Supplementary Material for SaccadeX

A. Motivation

A.1. Limitations of Existing Event-based Methods

Although event cameras are well-suited for eye tracking, most existing methods fail to fully leverage their asynchronous, sparse, and high-frequency nature [3]. Many systems adopt frame-based approximations of event streams by temporally aggregating events into fixed intervals, thereby converting the data back into image-like structures [31, 41, 42]. While this makes it easier to apply convolutional architectures, it also sacrifices the precise temporal ordering that makes event data unique. Others rely on heuristic event thresholds or velocity cues to segment saccades or detect fixations [23], but these methods are highly sensitive to noise, and unable to generalize to complex, non-canonical movement patterns.

On the supervision front, several recent works train fully supervised models using dense annotations from RGB-based eye trackers [1, 50]. However, the ground-truth labels are sparse relative to the event stream and often misaligned due to differences in sensing modality. Hybrid systems that combine event data with synchronized RGB images attempt to address this, but they inherit the limitations of the frame-based modality: motion blur, occlusion, and temporal lag [4, 49].

In contrast, our method is event-centric and label-efficient. Using a semi-supervised framework, it learns from sparse annotations while exploiting the DAG structure to interpolate high-frequency motion. This enables dense, continuous trajectory reconstruction without dense labeling or synchronized video, and supports generalization to high-speed, free-viewing scenarios that conventional frame-based sensors cannot reliably capture.

A.2. Directed Acyclic Graphs for Event Representation

Effective event representations for ocular motion must capture temporal causality, preserve sparsity, and support efficient processing of asynchronous data. Directed Acyclic Graphs (DAGs) provide these properties by linking each event to its forward-in-time, spatially local neighbors. This construction ensures acyclicity, admits a natural topological order, and prevents information leakage from future to past events. Node attributes encode spatial trajectories, while directed edges capture temporal progression, together preserving both granularity and continuity of pupil dynamics. Unlike frame-based aggregation, which blurs high-frequency motion, or fully connected graphs, which incur quadratic cost, DAGs encode the causal structure of eye movements in a compact and computationally efficient form, making them an ideal foundation for event-based tracking.

A.3. Semi-supervised Framework for Label Sparsity

A key challenge of event-based ocular motion tracking is the extremely high event rate of event cameras (up to 1.06 Geps for dynamic scenes), leading to the label sparsity problem. Annotations for event streams are typically derived from RGB frames recorded at much lower temporal frequencies (e.g., the DAVIS 346 captures grayscale images at 30 fps, and the EV-Eye dataset [50] provides gaze points at 100Hz using a commercial eye tracker). This sparsity limits the resolution and accuracy of supervised learning, which is crucial for precise tasks like pupil tracking and gaze estimation.

To address this challenge, we propose a novel framework that generates dense, event-level labels through semi-supervised learning, using sparse RGB-derived labels as temporal anchors. This approach exploits the high-frequency nature and temporal continuity of the event stream to interpolate intermediate labels, maintaining consistency with the sparse ground truth. By aligning events with RGB timestamps, the framework learns a temporal mapping that captures the microdynamics of eye movements. The integration of semi-supervised learning enhances model generalization, leveraging both labeled and unlabeled event data. This method effectively mitigates label sparsity and maximizes the utility of event data, thereby improving the accuracy and granularity of eye-tracking systems.

A.4. Applications

Fine-grained eye trajectories are more than just biomechanical signatures – they are rich, temporally structured signals of internal cognitive, affective, and attentional states. The frequency of microsaccades has been associated with moment-to-moment shifts in arousal and vigilance; saccade curvature and trajectory irregularities can indicate decision conflict, motor inhibition, or uncertainty; and saccade landing dynamics reflect anticipatory processing and intent formation. These microtemporal features are largely invisible in conventional gaze datasets but become tractable through dense, event-based trajectory reconstruction.

The ability to reconstruct eye movements at sub-frame granularity uncovers a new class of context-aware and cognition-sensitive applications. Beyond anticipatory user interfaces (which adapt content before a fixation is complete), fine-grained gaze dynamics can enable multimodal fusion for affective computing, where subtle changes in eye motion are integrated with speech, posture, or physiological data to infer emotion, stress, or social engagement.

In education, micro-saccadic behavior during problem solving or concept transitions can provide insight into cognitive load, conceptual confusion, or disengagement—enabling real-time interventions. In neuroadaptive learning environments, such signals could be used to dynamically modulate pacing, highlight key visual cues, or prompt reflection at moments of cognitive dissonance.

In mental health and neurodiagnostics, trajectory-level irregularities may serve as early biomarkers for conditions like Parkinson's disease, ADHD, or anxiety disorders, where oculomotor control is subtly disrupted. Monitoring these patterns continuously and passively could support early screening, therapy personalization, or long-term behavioral tracking in naturalistic settings.

B. Theoretical Justification for DAG for Event Vision

In this section, we provide a theoretical justification for representing event streams as timestamp-respecting directed acyclic graphs (DAGs). This construction is not only acyclic by design but also guarantees causal ordering, admits a natural topological (time) order, preserves the correct probabilistic factorization under causal Markov assumptions, and enables efficient $O(Nk)$ message passing with bounded degree. We further contrast against alternative graph constructions (undirected, bidirectional, fully connected, or spatial-only) and show via counterexamples that they either violate causality, permit future-to-past information leakage, or are computationally intractable. These results establish timestamp-respecting DAGs as the natural and theoretically sound choice for modeling asynchronous event streams.

Definition B.1 (Timestamp-respecting graph)**.** *Fix a spatio-temporal neighborhood predicate $\mathcal{N}(\cdot)$ such that $j \in \mathcal{N}(i)$ if event v_j lies within a chosen spatio-temporal radius of ζ_i. Define the directed graph $G = (V, E)$ with $V = \mathcal{E}$ and $E = \{(i \to j) : j \in \mathcal{N}(i),\ t_i < t_j\}$. We call such G a* timestamp-respecting graph*. As the time-respecting is strict, G is also a DAG (i.e., yields a valid topological order).*

Assumption B.2 (Causal Markov property)**.** *Each event e_j depends only on a finite set of* parents $\mathrm{Pa}(j) \subset \{i : t_i < t_j\}$*. Formally, $P(e_j \mid \{e_i : t_i < t_j\}) = P(e_j \mid \mathrm{Pa}(j))$.*

Corollary B.3 (Factorization)**.** *Under assum. (B.2), the joint distribution factorizes: $P(e_1, \dots, e_N) = \prod_{j=1}^{N} P\big(e_j \mid \mathrm{Pa}(j)\big)$.*

Proof. Order the events by increasing timestamps: $e_{(1)}, \dots, e_{(N)}$. By the chain rule, $P(e_{(1)}, \dots, e_{(N)}) = \prod_{j=1}^{N} P\big(e_{(j)} \mid e_{(1)}, \dots, e_{(j-1)}\big)$. By the causal Markov property (B.2), each factor reduces to $P(e_{(j)} \mid \mathrm{Pa}(j))$, giving (B.3). □

Corollary B.4 (No temporal leakage)**.** *Let M be any message-passing computation on G where messages propagate only along edge directions. Then the state or prediction at node j depends only on events i with $t_i < t_j$.*

Proof. Because G is acyclic and edges satisfy $t_i < t_j$, the ancestors of j in G are all earlier in time. Any message passed to j must originate from an ancestor. Thus prediction at j depends only on events with timestamps strictly smaller than t_j. □

Corollary B.5 (Complexity)**.** *Suppose each node has at most k parents. Then $|E| = O(Nk)$, and message passing over G costs $O(Nk)$ operations.*

Proof. Each of N nodes contributes at most k incoming edges. Hence $|E| \le Nk$, and each edge is processed once in message passing. □

Corollary B.6 (Minimal I-map)**.** *If the true generative model satisfies the causal Markov property with parent sets* $\mathrm{Pa}(j)$*, then the DAG with edges $\{i \to j : i \in \mathrm{Pa}(j)\}$ is a minimal I-map of the distribution.*

Proof. By corollary B.3, the DAG is an I-map. Minimality follows: removing any true parent edge $i \to j$ would imply $P(e_j \mid \mathrm{Pa}(j) \setminus \{i\}) = P(e_j \mid \mathrm{Pa}(j))$, contradicting the assumption that i is a genuine parent. □

The results establish that timestamp-respecting DAGs:

1. Are guaranteed acyclic and admit a natural topological (time) order.
2. Provide the correct probabilistic factorization under causal Markov assumptions.
3. Prevent temporal leakage, ensuring causality-preserving predictions.
4. Enable $O(Nk)$ online inference for bounded-degree neighborhoods.

5. Constitute a minimal I-map, embedding the correct independencies while limiting model complexity.

These properties justify DAGs as a natural representation for event-based vision. Further, below we examine several alternative graph constructions and show, via short proofs and counterexamples, how they fail to satisfy above key requirements for modeling asynchronous event streams. For concreteness we use the same notation as above.

B.1. Undirected graphs (symmetric message passing) violate causality

Proposition 1 (Undirected graphs allow temporal leakage)**.** *Let $G_{undirected} = (V, E)$ be formed by connecting events that are spatio-temporally close but with undirected edges (i.e. $(i, j) \in E \iff (j, i) \in E$). Then symmetric message passing on $G_{undirected}$ can enable a node j to depend on events with timestamps $> t_j$ (future events), violating causal ordering required for online inference.*

Counterexample. Consider three events a, b, c with timestamps $t_a < t_b < t_c$ and suppose the true generative process is the causal chain $a \to b \to c$ (so b's parent is a, and c's parent is b). Construct an undirected triangle connecting a, b, c. Under symmetric message passing (e.g., two rounds of standard GCN updates),

$$h_b^{(1)} \leftarrow \mathrm{AGG}\big(h_a^{(0)}, h_b^{(0)}, h_c^{(0)}\big)$$

so the embedding $h_b^{(1)}$ can incorporate information from c (a future event). Thus a predictor at time t_b that uses $h_b^{(1)}$ may exploit features of c, which are not available in an online/streaming setting. Hence temporal leakage occurs. □

Consequence. Training with undirected graphs can produce models that rely on future information; such models will fail under online deployment or when required to make causal predictions.

B.2. Bidirectional temporal edges (allowing edges both directions) produce cycles

Proposition 2 (Bidirectional temporal edges admit directed cycles)**.** *If a graph construction permits edges both from earlier to later nodes and from later to earlier nodes (for example by creating edges whenever temporal distance is small but not enforcing an orientation), then directed cycles may exist, preventing topological ordering and complicating streaming inference.*

Counterexample. Let nodes i, j satisfy $t_i < t_j$. If the rule allows both $i \to j$ and $j \to i$ (because the temporal window is symmetric or orientation is ignored), then $i \to j \to i$ is a directed 2-cycle. By Lemma **??**, cycles prevent a global topological order; messages can circulate indefinitely and online one-pass processing is not guaranteed to respect causal availability. □

Consequence. Bidirectional temporal edges break the acyclicity property which is fundamental for streaming/online evaluation and for using Bayesian-network factorization arguments.

B.3. Fully-connected directed temporal graphs are computationally intractable

Proposition 3 (Full temporal connectivity is $O(N^2)$ and over-smooths)**.** *A directed graph that connects each event i to every later event j with $t_i < t_j$ (the fully connected temporal DAG) has $\Theta(N^2)$ edges; this is computationally prohibitive for large N and encourages over-smoothing of local dynamics.*

Proof. Counting edges: for N nodes, each node i connects to roughly $N - i$ later nodes. Summing yields $\sum_{i=1}^{N}(N - i) = \Theta(N^2)$. Thus message passing requires $\Theta(N^2)$ aggregate operations per global update, which is impractical for high-rate event streams. Moreover, when every event influences every later event directly, local causal structure is blurred: information from distant-in-time events is mixed with local signals in a single hop, which can degrade learning of localized, short-range causal dependencies (over-smoothing). □

Consequence. Fully-connected temporal DAGs preserve causality but are impractical; they also eliminate the locality prior that is physically meaningful (signals propagate locally).

Table 7. Summary of violations from each graph type and how our timestamp-respecting DAG avoids such failure modes.

Graph type	Causality preserved	DAG?	Streaming-friendly	Complexity
Undirected spatial/temporal	No	No	No	$O(Nk)$ (but leaks)
Bidirectional temporal	No	No (cycles)	No	$O(Nk)$
Fully-connected temporal DAG	Yes	Yes	Yes	$O(N^2)$
Spatial-only (undirected)	No (temporal)	No	No	$O(Nk)$
Spatio-temporal undirected	No	No	No	$O(Nk)$
Timestamp-respecting DAG (Ours)	Yes	Yes	Yes (topological)	$O(Nk)$

B.4. Spatial-only graphs (ignore time) break causal factorization

Proposition 4 (Spatial-only graphs fail to represent temporal conditional independences)**.** *If graph edges are formed purely on spatial proximity (ignoring timestamps), then the graph need not represent the conditional independences implicit in a causal temporal generative model. Consequently, the undirected/spatial graph is not an I-map of the true temporal distribution.*

Counterexample: temporal dependence. Consider events a, b at nearby spatial locations but widely separated times, $t_a \ll t_b$, and a later event c at time $t_c > t_b$ whose generation depends only on b and not on a (i.e., c's parent is b). A spatial-only graph connects a and b (and possibly c) by proximity. If one tries to factorize the joint probability based on that undirected/symmetric graph, the graph suggests possible symmetric dependence between a and c (through b), but it lacks the directional/time ordering required to assert that a cannot influence c except via b in time. More concretely, the conditional independence $c \perp a \mid b$ (true given the causal generating process) is not encoded directionally by a spatial-only undirected graph, so it is not an I-map of the causal distribution. □

Consequence. Spatial graphs are good for capturing static spatial structure but they fail to capture temporal causal relationships intrinsic to event streams.

B.5. Spatio-temporal graphs without enforcing time orientation

Proposition 5 (Spatio-temporal undirected graphs allow temporal cycles and leakage)**.** *If edges are formed based on spatio-temporal proximity but without orienting edges according to time (i.e., edges connect i and j when both are close in space-time, but are undirected), then the graph may permit information flow from future to past during symmetric message passing, hence violating causal constraints.*

Proof. This combines the phenomena of sections B.1 and B.4: because the edges are undirected, symmetric message passing can propagate information backward in time. A concrete three-node example (a, b, c) with $t_a < t_b < t_c$ all mutually linked by undirected edges permits c's features to influence b's embedding and then a's embedding via symmetric updates, allowing future information to leak backward. □

Consequence. Without explicit orientation, spatio-temporal graphs fail to guarantee causal predictions and are thus less suited to streaming event inference.

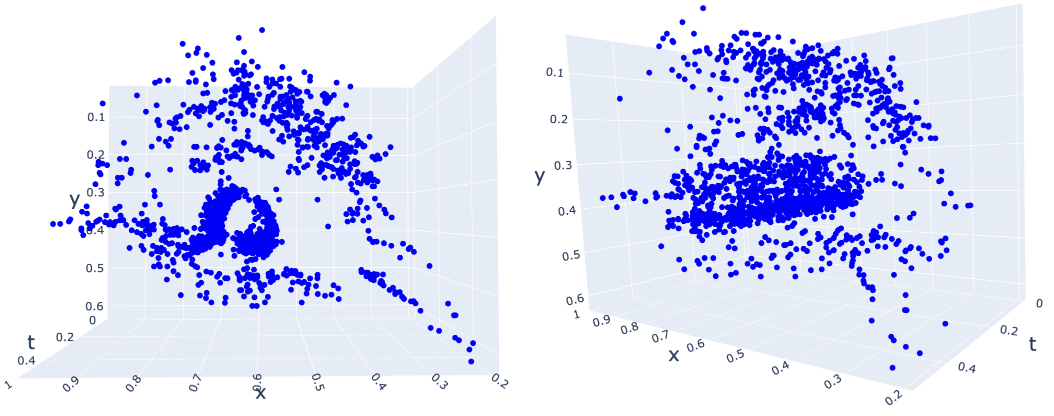
B.6. Summary table of violations

The counterexamples above show that the timestamp-respecting DAG avoids specific failure modes of other constructions: it enforces causality (so that trained models generalize to streaming deployment), it admits a factorization that matches causal generative assumptions, and it enables efficient topologically-ordered computation. Alternative graphs either violate causal constraints, allow information leakage, or are computationally impractical.

C. Qualitative Results Supporting the Spatio-Temporal GNN

As discussed in Sec. 3.3, Fig. 4 illustrates two key properties of near-eye event recordings that motivate our DAG design: (i) *spatial modularity*, where events cluster by anatomical components such as pupil/iris, eyelids, and brows, and (ii) *temporal continuity*, where event streams preserve smooth motion within each component.

To further support our curriculum learning strategy for handling graph heterogeneity, Fig. 5 shows that graphs with fewer nodes provide less reliable information. Such sparse graphs typically arise from (1) incomplete anatomical capture

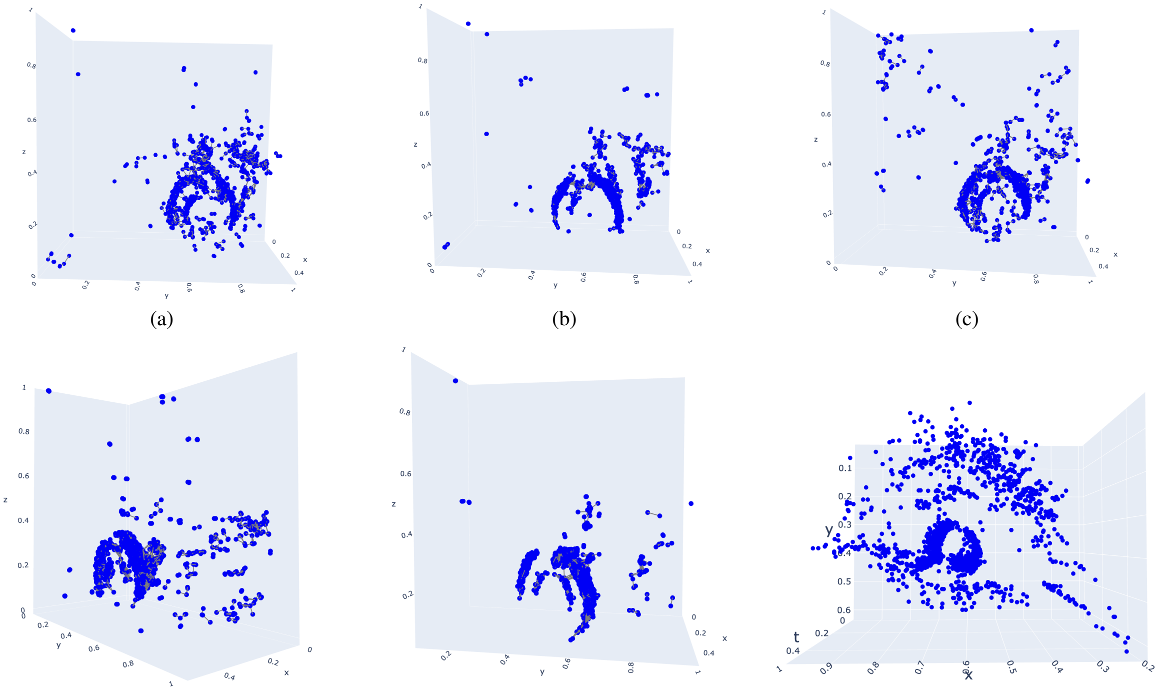


(a)

(b)

Figure 4. Empirical results showing the near-eye event recordings exhibit (i) spatial modularity and (ii) temporal continuity

(Fig. 5(a–c)) or (2) discontinuous temporal trajectories (Fig. 5(d–e)). In contrast, graphs with more nodes (Fig. 5(f)) offer richer anatomical coverage and smoother trajectories, making them more informative and stable for training. This motivates our progression from larger, information-rich graphs to smaller, noisier ones in curriculum learning.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5. Empirical results showing the graph heterogeneity: the observation which lead us to embrace a curriculum learning based model training strategy.

D. More Details on the Design Choices behind DAG and Spatio-Temporal GNN

D.1. Hawkes Process-based Edge Encoding

Our design choice to encode the directed edges in the DAG using Hawkes process is inspired by its demonstrated success in [3] where the influence of historical events are captured holistically through the Hawkes-based features. To be specific, Hawkes-process is a self-exciting (i.e., in other words, one arrival, at a certain timestamp, creates an elevated probability of further arrivals in the near future) point process which is modelled to capture the effect of past events, especially in cases where event clustering or bursts of events/activity is naturally observed. In our implementation, we follow the exact version presented in [3] with the number of divisions to be 8, and the decay factor to be 0.5. We refer to readers to [3] for more details on the algorithm.

D.2. Dual-branch Architecture in Spatio-Temporal GNN

As briefed in section 3.3 in the main paper, we design a dual-branch architecture (i.e., spatial-processing block and temporal-processing block) to explicitly address the key properties: spatial modularity and temporal continuity (as observed in Fig. 4) in near-eye event recordings.

The high-level motivation behind the spatial branch design is to model an encoder which has the capability to accurately capture the observed structural coherence within the near-eye event data which can then be translated into deriving (or retrieving) the ocular dynamics information through a learned process. As depicted in Fig. 1a in the main text (and Fig. 9 in the supplementary), our spatial branch is comprised of a sequential convolution-oriented architecture with four notable components: (1) a N (in our implementation, $N = 2$) number of GCN layers at the beginning to aggregate the information present in the local node neighbourhood to construct a feature space where the nodes from distinct anatomical components result in modularized positions, which is prominently present than in the raw (input) spatial space, (2) stage-wise DMoN layers to learnably cluster and subsample spatially smooth, yet distinct, regions in the constructed feature space corresponding to separate anatomical components, (3) dense graph convolution layers, after DMoN, to lightly smoothen the modularized feature space as needed per the learning objective signal, and (4) skip connections to ensure that the learning signal is strong enough throughout the smoothing and modularization process. The activation is set to *elu* for faster convergence.

The objective of the temporal branch is to model smooth event evolution using an encoder f_t while preserving the temporal continuity of events as observed in near-eye recordings. Ideally, to faithfully model the sequence temporal propagation, one has to consider per-pixel-level implementation while anticipating the inter-dependencies between the pixels to be handled by a parallel spatial branch or by a dedicated cross-information fusion mechanism within the temporal branch itself. However, as this is considerably computationally expensive (and given that we are specifically interested in a computationally budget-friendly design), we systematically simplify our objective from retrieving the per-pixel sequence propagation to the collective (or relative) sequence propagation instead. To this end, we first draft a histogram of the number of events (within the accumulated event volume) vs the timestamp as a faithful representative of the collective sequence propagation, upon which the learning process is induced. The learning layers consist of (1) LSTM, to retrieve the temporal dependencies present and (2) a self-attention layer to learnably realize what dependencies to focus on as guided by the learning objective signal.

E. Procedural Calibration for Gaze Tracking

Inspired by the explicit calibration required for experimental gaze tracking studies [36], here we train the model for gaze tracking using two steps:

- Calibration training: In this first step which is designed to mimic the calibration step in experimental studies, the model is trained with the data where the labels are either *near* minimum possible or maximum possible values. As an example, if the dataset is constructed by asking participants to look at a digital screen positioned at a distance z, if the top-left corner of the screen is considered to be $(0, 0, z)$, and if the height and width of the screen in pixels are H and W respectively, then we impose a calibration threshold $c \in [0, 0.25]$ such that only the input event streams corresponding to the labels within $[0, 0, z]$ to $[cW, cH, z] \cup [(1-c)W, (1-c)H, z]$ to $[W, H, z]$ are considered in this step.
- Training with the rest of the data as usual

F. Adaptation Details for Eye-based Emotion Recognition

Raw Event Simulation Since the SEE [48] only contains the aggregated event frames (accumulated under a fixed time interval strategy) and the corresponding RGB frames, and our approach depends on raw event streams to construct DAGs, we simulate the raw events using RGB frames via the v2e [17] framework.

Event Accumulation Compared to pupil or gaze tracking which depend on the global-scale spatial and short-term temporal heuristics [31] primarily due to rapid movement changes [1], emotion expression is relatively more static and slower where an emotion typically lasts at least for hundreds of milliseconds [48]. Inspired by this observation, we explicitly adopt a fixed time-based event accumulation strategy (after simulating the raw events as above) only for emotion recognition adaptation such that the empirically selected threshold for fixed time interval is adequate for the model to capture the long-range emotion-induced motion dynamics; a feature which is highlighted by experimental human emotion studies [16].

Edge Construction for DAGs As per our empirical observations and observations made by recent studies [3], even though v2e [17] is highly capable in simulating the events corresponding to a RGB frame at a certain timestamp in consideration, it lacks the ability to sufficiently map the non-linear motion dynamics between frames to events and thereby, the continuity of the event flow between frames in a simulated event recording is sparser than a real event camera recording. To this end, we slightly modify the edge construction mechanism in our DAG approach such that two empirically determined radii are assigned for spatial and temporal domains separately to (1) connect the nodes in the spatial plane meaningfully (i.e., without over or under connections) such that the ocular anatomy is evident, and (2) establish the temporal continuity between frames using sufficient connections between the nodes corresponding to the same ocular anatomical component.

G. Offline Density Accumulation and Extensions

Although the algorithm 2 is presented with a fixed time approach, it is trivial to extend that to (1) fixed number of events-based approach and (2) offline event accumulation. To clarify, we extend the algorithm 2 below to cases (1) (in algorithm 3) and (2) (in algorithm 4).

Algorithm 3 Online density-based event accumulation: Extension for fixed number of events

```
Require: Continuous event stream $E^v_{(t,x,y,p)}$, density threshold $\Delta_d$, the number of events $N_{ev}$, number of spatial partitions
    $N_P$ and event spatial resolution $[x_{max}, y_{max}]$
Set (spatial) partition width: $W_P = x_{max}//\sqrt{N_P}$, height: $H_P = y_{max}//\sqrt{N_P}$ {Here, // denotes integer division}
Set number of partitions in $x$: $N_x = x_{max}//W_P$, in $y$: $N_y = y_{max}//H_P$
Initialize event number $n_{ev} = 0$
while Inference do
    while $n_{ev}\%N_{ev}\ ! =\ 0$ do
        Initialize partition density frame: $D_P = [\mathbf{0}]_{N_x \times N_y}$
        Collect events $E^v \longleftarrow e_i(t,x,y,p)$
    end while
    for each $e_i(t,x,y,p) \in E^v$ do
        Map $e_i(.)$ to partition: $E^v \mapsto \{1, ..., p_i, ..., N_P\}$
        $D_{P\{i\}}[p_i] \longleftarrow D_{P\{i-1\}}[p_i] + 1$
    end for
    Calculate $\mu(D_P)$, $\max(D_P)$
    if $\max(D_P) > (1 + \Delta_d) \times \mu(D_P)$ then
        Accumulate $E^v$
    end if
end while
```

H. Datasets and SOTA Baselines

In Table 8, we summarise the datasets we used and the corresponding baselines we considered.

Dataset	Goal	Attributes	SOTA methods
EBV-Eye	Gaze tracking	RGB (≈ 25 FPS) & events	[1, 3, 23]
EV-Eye	Gaze tracking	RGB (≈ 25 FPS) & events	[50]
3ET+	Pupil tracking	events only	[31, 41, 42, 47]
SEE	Emotion recognition	RGB frames (≈ 25 FPS) & event frames	[16, 44, 48, 51]

Table 8. Summary of Datasets and Baseline Methods

Algorithm 4 Offline density-based event accumulation: Fixed number of events-based

```
Require: Continuous event stream $E^v_{(t,x,y,p)}$, density threshold $\Delta_d$, time bin $T_b$, number of spatial partitions $N_P$ and event spatial resolution $[x_{max}, y_{max}]$
Set (spatial) partition width: $W_P = x_{max}//\sqrt{N_P}$, height: $H_P = y_{max}//\sqrt{N_P}$ {Here, $//$ denotes integer division}
Set number of partitions in $x$: $N_x = x_{max}//W_P$, in $y$: $N_y = y_{max}//H_P$
Initialize time $t = 0$
while Inference do
    Initialize partition density frame: $D_P = [\mathbf{0}]_{N_x \times N_y}$
    $E^v \longleftarrow e_i(t, x, y, p)$
    for each $e_i(t, x, y, p) \in E^v$ do
        Map $e_i(.)$ to partition: $E^v \mapsto \{1, ..., p_i, ..., N_P\}$
        $D_{P\{i\}}[p_i] \longleftarrow D_{P\{i-1\}}[p_i] + 1$
    end for
    Calculate $\mu(D_P)$, $\max(D_P)$
    if $\max(D_P) > (1 + \Delta_d) \times \mu(D_P)$ then
        Accumulate $E^v$
    end if
end while
```

I. Evaluation Metrics

The detailed description on the utilized metrics are presented below. It is to be noted that the notations are consistent with the main body of the paper.

p-accuracy This is used in the recent works [41] to evaluate the performance of pupil coordinate estimation methods when ground truth is present, usually at a coarser resolution. *p-accuracy* is defined as if the Euclidean distance between the predicted coordinates ($pred_i$) and true coordinates ($true_i$) is within a specified pixel threshold (Th), the prediction is said to be correct and vice versa.

$$p\{Th\} = \frac{1}{N}\sum_{i=1}^{N} f(true_i, pred_i, Th) \text{ with } f(true_i, pred_i, Th) = \begin{cases} 1 & \text{if } \|true_i - pred_i\| \leq Th \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In this work, we set three pixel thresholds: 10, 5, 3 and 1, following [41].

Mean Euclidean and Manhattan Distances Two regression metrics are utilized as well: Euclidean distance (l_2) (or MSE) and Manhattan distance (l_1) (or MAE), to further evaluate the pupil coordinates estimation.

$$l_2 = \frac{1}{N}\sum_{i=1}^{N} \|true_i - pred_i\|_2 \tag{4}$$

$$l_1 = \frac{1}{N}\sum_{i=1}^{N} |\, true_i - pred_i \,| \tag{5}$$

Gaze angle error By following [1], we define the following L2 norm error to evaluate the gaze estimation accuracy.

$$GAE = \frac{1}{N}\sum_{i=1}^{N} \|\hat{d}_i(\hat{\xi}, \hat{\theta}) - d_i(\xi, \theta)\|_2 \tag{6}$$

where $\hat{d}_i(\hat{\xi}, \hat{\theta})$ is the predicted gaze direction and $d_i(\xi, \theta)$ is true gaze direction.

Unweighted and Weighted average recall Here, we utilize two widely utilized metrics for assessing emotion classification [48]: unweighted average recall (UAR) and weighted average recall (WAR). In definition, UAR depicts the instances per class-agnostic average accuracy while WAR reflects the accuracy of overall emotions.

$$UAR = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{TP_i}{TP_i + FN_i} \tag{7}$$

$$WAR = \frac{TP + FN}{TP + TN + FP + FN} \tag{8}$$

where N_c, TP, FP, TN and FN are the total number of emotion classes, true positive, false positive, true negative, and false negative, respectively.

J. More Ablations

In this section, we present the ablation results related to DAG construction to analyse the sensitivity of its hyperparameters for the overall downstream performance. As a case study, here we utilize the task of pupil tracking and the dataset to be 3ET+, of which the main results are presented in the Table 1 in the main paper.

When conducting our ablations (and thereby optimizing the overall performance of the pipeline), we incorporate a principled hyperparameter selection which can be seamlessly integrated and followed by the readers when applied in other related dataset settings. We describe the selection process and the results below.

- As presented in section 3.2 in the main paper, λ_1, λ_2, and λ_3 refer to the normalization factors of timestamp, x and y coordinate of an event respectively. By following the typical practice in the vision community, we utilize a min-max normalization strategy in this task: for x, we divide each coordinate by the maximum possible $x-$coordinate pixel value (i.e., width in sensor resolution) and for y, we divide by the maximum possible $y-$coordinate pixel value (i.e., height in sensor resolution). However, as the time axis heavily depend on the event recording time (and, thus, it is impossible to assign a maximum possible value for time), we employ min-max normalization on the timestamp by the considering the accumulated event volume: both minimum possible time and maximum possible time to normalize all timestamps are *dynamically* assigned by the temporal propagation characteristics of the event volume in consideration. We follow the same approach for all our experiments, making λ_2 and λ_3 changing only when the experimental dataset is changed (due to different sensor resolutions in the datasets), and λ_1 changes dynamically even within one dataset per event volume. Therefore, as presented in Table 9, λ_2 and λ_3 is, consistently, $1/640$ and $1/480$ respectively in the 3ET+ experiments as the sensor resolution of the event sensor in 3ET+ dataset is $(640, 480)$.
- The ablations around the spatio-temporal threshold and the node degree conditions are based on the premise of the higher bandwidth of event vision sensors which allow them to capture almost all seemingly connected movement dynamics in the scene. Therefore, it is plausible to assume that the events with immediate previous timestamps under a defined receptive field are to be connected with each other as they are potentially generated from the same (or related) background movement. As presented in Table 9, we conduct the ablations to find the optimal size of the receptive field, via ζ, and the optimal number of connections within a receptive field, via D_{min} and D_{max}. As per our findings, having a smaller receptive field ($\zeta \approx 0.0015$ as in ablation 1) is better than having a larger receptive field ($\zeta \approx 0.003$ as in ablation 2 and $\zeta \approx 0.006$ as in ablation 3) under the same node degree conditions. Further, we find that the number of connections within the selected receptive field is dependent on the task at hand: in the task of pupil tracking on 3ET+ dataset, we find that the downstream performance is higher when (1) $D_{min} = 1$ (ablation 1), rather than $D_{min} = 2$ (ablation 4), and (2) D_{max} is at an optimal value: 8 as in ablation 1, than being significantly lower as in 4 (ablation 5) or highr as in 16 (ablation 6). We believe that this principled process and the results would guide the readers in faithfully and optimally deciding the effect of receptive field and the sufficient number of connections to be enforced when constructing event graphs, at least for the event-based eye tracking.

K. Extended Complexity Analysis

Here, we explicitly analyse the practical computational complexity overhead of DAG, (1) in comparison to other alternative graph structures and event representations, and (2) contextual complexity propagation tricks and the effect of subsampling.

Timestamp-respecting DAG as an event representation Here, in Table 10, we present the full processing time results, in complementary to our discussion in section 6.4.4 in the main paper, to highlight our DAG as an efficient and viable event

Table 9. DAG construction ablations on the pupil tracking using 3ET+ validation dataset. Here, we only change the the hyperparameters of DAG construction; i.e., the model architecture and learning parameters remain the same as our best performing model.

Ablation number	ζ	D_{min}	D_{max}	λ_1	λ_2	λ_3	$l_2\downarrow$
Ablation 1	$\sqrt{(\frac{1}{640})^2 + (\frac{1}{480})^2} \approx 0.0015$	1	8	dynamic	$1/640$	$1/480$	1.40
Ablation 2	$\sqrt{(\frac{2}{640})^2 + (\frac{2}{480})^2} \approx 0.003$	1	8	dynamic	$1/640$	$1/480$	1.51
Ablation 3	$\sqrt{(\frac{4}{640})^2 + (\frac{4}{480})^2} \approx 0.006$	1	8	dynamic	$1/640$	$1/480$	1.63
Ablation 4	$\sqrt{(\frac{1}{640})^2 + (\frac{1}{480})^2} \approx 0.0015$	2	8	dynamic	$1/640$	$1/480$	1.44
Ablation 5	$\sqrt{(\frac{1}{640})^2 + (\frac{1}{480})^2} \approx 0.0015$	1	4	dynamic	$1/640$	$1/480$	1.43
Ablation 6	$\sqrt{(\frac{1}{640})^2 + (\frac{1}{480})^2} \approx 0.0015$	1	16	dynamic	$1/640$	$1/480$	1.54

Table 10. Processing time (in milliseconds) of N events by each event representation method. Here, we report the average time to process the events using a subset in 3ET+ dataset on one T4 GPU.

Event representation				N		
	100	500	1000	2000	4000	10000
Event frame	0.301	1.155	2.307	4.284	8.136	21.043
Time surface	0.255	0.985	1.975	3.822	7.491	19.694
Voxel grid	1.005	2.331	3.995	6.984	11.978	28.436
TORE [2]	2.441	10.516	21.556	41.657	84.777	216.947
Undirected Spatio-Temporal graph	0.282	0.639	1.110	1.953	3.663	9.035
Fully-connected DAG	1.355	7.191	19.215	55.326	172.534	2260.864
Time-respecting DAG	0.402	0.774	1.318	2.239	3.967	9.526

representation.

Contextual DAG complexity propagation and subsampling In our DAG approach, we address the challenge of graph complexity via the following consecutive steps:

- Since event cameras trigger events independently based on intensity changes, a key preprocessing step in event vision is event accumulation—buffering a set number of events to ensure informative scene dynamics in a representation-agnostic manner [3]. Following prior works [23, 37], we adopt a fixed event count accumulation (e.g., collecting $N = 4000$ events, $\approx 30ms$ on average) to maintain a stable motion representation and manage computational complexity by avoiding sparse adjacency matrices. At the inference stage, we further optimize by using density-based event accumulation (sec 5 in the paper), which selects events based on spatial density before accumulation. As shown in Fig. 8, frames highlighted in red are discarded, as they lack relevant pupil movement information, reducing computational load by processing fewer events.
- As shown in Fig. 1a, within the spatial encoder, we employ a deep modularity network (DMoN) for learnable sub-sampling, dynamically selecting the most informative graph nodes. This preserves critical anatomical and functional eye structures while significantly reducing graph complexity. By applying two DMoN layers, we progressively, i.e., starting with the initial layers, reduce the adjacency matrix complexity to 25% and 3.13% of the original input. In addition, it is to be noted that our method maintains full temporal resolution despite learnable event subsampling because the subsampling does not occur at the input level i.e., specifically during the construction of the spatio-temporal graph, rather within the learning framework.
- Further, it is to be highlighted that the EAE operates separately from the tracking model, serving only for post-hoc interpretability. It receives latent inputs from the trained tracking model without influencing gaze or pupil prediction performance. Since the EAE is trained after the core model with frozen gradients, it does not introduce additional computational overhead to the tracking pipeline.

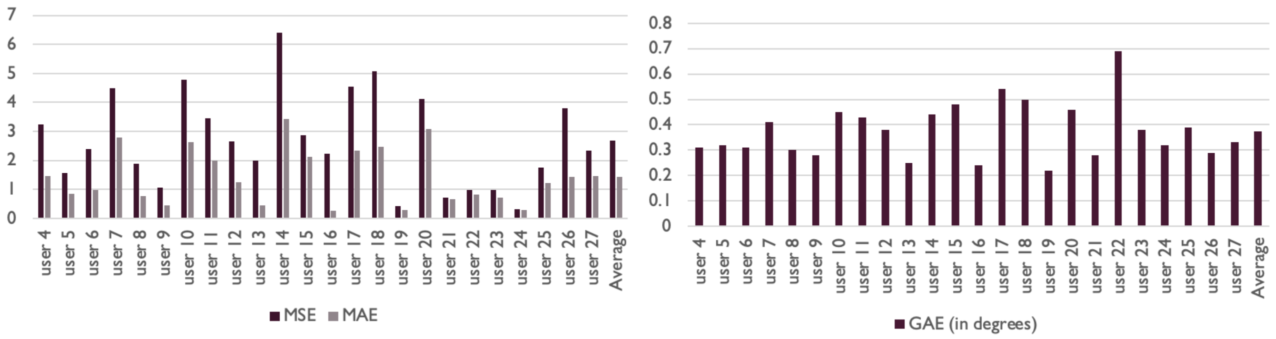


Figure 6. Extensive user-basis results for EBV-Eye [1] dataset

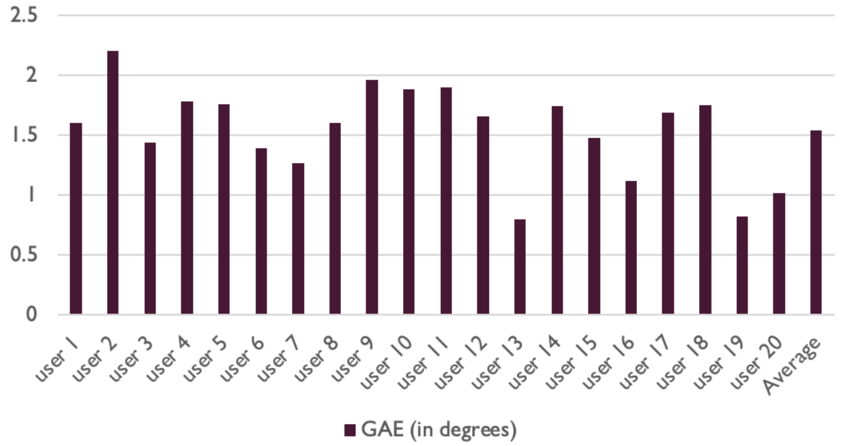


Figure 7. Extensive user-basis results for EV-Eye [50] dataset

L. More (Visual) Results

In the following section, we highlight and extend our performance analysis. In Fig. 6, we plot the per-user mean squared error (MSE), Mean absolute error (MAE), and gaze angle error (GAE) estimated on the EBV-Eye dataset. Similarly, in Fig. 7, we depict the per-user gaze angle error on Ev-Eye dataset. Based on the two figures, we observe that our approach achieves generalizable performance across datasets.

Figure 8 illustrates selected empirical examples that demonstrate how the proposed prior step selectively blocks “uninformative” event volumes from the accumulation process. In these examples, the frames marked in red are explicitly excluded from the accumulation, as they do not contain any significant movement or changes (i.e., they lack “excitement”). This selective filtering ensures that only the most informative frames are included, thus improving efficiency. In contrast, the vanilla baseline includes all frames, which leads to unnecessary computational and time overhead, as even frames without meaningful content are processed.

M. Extended Clarifications

Below we explicitly clarify a set of potential concerns for better readability of the paper.

Event accumulation It is to be noted that our DAG approach maintains entire fine-grained temporal resolution despite event buffering when constructing the graphs. To be specific, unlike frame-based methods, where events within a frame share a single timestamp (i.e., effectively smoothing temporal information), our DAG-based spatiotemporal graph preserves the original timestamps of individual events. Each node in the graph retains the exact timestamp of the corresponding event,

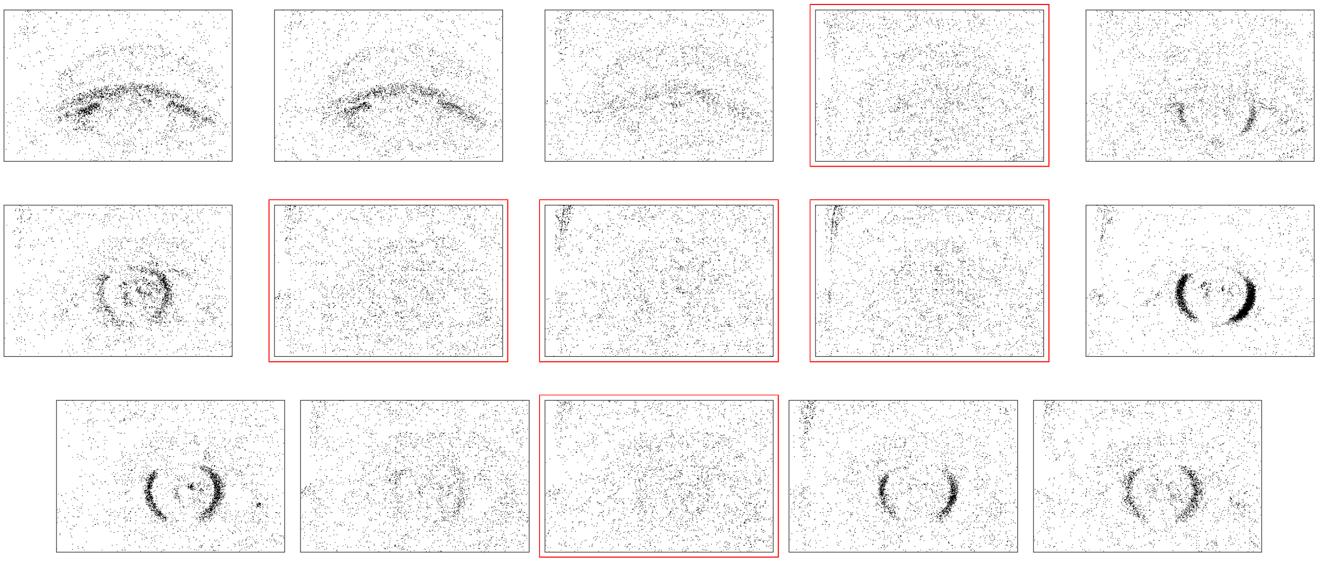


Figure 8. Each row presents a consecutive set of scene dynamics where each event-frame encompasses 4000 ocular event data from EyeGraph [3] dataset. Note that there is no relationship between rows. With our accumulation algorithm, the event-frames highlighted in red are discarded (this also aligns with the human observations).

ensuring that no temporal information is lost during accumulation. This allows our method to retain high temporal fidelity while still leveraging event aggregation for more efficient processing.

Semi-supervised learning for classification Due to the limited availability of temporally-propagated labels in the SEE dataset (i.e., which has only one label for a whole event recording), the emotion recognition, which is the only classification task we experiment in this work, cannot be meaningfully integrated with our semi-supervised strategy. Even so, we argue that our method is still valid for suitable classification datasets, especially where temporally propagating labels are meaningful (e.g., facial micro-expression recognition over time), thus, can be practically applied on.

N. Edge Masking for Pseudo-Interpretability

As widely discussed in the literature [24], the limited interoperability of deep learning models limits their adoption in physiopsychological applications like eye tracking, where interpretability is crucial for clinical validation and diagnostic reliability. Without transparency, it is difficult to determine whether a model captures meaningful physiological patterns or spurious correlations. To address this, as depicted in Fig. 9, we propose an edge autoencoder (EAE) architecture that learns graph topology while identifying key edges in event-based eye tracking data. To be specific, we aim to achieve partial interpretability through post-hoc rationalization by assessing the tracking model's latent space via an edge edge auto-encoder by minimizing the dissimilarity between the latent representations of the tracking and EAE models, assuming that if the tracking model's latent vector can be approximated using only the edges in the event graph, the highly weighted edges correspond to the most salient spatial and temporal regions. This approach enhances interpretability by revealing the structural dependencies in gaze dynamics, making the model more transparent and trustworthy for clinical and research applications.

The implemented edge encoder f_e attempts to map the edge space (E) to a low-dimensional latent vector $Z_e \in \mathbb{R}^{d_s + d_t}$ via a learnable masked edge strategy such that the dissimilarity between Z_{st} and Z_e is minimum: $\|Z_{st} - Z_e\|_{d_l} \leq \delta$. Since Z_{st} is the learned latent vector of event-based eye movement graph, our objective here is to approximate that knowledge (via Z_e) only with edges and consequently, recognize the edges through which the knowledge is passed, via the masked edge strategy. Further, the edge decoder f_d is guided via a reconstruction loss to topologically learn the edge structure in the graph, making the weighted joint objective for edge autoencoder be: $\gamma_1 \|Z_{st} - Z_e\|_{d_l} + \gamma_2 \|\mathbf{A} - \hat{\mathbf{A}}\|_{d_r}$.

N.1. Results on Pseudo-Interpretability

Although the proposed framework for eye tracking can be jointly trained, for simplicity in implementation, we first train the eye tracking model independently and subsequently use the pre-trained model to train the graph edge autoencoder. Further, it is to be highlighted that the primary role of the EAE is to enhance interpretability rather than directly contribute to pupil

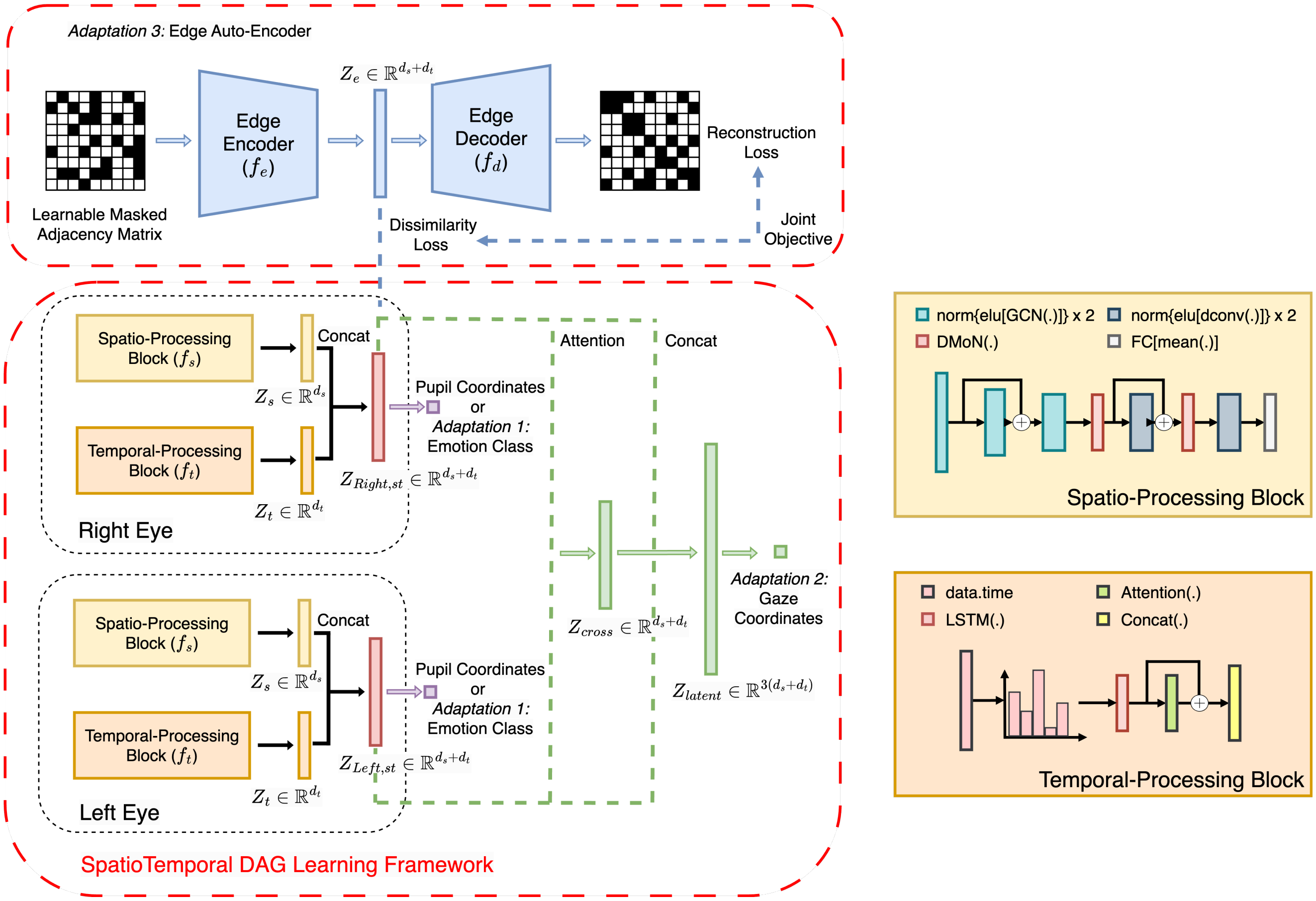

Adaptation 3: Edge Auto-Encoder
Learnable Masked Adjacency Matrix
Edge Encoder (f_e)
$Z_e \in \mathbb{R}^{d_s+d_t}$
Edge Decoder (f_d)
Reconstruction Loss
Joint Objective
Dissimilarity Loss
Spatio-Processing Block (f_s)
Temporal-Processing Block (f_t)
Concat
$Z_s \in \mathbb{R}^{d_s}$
$Z_t \in \mathbb{R}^{d_t}$
Pupil Coordinates or Adaptation 1: Emotion Class
$Z_{Right,st} \in \mathbb{R}^{d_s+d_t}$
Right Eye
$Z_{Left,st} \in \mathbb{R}^{d_s+d_t}$
Left Eye
Attention
Concat
$Z_{cross} \in \mathbb{R}^{d_s+d_t}$
$Z_{latent} \in \mathbb{R}^{3(d_s+d_t)}$
Adaptation 2: Gaze Coordinates
SpatioTemporal DAG Learning Framework
norm{elu[GCN(.)]} x 2
norm{elu[dconv(.)]} x 2
DMoN(.)
FC[mean(.)]
Spatio-Processing Block
data.time
Attention(.)
LSTM(.)
Concat(.)
Temporal-Processing Block

Figure 9. The proposed joint-task paradigm consisting of the DAG learning pipeline and the edge autoencoder for enhanced interpretability.

tracking accuracy. To be specific, the tracking performance is only driven by the primary spatiotemporal model, while the EAE serves as a complementary component that reveals meaningful graph structures, which would otherwise remain latent. This distinction is crucial, as it underlines the broader goal of improving model transparency in event-based eye tracking.

Due to the absence of benchmark datasets or labeled data in the current literature, we empirically evaluate the performance of the proposed edge masking technique. As shown in Fig. 10, the edge autoencoder is able to highlight edges that connect the pupil or iris movement profiles, distinguishing them from other anatomical components of the eye. This suggests that the edge encoder, and by extension, our DAG learning framework — emphasizes the pupil and its motion when estimating the regression coordinates, interestingly aligning with human annotation heuristics, and hinting that the tracking model autonomously learns a physiologically grounded representation. Given this is not a main contribution of this work and to achieve explicit interpretability sense and a quantitative metric, we expect to explore more on this direction in our future works.

References

[1] Anastasios N Angelopoulos, Julien NP Martel, Amit P Kohli, Jörg Conradt, and Gordon Wetzstein. Event-based near-eye gaze tracking beyond 10,000 Hz. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2577–2586, 2021. 1, 2, 5, 6, 7, 8, 11

[2] R Wes Baldwin, Ruixu Liu, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Time-ordered recent event (tore) volumes for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2519–2532, 2022. 7, 10

[3] Nuwan Bandara, Thivya Kandappu, Argha Sen, Ila Gokarn, and Archan Misra. EyeGraph: modularity-aware spatio temporal graph clustering for continuous event-based eye tracking. *NeurIPS*, 2024. 1, 3, 4, 8, 5, 6, 7, 10, 12

[4] Nuwan Bandara, Thivya Kandappu, and Archan Misra. Inference-time gaze refinement for micro-expression recognition: Enhancing event-based eye tracking with motion-aware post-processing. *arXiv preprint arXiv:2506.12524*, 2025. 2, 1

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual International Conference on Machine Learning (ICML)*, pages 41–48, 2009. 4

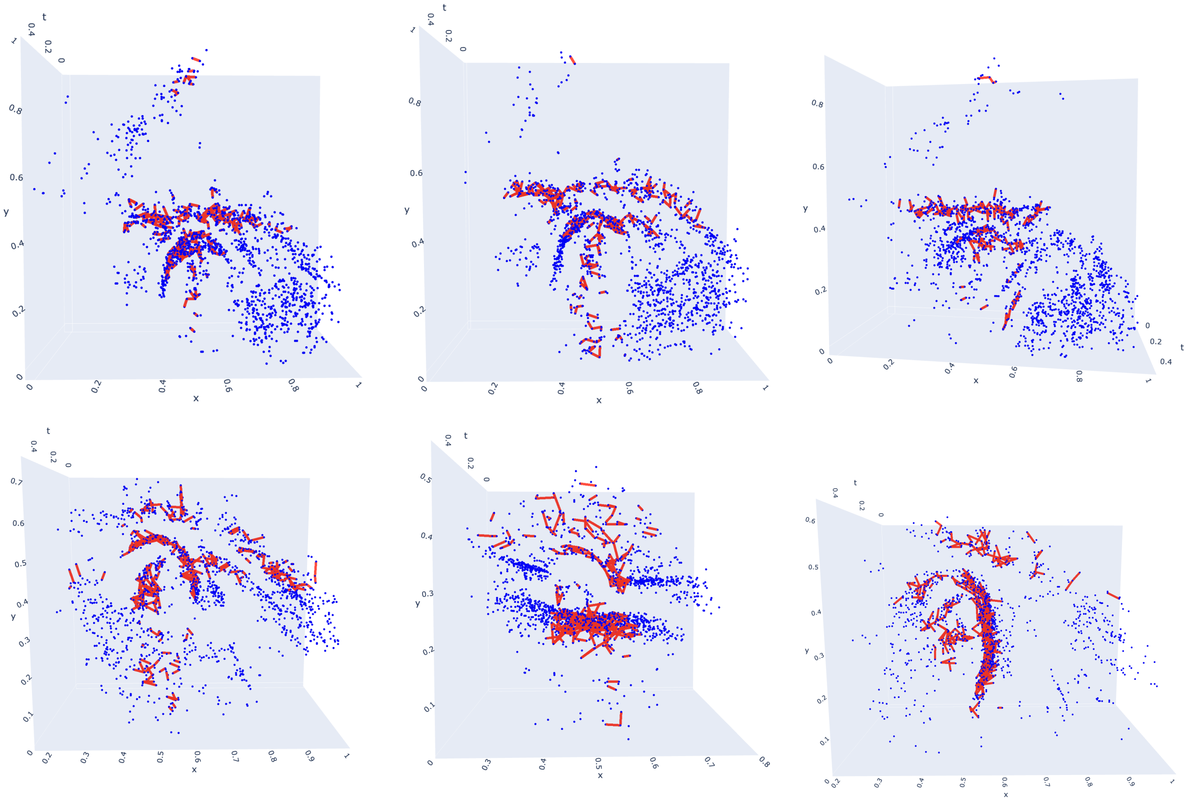


Figure 10. More edge masking results visualizations. Note that only the highlighted edges by the edge encoder are shown in the visualization for better clarity. Here, we illustrate that, for various vision scenes, the constructed graph shows that the edge autoencoder is able to highlight edges (highlighted in red) that connect the pupil or iris movement profiles, distinguishing them from other anatomical components of the eye.

[6] Aritra Bhowmick, Mert Kosan, Zexi Huang, Ambuj Singh, and Sourav Medya. DGCLUSTER: A neural framework for attributed graph clustering via modularity maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11069–11077, 2024. 4

[7] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 491–501, 2019. 2

[8] Pietro Bonazzi, Sizhen Bian, Giovanni Lippolis, Yawei Li, Sadique Sheik, and Michele Magno. Retina : Low-power eye tracking with event camera and spiking hardware, 2024. 2

[9] Qinyu Chen, Zuowen Wang, Shih-Chii Liu, and Chang Gao. 3ET: Efficient event-based eye tracking using a change-based convlstm network. In *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–5. IEEE, 2023. 1, 2, 8

[10] Hoonhee Cho, Hyeonseong Kim, Yujeong Chae, and Kuk-Jin Yoon. Label-free event-based object recognition via joint learning with image reconstruction from events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19866–19877, 2023. 2

[11] Peiqi Duan, Zihao W Wang, Xinyu Zhou, Yi Ma, and Boxin Shi. EventZoom: Learning to denoise and super resolve neuromorphic events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12824–12833, 2021. 5

[12] Andrew T Duchowski, Krzysztof Krejtz, Izabela Krejtz, Cezary Biele, Anna Niedzielska, Peter Kiefer, Martin Raubal, and Ioannis Giannopoulos. The index of pupillary activity: Measuring cognitive load vis-à-vis task difficulty with pupil oscillation. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018. 1

[13] Yu Feng, Nathan Goulding-Hotta, Asif Khan, Hans Reyserhove, and Yuhao Zhu. Real-time gaze tracking with event-driven eye segmentation. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 399–408. IEEE, 2022. 2

[14] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2020. 1, 2

[15] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. 3

[16] Steven Hickson, Nick Dufour, Avneesh Sud, Vivek Kwatra, and Irfan Essa. Eyemotion: Classifying facial expressions in vr using

eye-tracking cameras. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1626–1635. IEEE, 2019. 2, 6, 8, 7

[17] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1312–1321, 2021. 2, 6, 7

[18] Simon Klenk, David Bonello, Lukas Koestler, Nikita Araslanov, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2378–2388, 2024. 2

[19] Rakshit S Kothari, Aayush K Chaudhary, Reynold J Bailey, Jeff B Pelz, and Gabriel J Diaz. Ellseg: An ellipse segmentation framework for robust gaze tracking. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2757–2767, 2021. 6

[20] Patrick J Laub, Young Lee, and Thomas Taimre. *The elements of Hawkes processes*. Springer, 2021. 3

[21] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. Facial performance sensing head-mounted display. *ACM Transactions on Graphics (ToG)*, 34(4):1–9, 2015. 2

[22] Jiading Li, Zhiyu Zhu, Jinhui Hou, Junhui Hou, and Jinjian Wu. Denoising distillation makes event-frame transformers as accurate gaze trackers. *arXiv preprint arXiv:2404.00548*, 2024. 2

[23] Nealson Li, Muya Chang, and Arijit Raychowdhury. E-gaze: Gaze estimation with event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 6, 7, 1, 10

[24] Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, 64(12):3197–3234, 2022. 12

[25] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 934–943, 2021. 2

[26] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 $\times$ 128 120 db $15\mu s$ latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008. 2

[27] Sage L Matthews, Alvaro Uribe-Quevedo, and Alexander Theodorou. Rendering optimizations for virtual reality using eye-tracking. In *2020 22nd symposium on virtual and augmented reality (SVR)*, pages 398–405. IEEE, 2020. 1

[28] Nico Messikommer, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. *IEEE Robotics and Automation Letters*, 7(2):3515–3522, 2022. 2

[29] Meriem Ben Miled, Wenwen Liu, and Yuanchang Liu. Adaptive unsupervised learning-based 3D spatiotemporal filter for event-driven cameras. *Research*, 7:0330, 2024. 5

[30] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos. Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14414–14423, 2020. 2

[31] Yan Ru Pei, Sasskia Brüers, Sébastien Crouzet, Douglas McLelland, and Olivier Coenen. A lightweight spatiotemporal network for online eye tracking with event camera. *arXiv preprint arXiv:2404.08858*, 2024. 5, 6, 8, 1, 7

[32] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019. 2

[33] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. 2005. 6, 7

[34] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. AEGNN: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12371–12381, 2022. 2, 3

[35] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 156–163, 2020. 2

[36] Immo Schuetz and Katja Fiehler. Eye tracking in virtual reality: Vive pro eye spatial accuracy, precision, and calibration reliability. *Journal of Eye Movement Research*, 15(3), 2022. 6

[37] Argha Sen, Nuwan Sriyantha Bandara, Ila Gokarn, Thivya Kandappu, and Archan Misra. EyeTrAES: fine-grained, low-latency eye tracking via adaptive event slicing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4): 1–32, 2024. 1, 5, 10

[38] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023. 3

[39] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16155–16164, 2021. 6, 8

[40] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 608–619, 2021. 2

[41] Zuowen Wang, Chang Gao, Zongwei Wu, Marcos V Conde, Radu Timofte, Shih-Chii Liu, Qinyu Chen, Zheng-Jun Zha, Wei Zhai, Han Han, et al. Event-based eye tracking. ais 2024 challenge survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5810–5825, 2024. 1, 2, 5, 6, 7, 8

[42] Zhong Wang, Zengyu Wan, Han Han, Bohao Liao, Yuliang Wu, Wei Zhai, Yang Cao, and Zheng-jun Zha. Mambapupil: Bidirectional selective recurrent model for event-based eye tracking. *arXiv preprint arXiv:2404.12083*, 2024. 2, 5, 6, 8, 1, 7

[43] Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Event-based video reconstruction using transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2563–2572, 2021. 2

[44] Hao Wu, Jinghao Feng, Xuejin Tian, Edward Sun, Yunxin Liu, Bo Dong, Fengyuan Xu, and Sheng Zhong. EMO: Real-time emotion recognition from single-eye images for resource-constrained eyewear devices. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 448–461, 2020. 2, 6, 8, 7

[45] Ziyi Wu, Xudong Liu, and Igor Gilitschenski. Eventclip: Adapting clip for event-based object recognition. *arXiv preprint arXiv:2306.06354*, 2023. 2

[46] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10699–10709, 2023. 2

[47] Baoheng Zhang, Yizhao Gao, Jingyuan Li, and Hayden Kwok-Hay So. Co-designing a sub-millisecond latency event-based eye tracking system with submanifold sparse cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5779, 2024. 5, 6, 7

[48] Haiwei Zhang, Jiqing Zhang, Bo Dong, Pieter Peers, Wenwei Wu, Xiaopeng Wei, Felix Heide, and Xin Yang. In the blink of an eye: Event-based emotion recognition. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 1, 2, 6, 8, 7, 9

[49] Tongyu Zhang, Yiran Shen, Guangrong Zhao, Lin Wang, Xiaoming Chen, Lu Bai, and Yuanfeng Zhou. Swift-Eye: Towards anti-blink pupil tracking for precise and robust high-frequency near-eye movement analysis with event cameras. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 2, 1

[50] Guangrong Zhao, Yurun Yang, Jingwei Liu, Ning Chen, Yiran Shen, Hongkai Wen, and Guohao Lan. EV-Eye: Rethinking high-frequency eye tracking through the lenses of event cameras. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 5, 6, 7, 8, 11

[51] Zengqun Zhao and Qingshan Liu. Former-dfer: Dynamic facial expression recognition transformer. In *Proceedings of the 29th ACM international conference on multimedia*, pages 1553–1561, 2021. 2, 6, 8, 7

[52] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 2

[53] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras. In *2021 IEEE international conference on computational photography (ICCP)*, pages 1–11. IEEE, 2021. 2

[54] Zhiyu Zhu, Junhui Hou, and Xianqiang Lyu. Learning graph-embedded key-event back-tracing for object tracking in event clouds. *Advances in Neural Information Processing Systems*, 35:7462–7476, 2022. 2