Event-based 6-DoF object tracking with distance field reaching 130 Hz

Yufan Kang^{1,2}, Ryoichi Ishikawa^{2}, Guillaume Caron^{1,3}, Takeshi Oishi^{2}

***Abstract*—Event cameras report asynchronous, per-pixel brightness changes with microsecond latency, making them attractive for high-speed 6-DoF object tracking. We present EDFT, a real-time 6-DoF object tracker that uses only a monocular event camera and a 3D model. At the core is an *event-based distance field* (EDF) that endows sparse events with *two-sided* spatial gradients around object edges—overcoming the one-sided behavior of Time Surface maps—and a 3D–2D registration that aligns a sparse, model-derived point set to the EDF.**

We evaluate on the E-POSE benchmark and on the *5events16* dataset we introduce with this article. On 5events16 (UR10 arm providing ground truth poses, five objects, ten sequences plus 6 handheld sequences), EDFT tracks accurately all sequences successfully contrary to three previous state-of-the-art methods. On E-POSE where objects occupy only a small fraction of the field of view and the camera moves rapidly, EDFT still maintains start-to-finish tracking.

***Index Terms*—Visual Tracking, Computer Vision for Automation**

I. Introduction

SIX degrees of freedom (DoF) object tracking, which requires to estimate the 6 dimension pose between camera and target object, remains a main challenge in computer vision and robot perception. Although traditional frame-based cameras give enough information for accurate pose estimation in static case, they suffer from motion blur when the system is dynamic, especially when the camera or the object move in high speed. This motion blur leads to a sharp drop of estimation accuracy. What's more, the latency of image output in frame cameras seems too long in applications that require low latency and fast reaction. Therefore, frame cameras are facing great challenges against fast motion.

Event cameras are neuromorphic vision sensors that report asynchronous per-pixel intensity changes rather than full RGB frames [1]. Relative to conventional frame-based imagers, they offer markedly lower latency and power, while providing higher dynamic range and temporal resolution. These properties have already enabled agile flight control in robotics [2] and are being leveraged for autonomous driving [3], visual odometry [4], and 3D reconstruction [5]. When paired with complementary sensors, they further support 3D model–based tracking [6] and SLAM [7]. Community efforts to release large, well-curated datasets are accelerating their adoption [8].

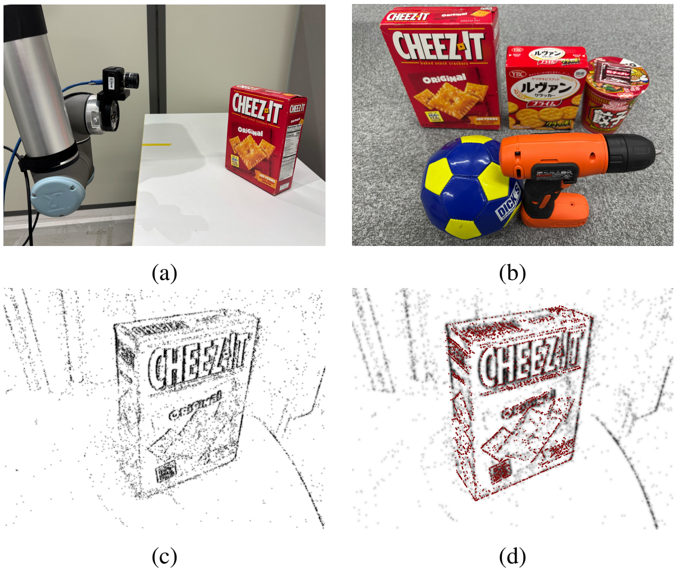

(a)
(b)
(c)
(d)

Fig. 1: (a) Experimental setting. (b) Objects used in the dataset. (c) Event-based distance field representation. (d) Reprojection from the point set to the event representation with estimated pose.

One main challenge in robot perception is 6-DoF object tracking, which requires to estimate the 6-dimension relative pose between robot and target object. Compared to frame-based camera, an event camera could capture object features clearly in high speed without motion blur. However, there comes new problems of processing event data in real-time while keeping high accuracy. Existing methods have taken strategies of 3D-2D registration [9], hypothesis selection [10], neural network [6] and probabilistic generative model [11]. Despite recent progress on event-based 6-DoF object tracking, [10] is the only method that reaches in real-time while it has only been evaluated on one real object. Qualitatively, the method is limited by its fixed pose variation per update and can fail in difficult conditions. Kang *et al.* [9] proposed BIAM, which is still the only work that includes quantitative results of real events, while it runs two orders of magnitudes slower than real-time requirements.

In this work, we propose a new real-time event-based 6-DoF

Manuscript received November 10, 2025; Revised January 29, 2026; Accepted February 22, 2026.

This paper was recommended for publication by Editor Abhinav Valada upon evaluation of the Associate Editor and Reviewers' comments.

[1] Yufan Kang and Guillaume Caron are with CNRS-AIST Joint Robotics Laboratory (JRL), National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba 305-8568, Japan kang-yufan@g.ecc.u-tokyo.jp.

[2] Yufan Kang, Ryoichi Ishikawa and Takeshi Oishi are with the Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan {ishikawa, oishi}@cvl.iis.u-tokyo.ac.jp.

[3] Guillaume Caron is also with Université de Picardie Jules Verne, MIS Laboratory, 80000 Amiens, France guillaume.caron@u-picardie.fr.

Digital Object Identifier (DOI): see top of this page.

object tracking method by 3D-2D registration with a new event representation. Specifically, we apply distance field to sparse events to generate a new event representation with natural spatial gradients. The optimization problem is defined by reprojecting a 3D point set extracted from the desired pose and 3D model to the new representation. Our system is capable of tracking several objects in real-time while keeping accuracy and robustness to noise. In summary, we make the following contributions in this paper:

- A novel and real-time event representation, Event-based Distance Field (EDF). The representation introduces a natural spatial gradient to sparse event data by applying a distance field, which is ideal for registration and pose estimation tasks and can be generated in real-time.
- A real-time event-based 6-DoF object tracking method. The method accurately estimates object pose by registering the object's 3D model with the proposed EDF representation.
- The first rigorous and quantitative evaluation of a real-time, event-based 6-DoF object tracking method. In contrast to prior real-time work without quantitative validation, we rigorously evaluate our method on a new dataset of real objects with ground-truth poses and on a public benchmark, demonstrating both accuracy and robustness. We will release the dataset as a new benchmark for event-based object tracking upon publication.

Outline. Section II surveys prior work on event-driven object and camera tracking. Section III details our tracking approach. Section IV reports quantitative and qualitative results on a new dataset. Section V summarizes our findings and outlines potential avenues for future research.

II. Related work

A. Event-based object Tracking

Event-driven 2D tracking aims to localize objects in the image plane from sparse, asynchronous brightness changes. Ramesh *et al.* [12] coupled detection with tracking and introduced a *local sliding window* strategy to remain reliable in cluttered, highly textured scenes. Along similar lines, Jiang *et al.* [13] paired an offline detector with an online tracker to follow multiple targets such as vehicles, exploiting the complementary strengths of both modules. Particle-filter formulations have also been explored—for example, tracking a moving ball even when the event camera itself is in motion [14]. Tang *et al.* [15] proposed an adaptive unified network for a large-scale and general RGB-Event tracking object tracking mission.

Estimating full 6-DoF poses typically leverages a 3D object model and aligns it to event data [6], [9]–[11]. Dubeau *et al.* [6] proposed a cascaded architecture that fuses a conventional RGB-D network with an event-specific module for high-speed pose tracking. Li and Stueckler [11] presented a probabilistic generative model that infers motion from high-rate events and refines trajectories via direct alignment to lower-rate frames. However, many such systems assume a static camera observing a moving object, so most events arise from the target itself—minimizing background-induced interference and simplifying data association.

The first real-time event-based 6-DoF object tracking system, EDOPT [10], evaluates a fixed set of motion hypotheses. While efficient, its design imposes a strict upper limit on the maximum trackable velocity—a deliberate trade-off to ensure real-time performance, since expanding the hypothesis space would exponentially increase computational cost. Failures may also arise from mismatches between the event representation and the projected model. Moreover, the authors reported only a single qualitative evaluation on a real object without quantitative metrics.

BIAM [9] addressed this by employing a 3D–2D registration between accumulated events and rendered brightness increment images, providing quantitative results on real objects and demonstrating remarkable accuracy. However, its computation speed is approximately two orders of magnitude slower than real-time, primarily because it utilizes nearly all object-related pixels. In contrast, we show in this article that leveraging only a sparse subset of informative pixels suffices for robust tracking.

B. Event-based camera tracking and visual odometry

Parallel to object tracking, substantial progress has been achieved in event-based camera tracking and visual odometry (VO), which estimate the camera's 6-DoF trajectory using events generated by the surrounding environment. These approaches often serve as the foundation for event-based SLAM systems and provide valuable insights into processing event data for motion estimation. Bryner *et al.* [16] introduced a direct alignment framework that estimates 6-DoF camera pose and velocity in scenes with an available photometric map. Their key idea is to compare two images of intensity variation: one formed by accumulating events, and one synthesized from the photometric map under a candidate motion. Building on this, Hidalgo-Carrió *et al.* [17] coupled the front end with a photometric bundle-adjustment back end to jointly reduce brightness residuals. To limit blur introduced during event-frame formation, they place greater weight on events occurring near the center of the temporal window. In their formulation, conventional intensity frames act as keyframes from which expected intensity changes are predicted under motion. ESVO [18] and its successor ESVO2 [19] addressed event-based visual odometry using the *Time Surface* representation, in which each pixel stores the timestamp of its most recent event. The implicit distance-field nature of the *Time Surface* enables efficient 3D–2D registration for camera tracking. However, this representation provides spatial gradients only on one side of edges, and the gradient quality can degrade depending on environmental texture and motion speed. Zhu *et al.* [20] introduced an adaptive decay rate related to scene complexity and motion speed to time surface for alleviating these motion-related issues. Zhong *et al.* [21] further enhanced ESVO with a novel and efficient static-stereo association strategy, achieving higher accuracy. Liu *et al.* [22] proposed a stereo event-based tracking method specifically for spacecraft applications, establishing event-line

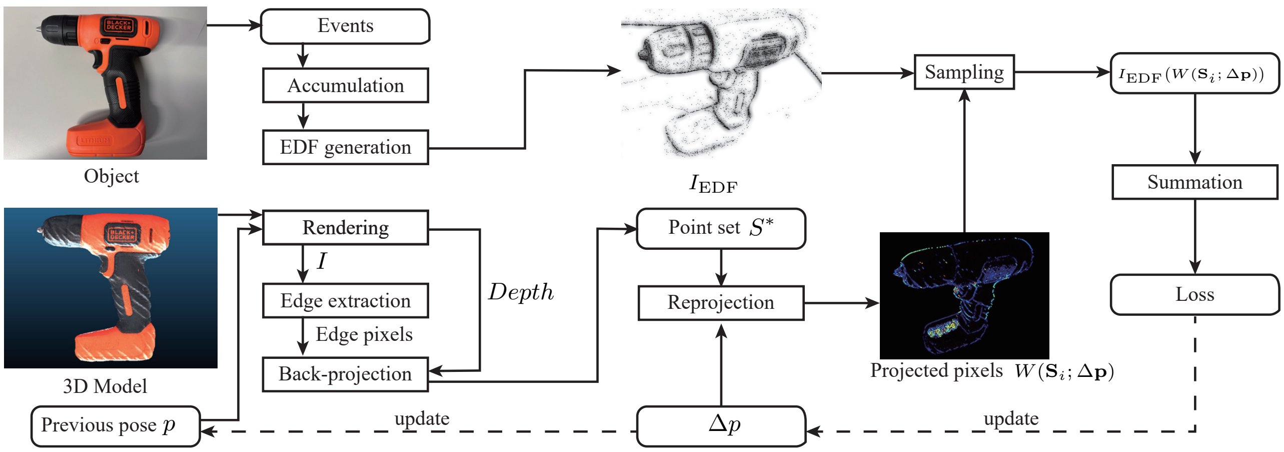


Fig. 2: Overview of EDFT. The pipeline consumes events and a 3D model, generates an I_{EDF} from accumulated events, renders a keyframe to select a sparse 3D point set S^*, and minimizes the EDF-based loss to update the 6-DoF pose. For visualization, gradients in I_{EDF} are enlarged, I_{EDF} and S^* are resized.

correspondences and minimizing event-line distances for robust 3D motion estimation. Lu *et al.* [23] proposed a map-free, real-time event-inertial estimator that, instead of recovering pose, focuses on instantaneous linear velocity; their dual-end pipeline (front-end normal flow + depth; back-end continuous-time sliding window) fuses stereo events with IMU to estimate metric-scale velocity and IMU bias under high-speed maneuvers, assuming sufficient texture.

Although some studies have explored 6-DoF event-based object tracking, only one reached real-time thanks to a small set of motion hypotheses. In this work, we address these limitations and evaluate our method on multiple objects.

III. Event-based 6-DoF object tracking with distance field

This section describes the novel event-based 6-DoF object tracking method based on a new event representation. We first review how event cameras work and present our Event-based Distance Field (EDF) representation. Then we present the 3D-2D registration method for pose estimation. Finally we introduce some techniques to enhance the tracking robustness and accuracy.

A. Event-based Distance Field (EDF)

An event camera asynchronously outputs a stream of events

$$\mathbf{e}_k = (\mathbf{u}_k, t_k, p_k) \tag{1}$$

where each event is triggered at pixel location $\mathbf{u}_k$ and time $t_k \in \mathbb{R}^+$ whenever the change in logarithmic brightness L reaches the contrast threshold C. Formally, an event is generated when

$$\delta I(\mathbf{u}_k, t_k) \doteq I(\mathbf{u}_k, t_k) - I(\mathbf{u}_k, t_k - \Delta t_k) = p_k C, \tag{2}$$

where $p_k \in \{+1, -1\}$ denotes the polarity (sign) of the brightness change, and Δt_k represents the time elapsed since the previous event at the same pixel.

Unlike the standard Time Surface, which yields a single-sided gradient, our distance-field event representation preserves two-sided spatial gradients across object edges. An event frame (EF) image is a binary 2D map indicating the spatial distribution of recent events, defined as

$$I_{EF}(\mathbf{u}) = \begin{cases} 1, & \text{if } \exists t_k \in T \text{ such that } \mathbf{u}_k = \mathbf{u}, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $T = t_1, t_2, \ldots, t_{N_e}$ and $N_e \in \mathbb{N}$ denotes the number of events accumulated. In practice, N_e is fixed to maintain a consistent event density.

At each pixel $\mathbf{u}$, the proposed kernel-based event distance field (EDF) image is computed as

$$I_{\text{EDF}}(\mathbf{u}) = 255 - \mathcal{N}_{[0,255]}\left[\sum_{\mathbf{u}_i \in \mathbf{U}} \left(k_e - D(\mathbf{u}, \mathbf{u}_i)\right)\right], \tag{4}$$

where $k_e \in \mathbb{R} > 0$ denotes the kernel size, and $D(\mathbf{u}, \mathbf{u}_i)$ is the Euclidean distance between pixels $\mathbf{u}$ and $\mathbf{u}_i$. Specifically, we employ a radially symmetric kernel with circular support of radius k_e, where the weight of each event decays linearly from its center. This design allows transforming sparse event data into a continuous and differentiable field, providing a smooth cost surface for gradient-based optimization. The set $\mathbf{U}$ contains all neighboring pixels for which $D(\mathbf{u}, \mathbf{u}_i) \leq k_e$ and $I_{EF}(\mathbf{u}_i) = 1$. Thus, nearby events collectively contribute to the I_{EDF} value at pixel $\mathbf{u}$, resulting in a smooth field that enhances edge continuity and gradient symmetry. In particular, the EDF behaves like a smoothed distance transform: it produces a natural, monotone fall-off normal to the edge and well-behaved, symmetric gradients on both sides.

In practice, certain regions of the event frame can become overly dense—such as areas where multiple edges are in close proximity—while other regions remain sparse. This imbalance causes excessive overlap of distance contributions in crowded areas and reduces spatial sensitivity in sparse areas. To address this issue, we take an adaptive strategy in generation of I_{EDF}. In the regions where too many events are detected, we assign a smaller k_e, whereas low-density regions are assigned a larger k_e to preserve edge sharpness and prevent excessive smoothing.

B. Object tracking based on EDF (EDFT)

We estimate the object pose by reprojecting a representation of its 3D model to I_{EDF}. The overview of our method is shown in Fig. 2.

We assume that the initial pose of the object $\mathbf{p}_0 \in \mathbb{R}^6$ is known. Given the 3D model and pose $\mathbf{p}_0$, a brightness image I and a corresponding depth image $Depth$ are rendered. A Sobel filter is applied to the brightness image I to extract a set of edge pixels, which are then back-projected into 3D space using the depth image to obtain the 3D point set

$$S^* = \{\mathbf{S}_i \mid \mathbf{S}_i \in \mathbb{R}^3\}, \tag{5}$$

where each element $\mathbf{S}_i = [X_i, Y_i, Z_i]^\top$ represents a 3D coordinate on the object surface. These points correspond to the locations where new events are expected to occur under object motion. The sparsity of S^* reduces the overall computational cost.

We observe that edges play a dominant role in event generation, as events are primarily triggered by brightness changes along object boundaries. To facilitate reliable edge extraction and ensure that the rendered brightness image provides clear edge information, we enhance the color contrast of the 3D models prior to rendering. This adjustment amplifies intensity differences across adjacent surfaces, thereby producing sharper brightness gradients that lead to more distinct and stable event edges.

To estimate the pose variation $\Delta\mathbf{p} \in \mathbb{R}^6$ between consecutive frames, we formulate a 3D–2D registration problem using a warping function as

$$\Delta\mathbf{p} = \arg\min_{\Delta\mathbf{p}} \sum_{\mathbf{S}_i \in S^*} I_{\mathrm{EDF}}\big(W(\mathbf{S}_i; \Delta\mathbf{p})\big), \tag{6}$$

where the warping function $W(\cdot)$ transforms each 3D point $\mathbf{S}_i$ from the reference frame to the current frame and projects it onto the image plane:

$$W(\mathbf{S}_i; \Delta\mathbf{p}) \doteq \pi(T(\mathbf{S}_i, G(\Delta\mathbf{p}))), \tag{7}$$

where $T(\mathbf{S}_i, G(\Delta\mathbf{p}))$ applies the rigid-body transformation parameterized by $\Delta\mathbf{p}$, and $\pi(\cdot)$ denotes the camera projection function. The mapping

$$G : \mathbb{R}^6 \to SE(3) \tag{8}$$

computes the homogeneous transformation matrix corresponding to the 6-DoF pose variation $\Delta\mathbf{p}$.

To solve the optimization problem in (6), we employ the Levenberg-Marquardt (LM) algorithm. The analytical Jacobian $\mathbf{J}$ is computed via the chain rule as follows:

$$\mathbf{J}_{\mathbf{u}_i} = \frac{\partial I_{\mathrm{EDF}}}{\partial \mathbf{u}_i} \cdot \frac{\partial \mathbf{u}_i}{\partial \mathbf{S}_i} \cdot \frac{\partial \mathbf{S}_i}{\partial \Delta\mathbf{p}} \tag{9}$$

where $\mathbf{u}_i = W(\mathbf{S}_i; \Delta\mathbf{p})$ is the projected pixel location, and $\mathbf{S}_i$ represents the 3D coordinate of a point transformed into the current camera frame. Specifically, $\frac{\partial I_{\mathrm{EDF}}}{\partial \mathbf{u}_i}$ is the spatial gradient of the EDF, $\frac{\partial \mathbf{u}_i}{\partial \mathbf{S}_i}$ is the camera projection Jacobian [16], and $\frac{\partial \mathbf{S}_i}{\partial \Delta\mathbf{p}}$ is the motion Jacobian parameterized using Cayley coordinates [24] to ensure singularity-free optimization. The optimization terminates when either the relative change in the cost function falls below 1×10^{-3} or a maximum of 10 iterations is reached.

If the point set S^* is rendered only once at the beginning, the discrepancy between the projected points in S^* and the actual event distribution will gradually increase as the object or camera moves. To mitigate this drift and maintain alignment accuracy without increasing too much the processing time, we adopt a *conditional keyframe generation* strategy. Specifically, whenever the estimated pose variation exceeds a predefined threshold, a new keyframe is rendered from the current pose p, and a corresponding updated point set S^* is generated.

IV. Experiments

We evaluate the proposed method using real event-camera data. This section first introduces the datasets and experimental settings, followed by a quantitative and qualitative comparison against state-of-the-art approaches.

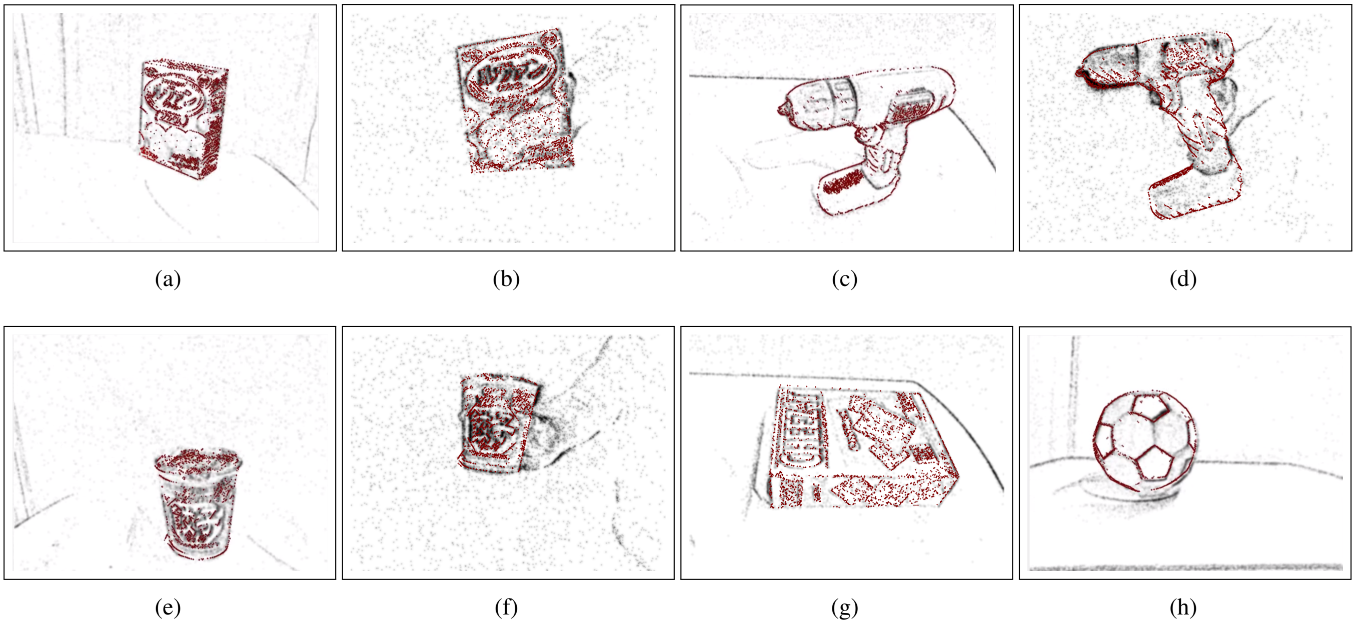
A. Datasets and Experimental Settings

Two datasets are used for evaluation of our EDFT method.

First, we introduce the new dataset 5events16 featuring a UR10 robot and five different objects, a cup noodle, a cheezit cracker box, a drill, a football, and a cookie box. Several objects are adopted from the Yale-CMU-Berkeley (YCB) dataset [25], while others are common household items. Each object is represented by a dense 3D point cloud. The models for the Cheezit and Cookie boxes were constructed in minutes with CAD and applying texture mapping from smartphone photographs. Other models were sourced from the public YCB dataset.

5events16 comprises 16 sequences captured using a Prophesee Gen 3.1 event camera (640 × 480 pixels). 10 sequences were recorded with the camera mounted on a UR10 robotic arm to provide accurate ground-truth poses via hand-eye calibration. Among these, five trajectories are pure translations, while the others feature combined 6-DoF motions with peak speeds of 0.1–$0.3\,\mathrm{m/s}$. To evaluate the tracking capability under more challenging conditions, we supplement the dataset with 6 hand-held sequences: one aggressive 6-DoF trajectory for each of the five objects, and an additional sequence for the Drill object in a dark environment. These trajectories exhibit significantly higher velocities than using the robot arm, with peak speeds of $0.7\,\mathrm{m/s}$, angular excursions exceeding 60° and substantial depth variations. Camera intrinsics were calibrated using a 9×6 blinking checkerboard (16.6 mm grid size) [24], achieving a mean reprojection error of 0.236 pixels ($f_x, f_y, c_x, c_y = 566.4, 567.7, 310.8, 200.5$). An overview of the objects and experimental setup is shown in Fig. 1.

We further evaluate on the recently released E-POSE dataset [26], a YCB-based benchmark for 6-DoF object pose estimation. The sequences were recorded with a UR10 robot


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3: Qualitative results of object tracking with EDFT on 5events16 dataset. The images depict the points in S^* (in red) reprojected from point set to I_{EDF} (in grayscale). (a), (c), (e), (g), (h) are captured with the camera on the robot arm wheras (b), (d) and (f) are captured with the object moved quickly manually.

arm moving a DAVIS346c event camera (resolution 346 × 260 pixels) while observing static objects. Unlike 5events16, E-POSE objects occupy only a small fraction of the field of view, which substantially increases tracking difficulty (lower event density and weaker edge gradients). We report results on four representative sequences—wood block, wrench, nine-hole peg, and Rubik's cube—captured under good lighting with a camera speed of 1 m/s.

We set the fixed number of accumulated events to $N_e = 10,000$ for the 5events16 dataset and $N_e = 2,000$ for the E-POSE dataset, reflecting the different camera resolutions. The threshold τ for Sobel filter is 40. The kernel size k_e ranges from 4 to 10, depending on event density.

B. Evaluation Metrics

Following the protocol of [16], we report errors on translation and rotation variation related to the initial camera pose. The translation error is the Euclidean distance between the estimated and ground-truth positions. The rotation error is computed as the geodesic distance on $\mathrm{SO}(3)$, i.e., the minimal angle required to align the two orientations. Unlike [16], which summarizes results primarily with median position and rotation errors, we additionally report root-mean-square (RMS) position and rotation errors, as done in several recent works like [9], to capture the influence of outliers.

Because the object pose in the robot frame is not directly observable, we evaluate camera motion instead. Specifically, we form the relative transform from the initial camera frame c_0 to the current camera frame c as

$$ {}^{c}\mathbf{M}_{c_0} = {}^{c}\mathbf{M}_b \left({}^{c_0}\mathbf{M}_b\right)^{-1}, \tag{10}$$

where ${}^{c}\mathbf{M}_b$ and ${}^{c_0}\mathbf{M}_b$ denote the transforms from the robot base frame b to the camera frames c and c_0, respectively. We then extract the relative translation and rotation from ${}^{c}\mathbf{M}_{c_0}$ and use these components to compute the above position and orientation errors.

C. Comparison to the State-of-the-Art

We compare our EDFT method against two state-of-the-art approaches. The first is BIAM [9], which performs 3D–2D alignment between accumulated events and rendered brightness increment images. To our knowledge, it is the only event-based 6-DoF object tracking method evaluated on real objects with quantitative results. BIAM achieves high accuracy but at the cost of very low computational efficiency—according to the authors, it operates at least two orders of magnitude slower than real time. The second method is EDOPT [10], the first event-based object tracking framework that achieves real-time performance. All three methods use only events and 3D models as input.

In our experiments, EDOPT could sustain tracking for roughly 1–2 s on most sequences captured with robot (except for Cheezit. 2 and Cookie. 2) but with large pose error, after which it typically diverged. On Drill. 2 for example, the average position and rotation errors were $20.1\,\mathrm{mm}$ and $8.49°$, respectively during 0-2 s, then the tracking failed. The errors are even larger on objects with more complicated textures (cup noodle and cookie box). On Cookie. 1, the average position error was $28.3\,\mathrm{mm}$ and the average rotation error was $12.36°$ before the tracking failed in 1.4 s.

BIAM successfully tracks eight sequences, whereas EDFT achieves stable tracking in the ten sequences captured with the robot of 5events16 dataset (Table I). In terms of accuracy, the two methods are complementary: BIAM attains lower RMS error on several trajectories—most notably in rotation—while EDFT is generally more accurate in translation and exhibits higher overall robustness across motion patterns. This

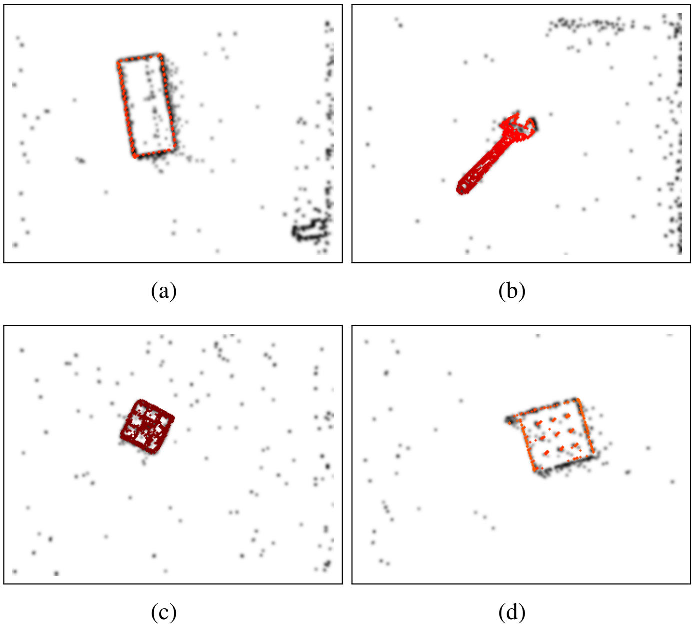


Fig. 4: Qualitative results of object tracking with EDFT on E-POSE dataset. EDFT performs well even though only 100-300 points can be used in S^*.

performance gap in rotation is primarily attributed to BIAM’s motion interpolation strategy, which facilitates a more effective decoupling of translational and rotational components during registration. In practice, EDFT prioritizes a sparse sampling strategy to achieve much faster performance. Fig. 6 compares ground truth and EDFT estimates on Cheezit. 1 and Drill. 2 sequences, illustrating reliable performance under both pure translation and full 6-DoF motion.

Fig. 3 shows qualitative results on our new dataset: the reprojected 3D points S^* align closely with the I_{EDF} images throughout the trajectories. EDFT remains robust to background activity and, even when shadows induce occasional events near object boundaries, it continues to track accurately. Aggressive hand-held experiments with rotations up to 60° and rapid Z-axis translations further demonstrate EDFT’s robustness. Quantitative analysis of these sequences confirms that the system maintains stable tracking even when up to 12% of the object’s visible surface is occluded by the hand. Despite high velocities, partial occlusions and even low-light condition, the wide attraction basin of our kernel-based EDF ensures reliable 6-DoF convergence during optimization as the accompanying video illustrates.

On the E-POSE benchmark (Fig. 4), EDFT likewise tracks from start to finish despite the objects occupying only a small portion of the field of view. Across most frames, the reprojected points coincide well with I_{EDF}; only a small number of frames exhibit noticeable deviations, after which the tracker quickly recovers and maintains lock. These deviations arise from the limited number of reliable points in S^* available for tracking. On E-POSE dataset, only about 100–300 points are typically usable.

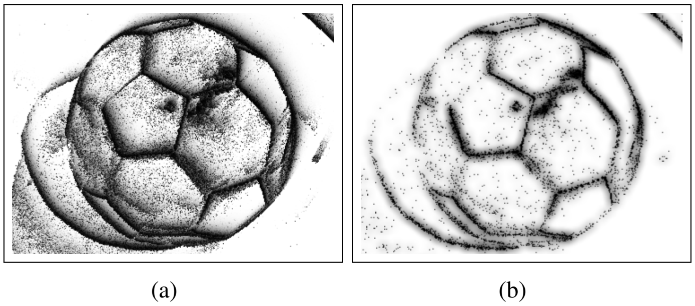


Fig. 5: Comparison between TimeSurface (a) and EDF (b). The gradients are enlarged for visualization.

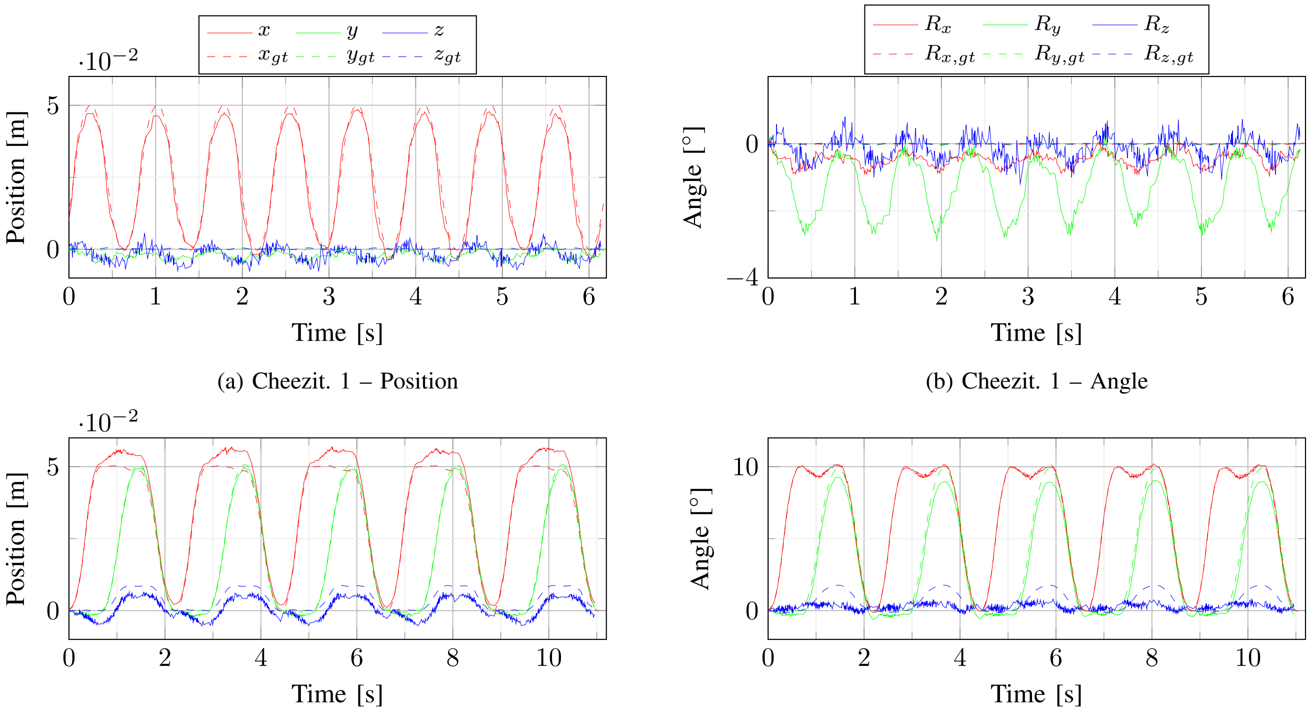
D. Ablation Study

For a fair comparison, we swept the TimeSurface decay constant from 10 to 50 μs—a range chosen to balance gradient decay and ’tail’ length [18]. Within this interval, 20 μs yielded the best tracking performance. Quantitative results (Tab. I) show TS [18] tracked only 5 sequences, while EDFT maintained stability across all cases with lower RMS errors. EDF’s symmetric, two-sided gradients (Fig. 5(b)) provide a wider attraction basin than TS’s one-sided (Fig. 5(a)), motion-dependent gradients. By accumulating a fixed number of events (N_e), EDF also remains robust to noise from ”stale” events. Furthermore, our adaptive kernel reduced translation and rotation errors by 1.3 mm and 0.45° out of 9.98 mm and 2.03° compared to fixed kernels, particularly on non-uniform textures like Cookie where it balances local smoothing and sensitivity.

E. Computational Performance

EDFT is implemented in C++ on ROS and processes recorded event streams in real time. The whole system runs on a laptop equipped with an Intel Core i7-1280P CPU and 32 GB of RAM, with Ubuntu 20.04 LTS as the operating system and ROS Noetic for communication. Notably, the entire EDFT pipeline is executed solely on the CPU without requiring GPU acceleration. The system comprises two threads: an *EDF generator* and a *pose estimator*. The EDF generator takes 2.5 ms to generate one frame from batches of 10,000 events. The pose estimator averages 7.6 ms per frame ($\approx$ 131 Hz) on 5events16 dataset using 3000 points in S^* for accurate tracking. On E-POSE dataset, the estimator takes less than 3 ms per frame as the point number is much smaller. Because the two modules execute in parallel, the end-to-end throughput is bounded by the slower stage. At comparable accuracy, this is roughly 200× faster than BIAM.

While events are generated asynchronously, the reported 131 Hz correspond to the maximum fixed frame rate at which EDFT maintains reliable tracking across all the sequences. In scenarios with high event rates, the system maintains real-time responsiveness by prioritizing the most recent N_e events and managing throughput based on object complexity. The rate may decrease during slow motion or low-light conditions. However, the resulting latency is acceptable as pose variations remain minimal during these intervals, ensuring previous estimates remain valid.



(a) Cheezit. 1 – Position

(b) Cheezit. 1 – Angle

(c) Drill. 2 – Position

(d) Drill. 2 – Angle

Fig. 6: Object pose tracking results of EDFT with position and rotation variations on Cheezit. 1 and Drill. 2 in 5events16 dataset. The X, Y, and Z axes are shown as red, green, and blue curves (dashed: ground truth, solid: estimated).

F. Robustness to initialization errors

We assess convergence under controlled perturbations of an estimated initial pose. Translations are swept in $[-10, 10]$ mm along each axis and rotations in $[-10°, 10°]$ about each axis. For every perturbed pose, we keep the same point set S^* and apply the rigid transform $T(\Delta)$ directly to S^* in the camera frame, then reproject with subpixel accuracy using the known intrinsics. The registration cost is the EDF-based objective defined in (6). For each perturbation Δ, we evaluate $C(\Delta)$ and plot the normalized change relative to the unperturbed pose,

$$\Delta C(\Delta) = 100 \cdot \frac{C(\Delta) - C(\mathbf{0})}{C(\mathbf{0})} \,\%,$$

where $C(\mathbf{0})$ denotes the cost at the initial pose. Fig. 7 shows $\Delta C(\Delta)$ when varying one degree of freedom at a time on Drill object.

a) Findings: Across sequences, the cost curves exhibit a clear basin with a minimum at the unperturbed pose. Translational perturbations along x and y show the strongest sensitivity (monotonic rise away from zero in most cases), while T_z produces a gentler slope, consistent with projective scaling: small changes in depth have a reduced effect on image reprojection when the object is relatively distant. Rotations yield smooth, convex-like profiles around zero, with steeper growth for R_x/R_y on elongated or textured objects and milder growth for R_z on near-symmetric or repetitively textured geometry.

b) Convergence region: Within $\|\Delta\mathbf{t}\| \leq 10$ mm and $\|\Delta\mathbf{r}\| \leq 5°$, the EDF-based registration reliably converges on most sequences. Beyond these bounds, convergence becomes less consistent because the reprojected points move outside the support of the distance kernel used in our EDF representation. In this regime, the distance values saturate and the gradient vanishes, flattening the cost surface and reducing the optimizer's ability to recover.

TABLE I: Comparison among TimeSurface, BIAM, and EDFT on 10 sequences with ground truth (RMS position [mm] / rotation [°]) in 5events16 dataset. Bold font indicates best results per sequence.

Sequence	TimeSurface	BIAM	EDFT
Cheezit. 1	39.2 / 4.87	6.6 / **0.74**	**4.4** / 1.32
Cheezit. 2	/	6.7 / 1.35	**5.1** / **0.89**
Football. 1	18.4 / 2.93	9.3 / **1.22**	**8.8** / 1.97
Football. 2	30.0 / 4.32	**7.8** / **0.96**	10.6 / 1.45
Cupnoodle. 1	/	**8.6** / **1.37**	9.4 / 1.88
Cupnoodle. 2	/	/	**10.6** / **1.43**
Drill. 1	21.1 / 3.13	13.2 / **1.46**	**6.6** / 1.69
Drill. 2	35.4 / 3.72	**9.3** / **1.09**	10.1 / 1.95
Cookie. 1	/	13.4 / **1.22**	**12.3** / 1.88
Cookie. 2	/	/	**8.9** / **2.35**

c) Practical implication: These results indicate that the method tolerates coarse initializations (e.g., detector outputs or coarse SLAM priors) without fine pose seeding. In practice, a lightweight pre-alignment (bringing errors within 10 mm and 5°) is sufficient to land inside the attraction basin of the EDF objective.

V. CONCLUSION

This paper presented EDFT, a real-time 6-DoF object tracking framework that operates using only a monocular event camera and a 3D object model. The proposed

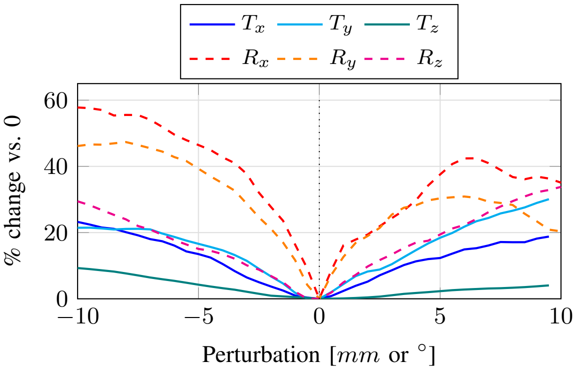


Fig. 7: Normalized cost sensitivity evaluation. The cost function $C(\Delta)$ shows a clear basin with a minimum at the unperturbed pose (dotted vertical line) across all six degrees of freedom. Translations (T_x, T_y, T_z) are shown as blue-toned solid lines, while rotations (R_x, R_y, R_z) are red-toned dashed lines. T_z displays a gentler slope due to projective scaling effects.

event-based distance field (EDF) representation preserves balanced spatial gradients around object edges, enabling stable and accurate tracking under both pure translational and general 6-DoF motions. We utilize the spatial gradient of EDF and estimate 6-DoF object pose with a 3D-2D registration between 3D model and the representation. Through a parallel implementation of the EDF generator and pose estimation modules, EDFT achieves real-time performance reaching 131 Hz. Comprehensive experiments on multiple objects and trajectories demonstrate that EDFT achieves comparable accuracy to the state-of-the-art BIAM method while operating 200 times faster to reach real-time. Unlike BIAM, which tracks only a subset of sequences, and EDOPT, which fails under large pose variations and background noise, EDFT successfully tracks all evaluated trajectories, showing superior robustness and general applicability. The accompanying 5events16 dataset, containing high-quality event data and ground-truth poses, will be publicly released to serve as a new benchmark for real-time event-based object tracking. Future works will be dedicated to relax the need of precise 3D object models toward more partial models and category-level templates.

References

[1] G. Gallego, T. Delbrück, G. Orchard, *et al.*, "Event-based vision: A survey," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[2] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, 2020.

[3] Y. Hu, J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction," in *Proc. of IEEE Int. Conf. on Intelligent Transportation Systems*, pp. 1–6, 2020.

[4] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5816–5824, 2017.

[5] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *Proc. of European Conf. on Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 349–364, 2016.

[6] E. Dubeau, M. Garon, B. Debaque, R. d. Charette, and J.-F. Lalonde, "RGB-D-E: Event camera calibration for fast 6-DoF object tracking," in *Proc. of IEEE Int. Symposium on Mixed and Augmented Reality*, pp. 127–135, 2020.

[7] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[8] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "TUM-VIE: The TUM stereo visual-inertial event dataset," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2021.

[9] Y. Kang, G. Caron, R. Ishikawa, *et al.*, "Direct 3d model-based object tracking with event camera by motion interpolation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2645–2651, 2024.

[10] A. Glover, L. Gava, Z. Li, and C. Bartolozzi, "EDOPT: Event-camera 6-DoF dynamic object pose tracking," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 18 200–18 206, 2024.

[11] H. Li and J. Stueckler, "Tracking 6-dof object motion from events and frames," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 14 171–14 177, 2021.

[12] B. Ramesh, S. Zhang, Z. W. Lee, Z. Gao, G. Orchard, and C. Xiang, "Long-term object tracking with a moving event camera.," in *the British Machine Vision Conf.*, p. 241, 2018.

[13] R. Jiang, X. Mou, S. Shi, *et al.*, "Object tracking on event cameras with offline–online learning," *CAAI Trans on Intelligence Technology*, vol. 5, no. 3, pp. 165–171, 2020.

[14] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3769–3776, 2017.

[15] C. Tang, X. Wang, J. Huang, *et al.*, "Revisiting color-event based tracking: A unified network, dataset, and metric," *Pattern Recognition*, p. 112 718, 2025.

[16] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, "Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 325–331, 2019.

[17] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, "Event-aided direct sparse odometry," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 5781–5790, 2022.

[18] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans on Robotics*, vol. 37, no. 5, pp. 1433–1450, 2021.

[19] J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou, "ESVO2: Direct visual-inertial odometry with stereo event cameras," *IEEE Trans on Robotics*, 2025.

[20] S. Zhu, Z. Tang, M. Yang, E. Learned-Miller, and D. Kim, "Event camera-based visual odometry for dynamic motion tracking of a legged robot using adaptive time surface," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3475–3482, 2023.

[21] S. Zhong, J. Niu, and Y. Zhou, "Deep visual odometry for stereo event cameras," *IEEE Robotics and Automation Letters*, 2025.

[22] Z. Liu, B. Guan, Y. Shang, Y. Bian, P. Sun, and Q. Yu, "Stereo event-based, 6-DoF pose tracking for uncooperative spacecraft," *IEEE Trans on Geoscience and Remote Sensing*, 2025.

[23] X. Lu, Y. Zhou, J. Mai, K. Dai, Y. Xu, and S. Shen, "Event-based visual-inertial state estimation for high-speed maneuvers," *IEEE Trans on Robotics*, 2025.

[24] P. Lébraly, C. Deymier, O. Ait-Aider, E. Royer, and M. Dhome, "Flexible extrinsic calibration of non-overlapping cameras using a planar mirror: Application to vision-based robotics," in *2010 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, pp. 5640–5647, 2010.

[25] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. of IEEE Int. Conf. on Advanced Robotics*, pp. 510–517, 2015.

[26] O. A. Hay, X. Huang, A. Ayyad, *et al.*, "E-POSE: A large scale event camera dataset for object pose estimation," *Scientific Data*, vol. 12, no. 1, p. 245, 2025.