of yolo algorithm developments. *Procedia Computer Science* **199**, 1066–1073 (2022).
68. Shi, Y. *et al.* Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint* arXiv:2009.03509 (2020).
69. Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision* **88**, 303–338 (2010).
70. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980 (2014).
71. Kelleher, J. D., Mac Namee, B. & D'arcy, A. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies* (MIT press, 2020).
72. Intel. Computer vision annotation tool (2021).
73. Kukkala, V. K., Tunnell, J., Pasricha, S. & Bradley, T. Advanced driver-assistance systems: A path toward autonomous vehicles. *IEEE Consumer Electronics Magazine* **7**, 18–25 (2018).
74. Baldisserotto, F., Krejtz, K. & Krejtz, I. A review of eye tracking in advanced driver assistance systems: An adaptive multi-modal eye tracking interface solution. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications*, 1–3 (2023).
75. Zhang, L. *et al.* Learning unsupervised world models for autonomous driving via discrete diffusion. *International Conference on Learning Representations* (2024).
76. Shi, J.-X. *et al.* Long-tail learning with foundation model: Heavy fine-tuning hurts. *International Conference on Machine* (2024).
77. Barz, M., Bhatti, O. S., Alam, H. M. T., Nguyen, D. M. H. & Sonntag, D. Interactive Fixation-to-AOI Mapping for Mobile Eye Tracking Data Based on Few-Shot Image Classification. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23 Companion, 175–178, https://doi.org/10.1145/3581754.3584179 (Association for Computing Machinery, New York, NY, USA, 2023). Event-place: Sydney, NSW, Australia.
78. Jiang, Y. *et al.* Ueyes: Understanding visual saliency across user interface types. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–21 (2023).
79. Yfantidou, S. *et al.* The state of algorithmic fairness in mobile human-computer interaction. In *Proceedings of the 25th International Conference on Mobile Human-Computer Interaction*, 1–7 (2023).
80. Shaily, R., Harshit, S. & Asif, S. Fairness without demographics in human-centered federated learning. *arXiv preprint* arXiv:2404.19725 (2024).
81. Marinó, G. C., Petrini, A., Malchiodi, D. & Frasca, M. Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing* **520**, 152–170 (2023).
82. Xu, C. & McAuley, J. A survey on model compression and acceleration for pretrained language models. *In Proceedings of the AAAI Conference on Artificial Intelligence* **37**, 10566–10575 (2023).
83. Bolya, D. *et al.* Token merging: Your ViT but faster. In *International Conference on Learning Representations* (2023).
84. Tran, H.-C. *et al.* Accelerating transformers with spectrum-preserving token merging. *arXiv preprint* arXiv:2405.16148 (2024).

Acknowledgements

This work was funded, in part, by the European Union under grant number 101093079 (MASTER), the German Federal Ministry of Education and Research (BMBF) under grant number 01IW23002 (No-IDLE), and the Lower Saxony Ministry of Science and Culture (MWK) as part of the project zug.KI. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Duy M. H. Nguyen.

Author contributions

Hoang H. Le and Duy M. H. Nguyen implement the proposed method and run experiments on datasets. Omair, Laszlo, and Michael support the collection of datasets, providing tools for annotation. Thinh Ngo supports in providing data annotations Binh T. Nguyen, Michael and Daniel Sonntag guide the project.

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

We confirm that this study used a dataset sourced with approval from the Ethics Board of the University of Saarland, ensuring adherence to all relevant ethical guidelines and regulations, and we have the right to use this dataset for our research. The dataset did not involve direct interaction with participants by the authors. The human annotations referenced in the study were performed by employed students who were compensated for their work and conducted the annotations as part of their professional duties.

Additional information

Correspondence and requests for materials should be addressed to H.H.L. or D.M.H.N.

Reprints and permissions information is available at www.nature.com/reprints.