Event6D: Event-based Novel Object 6D Pose Tracking

Supplementary Material

In this supplemental document, we provide additional details about our datasets and the EventTrack6D method. Specifically, we provide

- Details of the introduced EventBlender6D, EventHO3D, and Event6D datasets in Sections 1, 2, and 3.
- Details of the object assets and evaluation protocol in Section 4.
- Implementation details of the proposed method and other methods in Section 5.
- Experiments on additional datasets and methods, along with further analyses, qualitative results, and video demonstrations, in Section 6.

1. EventBlender6D Dataset

EventBlender6D is a synthetic benchmark for 6D object pose estimation in dynamic scenarios, constructed at three difficulty levels (easy, medium, hard) according to the number of objects present in each scene. The easy setting contains single-object scenes with 1,033 sequences, whereas the medium setting includes 2–4 objects per scene and 2,066 sequences featuring collisions and mutual occlusions. The hard setting further increases the complexity to 5–10 objects per scene with 1,033 sequences. Each sequence comprises 120 frames recorded at 60 fps, resulting in 2-second clips that capture the full evolution of the scene, from initial object placement to free fall under gravity and eventual rest.

The dataset uses Google Scanned Objects (GSO) [4] with a balanced sampling strategy that ensures uniform representation across all models. Each object is assigned randomized material properties, including surface roughness and specular reflectance values between 0 and 1.0. Objects are initialized at random positions and orientations within the workspace, with collision checking to ensure valid starting configurations. The physics simulation uses realistic parameters with mass, friction coefficient, and damping values for stable dynamics.

Object motion is governed by realistic gravitational physics, where objects fall naturally, undergo collisions in multi-object scenes, and settle on the floor following physically-based dynamics. Camera motion follows a hemispherical orbital trajectory with azimuthal rotation completing 2.0 to 3.5 full revolutions per sequence, while elevation angles are constrained between 5° and 85°. The orbital radius is adaptively determined based on object bounding boxes, with scaling factors of 1.2–1.5 for easy mode and 1.5–2.0 for medium mode. Throughout the sequence, the camera continuously tracks a dynamically updated point of interest positioned at the median location of all objects, ensuring that the workspace remains centered in the field of view as objects descend under gravity.

To generate event data, we follow the protocol of video2events [7]. We first upsample the video frame rate using the method [20] described in their pipeline, and then synthesize events using ESIM [18]. Following prior work [10, 14], we additionally adapt the generated events by applying random contrast sensitivities sampled from $\mathcal{U}(0.16, 0.34)$.

Dataset samples are provided in Fig. 5, and since the EventBlender6D data are rendered, the ground-truth 6D object poses are highly accurate.

2. EventHO3D Dataset

For the HO3D dataset [11], which consists of real-world markerless RGB-D hand–object interactions with 3D hand poses and 6D object poses obtained via sequence-level joint optimization, we generate event data using the same pipeline as EventBlender6D. Through this process, we construct the EventHO3D dataset. Note that EventHO3D is used only to assess the model's generalization capability under diverse conditions, and none of its data are used for training. Examples from the EventHO3D dataset are illustrated in Fig. 6.

3. Event6D Dataset

To acquire the Event6D dataset, we used three primary sensing systems: an RGB-D camera, an event camera, and an OptiTrack motion-capture system for providing ground-truth poses. To reliably collect data from these heterogeneous sensors, two key procedures are required: cross-system calibration to align their coordinate frames, and time synchronization to ensure that all systems share a consistent temporal reference.

3.1. Calibration

3.1.1. Camera Parameter Calibration

Event cameras are inherently sparse and asynchronous, which makes their standalone calibration already challenging. Calibrating them jointly with conventional cameras is even more difficult. To address this, following prior works, we convert event streams into dense, temporally aligned images using a pretrained event-to-image reconstruction model. As shown in Fig. 1, we reconstruct intensity images from the raw events using E2VID [19], and perform calibration on these reconstructed frames. For the calibration toolbox, we adopt Kalibr [6], which is robust to noisy

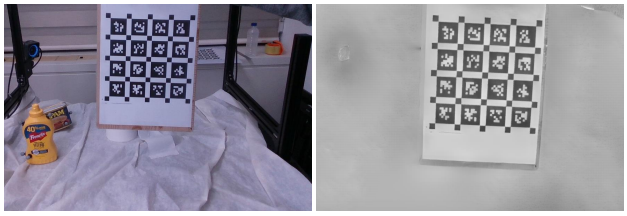

Figure 1. Examples of the data used for camera calibration.

measurements and allows us to obtain both the intrinsic and extrinsic parameters of each camera. Through this process, we obtain depth that is aligned with the event camera.

3.1.2. Hand-Eye Calibration

Our objective is to estimate the 6D pose of each object in the camera coordinate frame. However, the OptiTrack motion-capture system provides measurements in its own world coordinate frame, which makes cross-system alignment essential. To bridge this gap, we estimate the transformation from the OptiTrack world frame to the camera coordinate frame by directly aligning the 2D observations in the camera images with the corresponding 3D points measured by the OptiTrack system. Specifically, we formulate the problem as a direct 2D–3D registration and solve it through a robust nonlinear optimization procedure. This allows us to accurately map the OptiTrack world frame onto the camera coordinate frame and ensures that all subsequent 6D pose annotations are expressed consistently in the camera's reference system.

Coordinate Frames. As shown in Fig. 2, we denote the OptiTrack (motion-capture) world coordinate frame by O and the camera's optical frame by C. At each timestamp t_m, the OptiTrack system provides the 3D positions of the checkerboard corners $\mathbf{P}_n^O(t_m)$, where n indexes individual checkerboard corners. The camera simultaneously observes the same corners in the image plane, yielding the corresponding 2D measurements $\mathbf{x}_{mn}$.

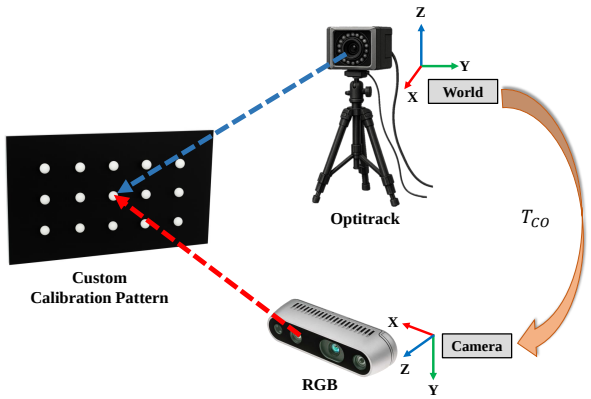

Figure 2. Illustration of Hand-eye calibration. We denote the OptiTrack (motion-capture) world coordinate frame as O and the camera's optical frame as C. The transformation from the OptiTrack frame to the camera frame is represented by T_{CO}.

2D-3D Optimization. Our goal is to estimate the transformation from the OptiTrack world frame to the camera frame,

$$T_{CO} = \begin{bmatrix} R_{CO} & \mathbf{t}_{CO} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{1}$$

where R_{CO} and $\mathbf{t}_{CO}$ denote rotation and translation, respectively. Given camera intrinsics K, the predicted image projection of a 3D point is

$$\hat{\mathbf{x}}_{mn} = \pi\left(K\, T_{CO}\, \mathbf{P}_n^O(t_m)\right), \tag{2}$$

where $\pi(\cdot)$ denotes the perspective projection function. We estimate T_{CO} by minimizing the total reprojection error:

$$\min_{R_{CO},\mathbf{t}_{CO}} \sum_{m,n} \left\|\mathbf{x}_{mn} - \hat{\mathbf{x}}_{mn}(R_{CO},\mathbf{t}_{CO})\right\|^2. \tag{3}$$

This non-linear least-squares problem is solved via Levenberg-Marquardt algorithm.

RANSAC-based Outlier Rejection. To handle noisy 2D detections from the camera coordinate, we adopt a RANSAC scheme before the final refinement. At each iteration, a minimal subset of 2D-3D correspondences is sampled to compute a candidate pose $\hat{T}_{CO}$. The remaining correspondences are tested for inlier support:

$$\left\|\mathbf{x}_{mn} - \hat{\mathbf{x}}_{mn}(\hat{T}_{CO})\right\| < \tau, \tag{4}$$

where τ is a reprojection error threshold. The hypothesis with the largest inlier set is retained, and the final pose estimate is obtained by solving (3) using only the inliers.

Ground-truth 6D Object Pose. The resulting optimized transformation T_{CO} directly represents the camera pose with respect to the OptiTrack world coordinate frame. Since the object poses provided by OptiTrack are expressed in the OptiTrack world frame, we first transform them into the camera coordinate frame using T_{CO}. However, the resulting object pose centers are not perfectly aligned with the true centers of the corresponding CAD models. To address this, we obtain an initial 6D object pose by combining FoundationPose [22] with masks generated by the Segment Anything Model [13], and then manually refine this pose. We subsequently convert only this refined pose into the OptiTrack coordinate frame and use it as the ground-truth annotation.

3.2. Trigger System

To ensure that all data is captured in the same precise time domain, we employed a hardware trigger system to synchronize the acquisition times. The RGB-D camera used

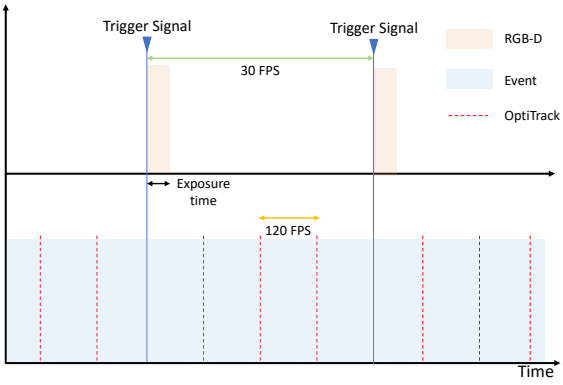

Figure 3. Visualization of trigger signals for overall system.

in our setup, the RealSense D435i, can internally generate external trigger signals at 30 FPS. These signals are then received and processed by both the event camera and the OptiTrack system. As illustrated in Fig. 3, the RGB-D camera captures data at 30 FPS and simultaneously outputs a trigger signal. Based on this trigger, the event camera can segment its event stream into slices, and the OptiTrack system can align its ground-truth acquisition with the same timing. Furthermore, OptiTrack can subdivide each external trigger interval into smaller segments using its internal multiplier, enabling ground-truth capture at 120 FPS, which is four times faster than the RGB-D camera rate.

3.3. Dataset Details

We acquired the Event6D dataset such that each object exhibits dynamic, challenging, yet realistic motions. To this end, we designed the motions by imagining typical real-world usage of each object and mimicking the kinds of movements that would naturally occur. In our experiments, we only use the Event6D dataset as a test set and do not use the training split at all. However, Event6D differs from existing datasets in two major aspects: (i) it includes challenging and highly dynamic object motions, and (ii) it provides highly accurate ground-truth poses together with event and depth data. These aspects underscore the strengths of our Event6D dataset. Consequently, we also collected a training split to facilitate future research. Detailed descriptions of the training and test sequences of the proposed Event6D dataset are provided in Table 3 and Table 4, respectively, and representative dataset samples are illustrated in Fig. 7.

4. Object Assets and Novel Object Evaluation

Object Assets. For object assets, EventBlender6D consists of Google Scan Objects (GSO) [4], which provides 1033 high-quality 3D scanned models with realistic textures. The Event6D dataset consists of a subset of HOGrasp [3] and Yale-CMU-Berkeley (YCB) dataset [2], while HO3D consists of a subset of the YCB [2] dataset, as shown in Fig. 4.

Figure 4. The object assets used in the Event6D dataset. The object assets do not overlap between EventBlender6D (used for training), ensuring proper novel-object testing.

For novel object pose estimation evaluation, we ensure that the training and test sets are strictly disjoint. Specifically, the objects used in EventBlender6D (training) do not overlap with those in Event6D and EventHO3D (test), enabling rigorous evaluation of generalization to unseen objects. In total, our dataset comprises 1047 unique object instances across diverse categories, including household items, tools, and objects relevant to manipulation.

Object Instance Split for Train and Test. To evaluate novel object pose estimation capabilities, we maintain strict separation between training and evaluation objects. The 1033 GSO objects in EventBlender6D serve as the training set, while Event6D and EventHO3D provide test scenarios with completely unseen objects from HOGrasp and YCB datasets. This split ensures that models cannot rely on object-specific priors learned during training and must generalize to novel geometric and appearance characteristics.

CAD Model Acquisition. CAD models for GSO objects are directly obtained from the official repository with their provided high-quality meshes. For YCB objects, we use the standardized CAD models from the official YCB Object and Model Set. HOGrasp object meshes are either obtained from the original dataset or reconstructed using structure-from-motion techniques when high-quality CAD models are unavailable. All meshes are preprocessed to ensure consistent coordinate frames, metric scale, and watertight geometry for physics simulation and rendering. Fig. 4 shows representative object assets from the Event6D dataset, illustrating the diversity of geometric complexity and visual appearance in our evaluation benchmark.

5. Implementation Details

5.1. EventTrack6D

For the event representation used in dual-modal reconstruction, we adopt a voxel grid [8, 19, 23] with a bin size of 5 for both the image and depth modalities. For training, we use two NVIDIA RTX A6000 GPUs and adopt a modular training strategy to improve stability. To effectively leverage prior knowledge learned from existing datasets, we initialize the image reconstruction module from a pretrained checkpoint [19] and similarly initialize the refiner using a pretrained model [22]. Specifically, we first train the dual-modal reconstruction as separate modules, with the image reconstruction module frozen, and then fine-tune the entire pipeline in an end-to-end manner, except for the LSTM parameters in the image reconstruction module, which are not further trained since they are not designed for sequential data. We train our model using only the easy difficulty level of the EventBlender6D dataset, which already provide sufficient complexity and diversity for robust generalization across various scenarios.

5.2. Event-based Baselines

5.2.1. Implementation Details of Each Model

E2VID + MegaPose (MG). MegaPose (MG) [15] performs pose tracking on RGB or RGB-D images. We bridge the gap between event streams and image-based tracking by converting events to intensity images using E2VID [19]. We use the pretrained MegaPose checkpoint, which has been trained on a diverse collection of datasets.

E2VID + FoundationPose (FP). FoundationPose (FP) [22] is a state-of-the-art RGB-D pose tracking method. Following the original implementation, we set the number of iterations in the FP pose refiner to 2. To enable event-based tracking, we employ E2VID [19] to reconstruct intensity images from event streams. These reconstructed images are fed into FP's RGB-D tracking pipeline, serving as our baseline configuration. We utilize the FP checkpoint that has been pretrained on a large and diverse collection of datasets.

ETAP. We consider a hybrid formulation that combines the pre-trained event-based point tracking [10] with a rigid transformation–based update. Given the pose estimate from the previous timestamp, $\mathbf{T}_{t-\Delta t}$, we assume that the CAD model of the object is placed at the estimated pose, and then n points are uniformly sampled from its 3D surface. These 3D points, $\mathbf{X}_{t-\Delta t} = \{\mathbf{X}^i = (x^i, y^i, z^i)\}_{i=1}^n$ are then projected onto the current event frame, resulting 2D pixel coordinates $\mathbf{x}_{t-\Delta t} = \{\mathbf{x}^i = K\mathbf{X}^i\}_{i=1}^n$, where K represents a projection matrix of the event camera. We track the projected points using the event-based point tracker ETAP [10], and denote the tracked 2D points as $\tilde{\mathbf{x}}_t$. Depending on the availability of depth measurements, the final pose is computed using either a PnP [16] formulation or an ICP refinement [1].

At time steps where depth measurements are available, the 3D coordinate of a tracked 2D point can be recovered from the depth map. Its depth is obtained by sampling the depth map D_t at the tracked pixel location: $d_t^i = D_t(u_t^i, v_t^i)$. The 3D coordinate is then computed by back-projection:

$$\mathbf{X}_t^i = d_t^i K^{-1} \tilde{\mathbf{x}}_t^i, \tag{5}$$

where $\tilde{\mathbf{x}}_t^i = [u_t^i, v_t^i, 1]^\top$ is the tracked 2D point in homogeneous coordinates and K denotes the camera intrinsic matrix. We align the previous 3D point cloud with the current observation using an ICP-based registration step.

$$\Delta T_{t-\Delta t,t} = \mathrm{ICP}(\mathbf{X}_{t-\Delta t}, \mathbf{X}_t) \tag{6}$$

$$T_t = \Delta T_{t-\Delta t,t} T_{t-\Delta t} \tag{7}$$

During time steps where no depth measurement is available, typically the interval between two consecutive depth inputs, we apply a PnP-based 2D–3D matching between the current 2D points and previously observed 3D points.

$$T_t = \mathrm{P}n\mathrm{P}(\mathbf{X}_{t-\Delta t}, \tilde{\mathbf{x}}_t) \tag{8}$$

Event-based FoundationPose. Since there are no existing learning-based event-driven methods that generalize well to novel objects, we train an adapted version of FoundationPose (FP) [22] that takes event data as input to serve as a strong event-based baseline. We build the training pipeline on top of the publicly available official implementation of FP, modifying the input interface to accept event voxel grids instead of RGB images. The network is initialized from the pretrained FP checkpoint, and training is carried out on the EventBlender6D dataset, using the same setup as for our method.

5.2.2. Experimental Details at 120 FPS

Unlike RGB-based models, the event-based baselines can perform inference at a higher temporal resolution, operating at 120 FPS as in our main experiments, rather than being limited to the 30 FPS of the depth stream. For MegaPose, FoundationPose, and our event-based adaptation of FP, we observe that they can still run without depth by masking the depth input with zeros. Based on this, we feed reconstructed images from E2VID in intervals where depth is not available and provide the depth input at timestamps where depth measurements are present. For the ETAP baseline, we use ICP-based tracking when depth is available, and fall back to a PnP-based pose update when only reconstructed image information is present.

6. Additional Experiments

6.1. Experiments on Other Datasets

In addition to EventHO3D and Event6D, we further evaluate the trained model on the YCB-Ev dataset [21], com-

Table 1. Experiments on the YCB-Ev dataset.

Methods	FP [22]	**EventTrack6D (Ours)**
Modality	RGB+Depth	Event+Depth
AR ↑	5.82	17.87

paring our proposed EventTrack6D with the RGB+Depth-based FoundationPose (FP). Since the ground-truth (GT) annotations in YCB-Ev are generated using an RGB-D–based method, they can exhibit temporal inconsistencies within some sequences. To mitigate this issue, we exclude such sequences from our evaluation. As shown in Table 1, our method achieves higher quantitative scores than FP. However, we believe that these gains may not solely be attributed to the merits of our approach, but are also influenced by the fact that YCB-Ev does not employ hardware-level triggering, which can lead to misalignment between the different sensor streams. For this reason, we do not include the YCB-Ev results in the main paper and instead report them here for completeness.

We also considered conducting additional experiments on E-POSE [12] and RGB-DE [5], but were unable to do so because full public access to the necessary data is currently not available. In contrast, our Event6D dataset provides accurate ground-truth annotations using a motion capture system and ensures precise time synchronization across different modalities at the hardware level. This design highlights the reliability of Event6D as a benchmark, and we plan to maintain and release it as a well-curated public resource for the community.

6.2. Initialization Sensitivity

Our method is designed to recover from rotation errors up to 20° and translation errors up to half the object diameter. Table 2 evaluates robustness to first-frame pose errors by applying Δ rotation and translation perturbations. Our method remains reliable within approximately 10° and 10 cm.

Table 2. Performance changes resulting from adding errors to the first-frame pose. $\Delta 0^\circ$ and $\Delta 0$ cm indicate that no error was added.

$\Delta 0^\circ$ & $\Delta 0$ cm		$\Delta 3^\circ$ & $\Delta 3$ cm		$\Delta 5^\circ$ & $\Delta 5$ cm		$\Delta 10^\circ$ & $\Delta 10$ cm		$\Delta 15^\circ$ & $\Delta 15$ cm	
ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD
52.79	25.26	50.74	23.75	51.30	23.69	50.08	23.38	5.19	1.81

6.3. Experiments with Other Existing Methods

We first clarify that our task focuses on *Novel Object 6D Pose Tracking*, where the model must track previously unseen objects during inference. Methods that rely on instance-level training do not fall within this scope. For example, RGB-D-E [5] methods are typically trained on a specific object instance and therefore do not exhibit the level of object generalization required for our setting. LOPET [17] also presents challenges for our evaluation protocol. The method assumes line-based geometric priors and requires the target object to consist predominantly of linear structures. As illustrated in Fig. 4, many objects in our benchmark have curved or complex geometries, making it difficult to apply LOPET in a principled way. In addition, LOPET requires an initial line specification, which cannot be reliably provided for curved objects. Although EDOPT [9] is not learning-based, it is capable of handling unseen objects and represents a valuable feature-based approach. However, our Event6D dataset contains objects moving at an average speed of 2,m/s, and, as noted in the official implementation, EDOPT is sensitive to rapid motion. In our experiments, the tracker quickly diverged under this dynamic setting, and we were therefore unable to obtain stable results suitable for reporting.

Given these considerations, we include the event-based FoundationPose [22] as a comparison method. FoundationPose has recently demonstrated strong generalization capabilities across novel objects and diverse scenarios. To ensure a fair and meaningful comparison within our event-based framework, we train an event-driven version of FoundationPose and use it as a competitive baseline in our evaluation.

6.4. Qualitative Results

We provide additional qualitative results comparing the proposed EventTrack6D with several strong baselines: the state-of-the-art RGB-D tracker FoundationPose (FP) [22], an E2VID [19] + FP pipeline, and an event-adapted variant of FP that operates on event and depth inputs. As can be seen in Figures 8 and 9, RGB-D-based FP quickly loses track of the object once the motion becomes large and highly dynamic. Moreover, using only E2VID for image reconstruction still provides insufficient geometric information, E2VID + FP often leading to additional tracking failures. The event-adapted FP variant is able to roughly follow the object motion, but it struggles to estimate the correct object scale and frequently produces inaccurate boundaries. In contrast, the proposed EventTrack6D accurately recovers the object pose even under highly dynamic motion, where RGB-D-based approaches face significant challenges. These results highlight that our method offers robust 6D tracking for novel objects in event-driven settings, providing a strong foundational baseline for future research on event-based object pose tracking.

6.5. Video Demo in Dynamic Motion

We additionally provide qualitative video results extracted from the test set to demonstrate that our method operates reliably over the time dimension. The accompanying demo includes highly dynamic and extreme motions in realistic scenes. Furthermore, we present additional videos recorded without the motion-capture system's IR markers to show-

case performance in even more realistic and challenging scenarios. As can be seen in these videos, the proposed method remains stable even under such extreme conditions, whereas RGB-D–based approaches often struggle to maintain accurate tracking.

References

[1] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 4

[2] 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 *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 3

[3] W. Cho, J. Lee, M. Yi, M. Kim, T. Woo, D. Kim, T. Ha, H. Lee, J.-H. Ryu, W. Woo, et al. Dense hand-object (ho) graspnet with full grasping taxonomy and dynamics. In *European Conference on Computer Vision*, pages 284–303. Springer, 2024. 3

[4] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. Ieee, 2022. 1, 3

[5] E. Dubeau, M. Garon, B. Debaque, R. de Charette, and J.-F. Lalonde. Rgb-de: Event camera calibration for fast 6-dof object tracking. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–135. IEEE, 2020. 5

[6] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286. IEEE, 2013. 1

[7] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3586–3595, 2020. 1

[8] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021. 4

[9] A. Glover, L. Gava, Z. Li, and C. Bartolozzi. Edopt: Event-camera 6-dof dynamic object pose tracking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 18200–18206. IEEE, 2024. 5

[10] F. Hamann, D. Gehrig, F. Febryanto, K. Daniilidis, and G. Gallego. Etap: Event-based tracking of any point. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27186–27196, 2025. 1, 4

[11] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020. 1

[12] O. A. Hay, X. Huang, A. Ayyad, E. Sherif, R. Almadhoun, Y. Abdulrahman, L. Seneviratne, A. Abusafieh, and Y. Zweiri. E-pose: A large scale event camera dataset for object pose estimation. *Scientific data*, 12(1):245, 2025. 5

[13] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2

[14] S. Klenk, M. Motzet, L. Koestler, and D. Cremers. Deep event visual odometry. In *2024 International conference on 3D vision (3DV)*, pages 739–749. IEEE, 2024. 1

[15] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022. 4

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the p n p problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 4

[17] Z. Liu, B. Guan, Y. Shang, Q. Yu, and L. Kneip. Line-based 6-dof object pose estimation and tracking with an event camera. *IEEE Transactions on Image Processing*, 33:4765–4780, 2024. 5

[18] H. Rebecq, D. Gehrig, and D. Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 1

[19] H. Rebecq, R. Ranftl, V. Koltun, and D. 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. 1, 4, 5

[20] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, and B. Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision*, pages 250–266. Springer, 2022. 1

[21] P. Rojtberg and T. Pöllabauer. Ycb-ev 1.1: Event-vision dataset for 6dof object pose estimation. In *European Conference on Computer Vision*, pages 1–13. Springer, 2024. 4

[22] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: unified 6d pose estimation and tracking of novel objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868—-17879, 2024. doi: 10.1109/cvpr52733.2024.01692. 2, 4, 5

[23] A. Z. Zhu, L. Yuan, K. Chaney, and K. 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. 4

Table 3. An overview of the proposed Event6D training set, which is released for future research and not used for training in our experiments. *No. Frames* denotes the number of 30 FPS RGB and depth frames; the 6D poses are provided at 120 FPS.

Sequence Name	No. Frames	Description
Train Sequences		
banana_001	351	The banana is moved by applying a translation at an appropriate speed.
banana_002	294	The banana is dynamically moved in all directions and orientations across 6-DoF.
banana_003	320	The banana is rapidly moved while varying its depth.
banana_004	291	The banana is moved rapidly by applying both rotation and translation.
bowl_001	193	The bowl is moved rapidly with translation and then rotated to include diverse motions.
bowl_002	168	The bowl is rapidly translated in multiple directions while its depth is quickly varied.
bowl_003	166	The bowl is rapidly rotated.
clamp_001	244	The clamp is rapidly rotated at various angles to include motion across all axes.
clamp_002	328	The clamp undergoes rapid combined rotational and translational motion as it is thrown and caught.
cracker_001	266	The cracker box is first translated rapidly and then rotated to enrich its motion.
cracker_002	226	The cracker box exhibits strong, rapid rotation, with translation occurring simultaneously.
cracker_003	142	The cracker box undergoes a throw-and-catch motion with rapidly and continuously varying depth.
cracker_004	162	The cracker box is repeatedly passed between both hands to generate dynamic motion.
drill_001	493	The drill undergoes rapid movement across diverse motion patterns.
drill_002	408	The drill is manipulated with abrupt, forceful movements, mimicking real drilling on various objects.
drill_003	164	The drill, placed among many objects, performs rotation-heavy motions that mimic drilling.
hammer_001	119	The hammer is repeatedly rotated by 180° and driven through large translational motion.
marker_001	201	The marker is held in hand and moved rapidly with translational motion.
marker_002	160	The marker is held in hand and moved with larger, faster translational motion.
mouse_001	257	The mouse used for computers is held in hand and moved rapidly in various directions.
mug_001	205	The mug is moved rapidly with combined rotation and translation.
mug_002	178	The mug undergoes fast rotation and rapid motion, including collisions with another cup in a cheers gesture.
mustard_001	215	The mustard case undergoes rapid shaking and is translated over bowls.
pitcher_001	289	The pitcher is moved with rapid rotation, intermittently passed back and forth between both hands.
pitcher_002	291	The pitcher is used to rapidly pour water into multiple cups and bowls.
pitcher_003	493	The pitcher is used to rapidly pour water into multiple cups and bowls.
pudding_001	151	The pudding box undergoes extremely fast rotation while being moved.
pudding_002	270	The pudding box is spun at very high speed while being moved and placed on multiple bowls.
pudding_003	364	The pudding box is placed inside a bowl and shaken rapidly.
pudding_004	271	The pudding box moves with very fast translation and rotation, intermittently switching the holding hand.
scrub_001	271	The scrub cleanser bottle is rapidly translated and used to dispense cleanser onto multiple bottles.
scrub_002	261	The scrub cleanser bottle is held by its end and rotated widely with occasional hand switching.
spam_001	661	The spam can undergoes repeated rotations with varying depth, while the holding hand is switched.
spam_002	221	The spam can shows translation-dominant motion with repeated hand-to-hand throwing.
spatula_001	224	The spatula starts with fast motion and then performs cooking-like rotations perpendicular to the plane.
spatula_002	261	The spatula is driven quickly in a shaking motion, as if mixing something.
wine_001	190	The wine glass is moved dynamically with rotation-dominant motion at various angles, then placed on several bowls.
Total	9,769	

Table 4. An overview of the proposed Event6D test set. *No. Frames* denotes the number of 30 FPS RGB and depth frames; the 6D poses are provided at 120 FPS.

Sequence Name	No. Frames	Description
Test Sequences		
banana	301	The banana is held by its stem and moved dynamically with both translation and rotation around that axis.
bowl	261	The bowl is moved with varying depth and rotated to reveal diverse viewpoints.
cracker	308	The cracker box is rotated through various angles and exchanged between both hands.
drill	431	The drill is moved quickly in a fixing-like action, performed at multiple orientations with repeated 180° angle changes.
hammer	246	The hammer rapidly executes smashing motions as if breaking an object.
marker	146	The marker is rapidly moved with translation-dominant motion.
mouse	192	The mouse is held in hand and rapidly moved with rotation.
mug	347	The mug is grasped at the top and driven through wide and varied rotations.
mustard	196	The mustard bottle is tossed between both hands and moved back and forth over several bowls.
pitcher	562	The pitcher is rapidly rotated in one hand and then thrown and caught between both hands.
scrub	276	The scrub cleanser bottle undergoes multi-angle rotation while being moved with varying depth.
spam	204	The spam can undergoes dynamic 6-DoF movement involving both translation and rotation.
spatula	263	The spatula moves rapidly and includes stirring or flipping motions as in real cooking.
wine	137	The wine glass is rotated around the camera's z-axis while undergoing depth variation.
Total	3,870	

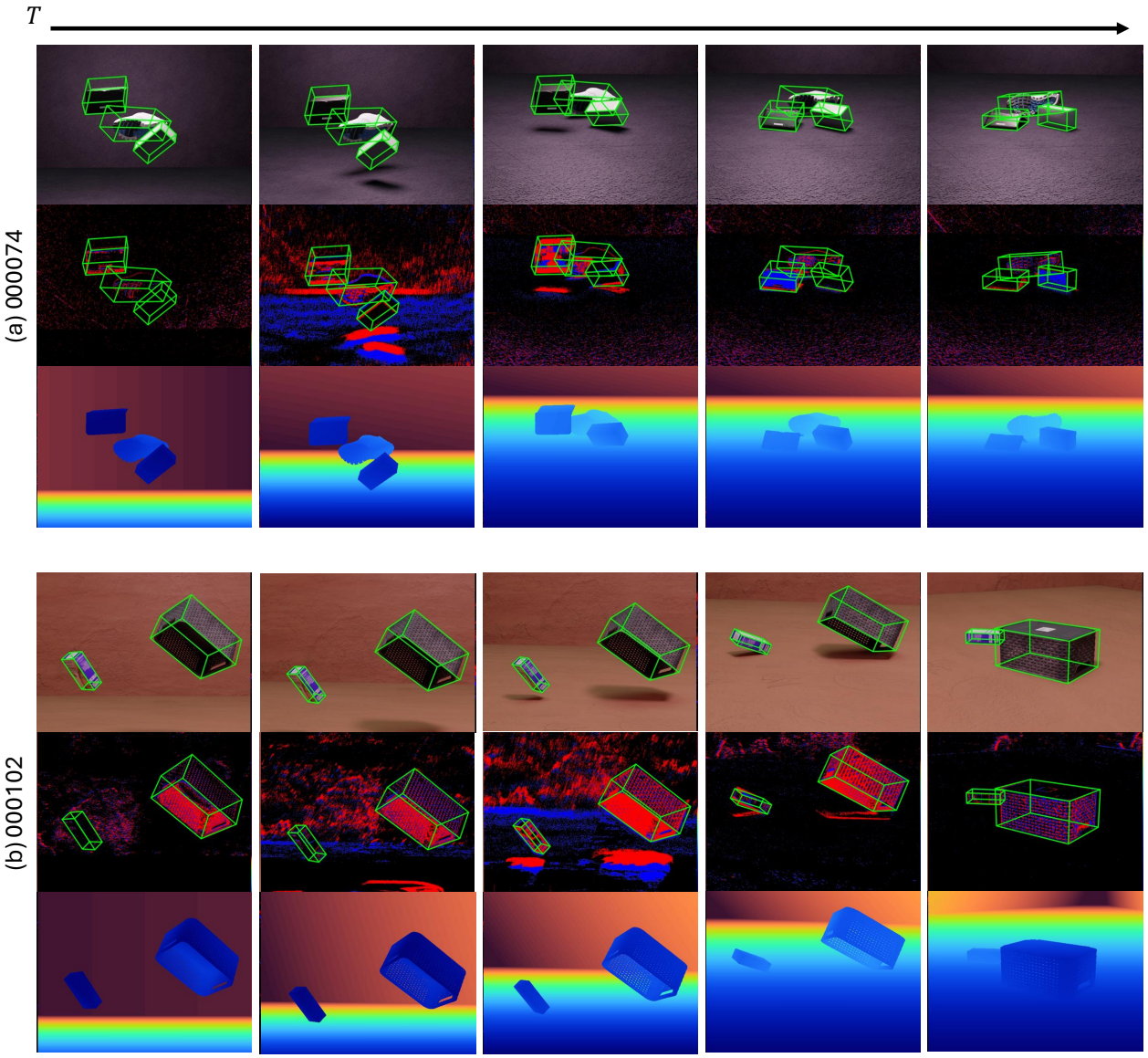

Figure 5. EventBlender6D samples visualized as temporal streams of RGB, event, depth, and corresponding 6D object poses.

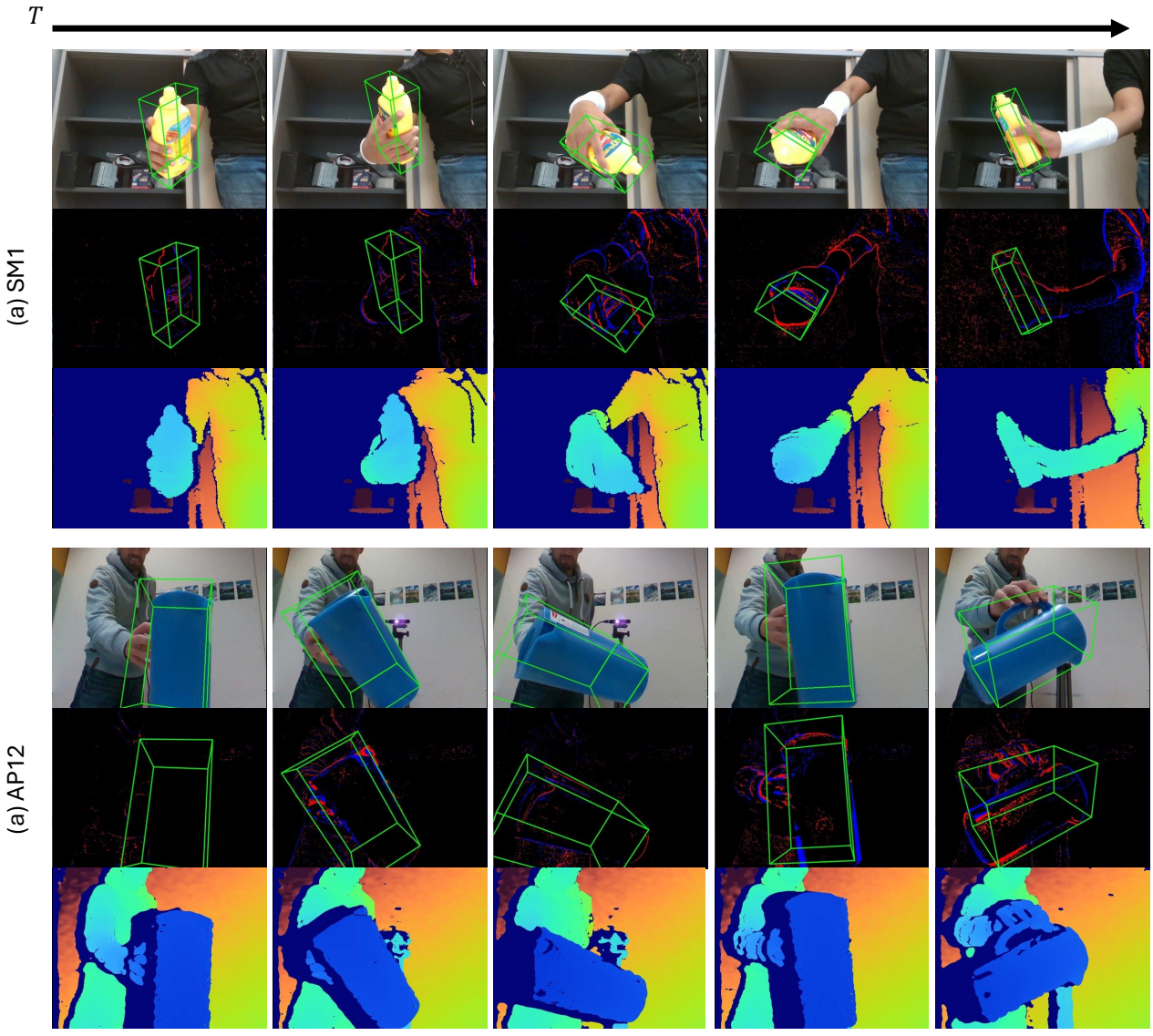

Figure 6. EventHO3D samples visualized as temporal streams of RGB, event, depth, and corresponding 6D object poses.

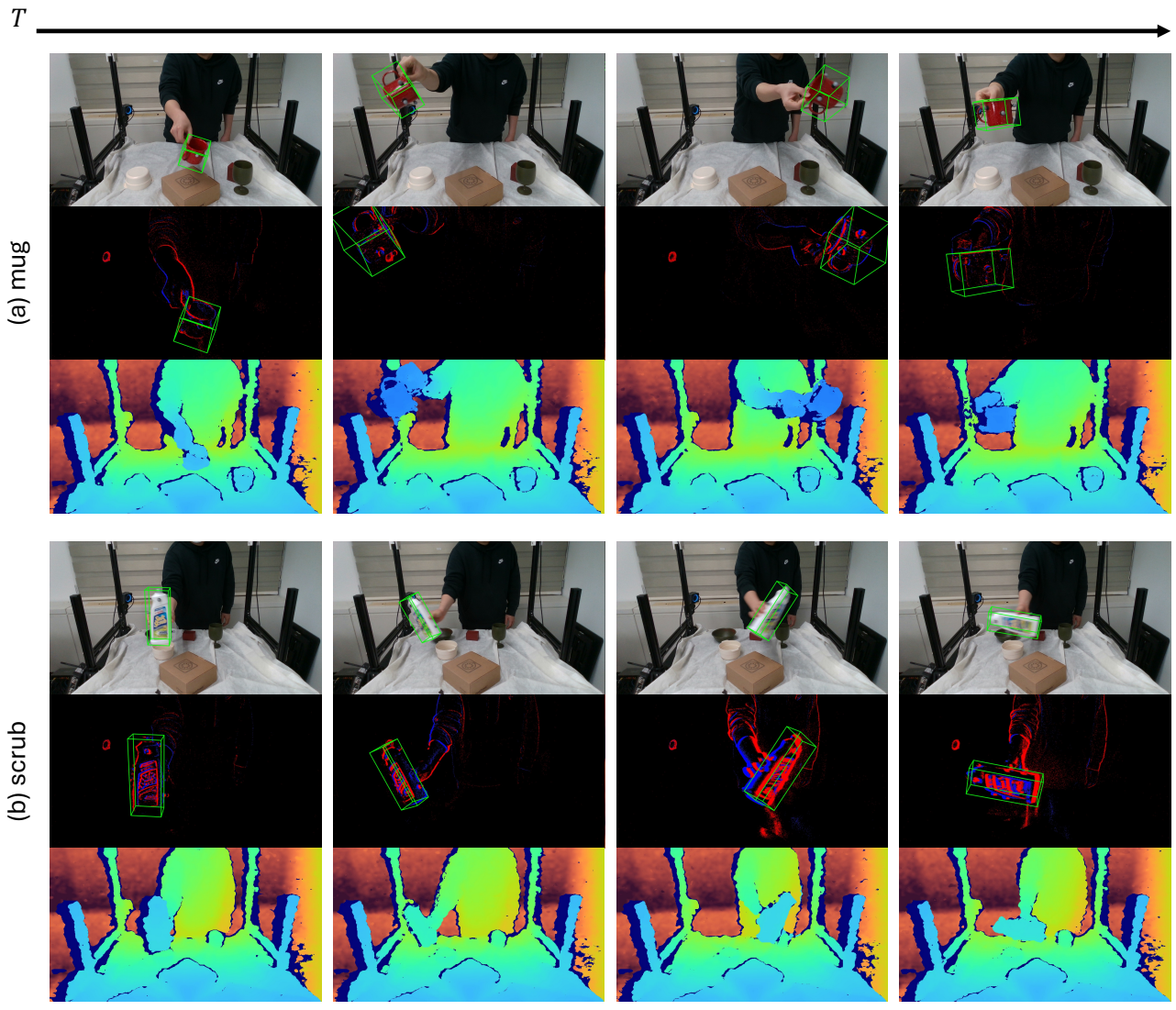

Figure 7. Event6D test samples visualized as temporal streams of RGB, event, depth, and corresponding 6D object poses.

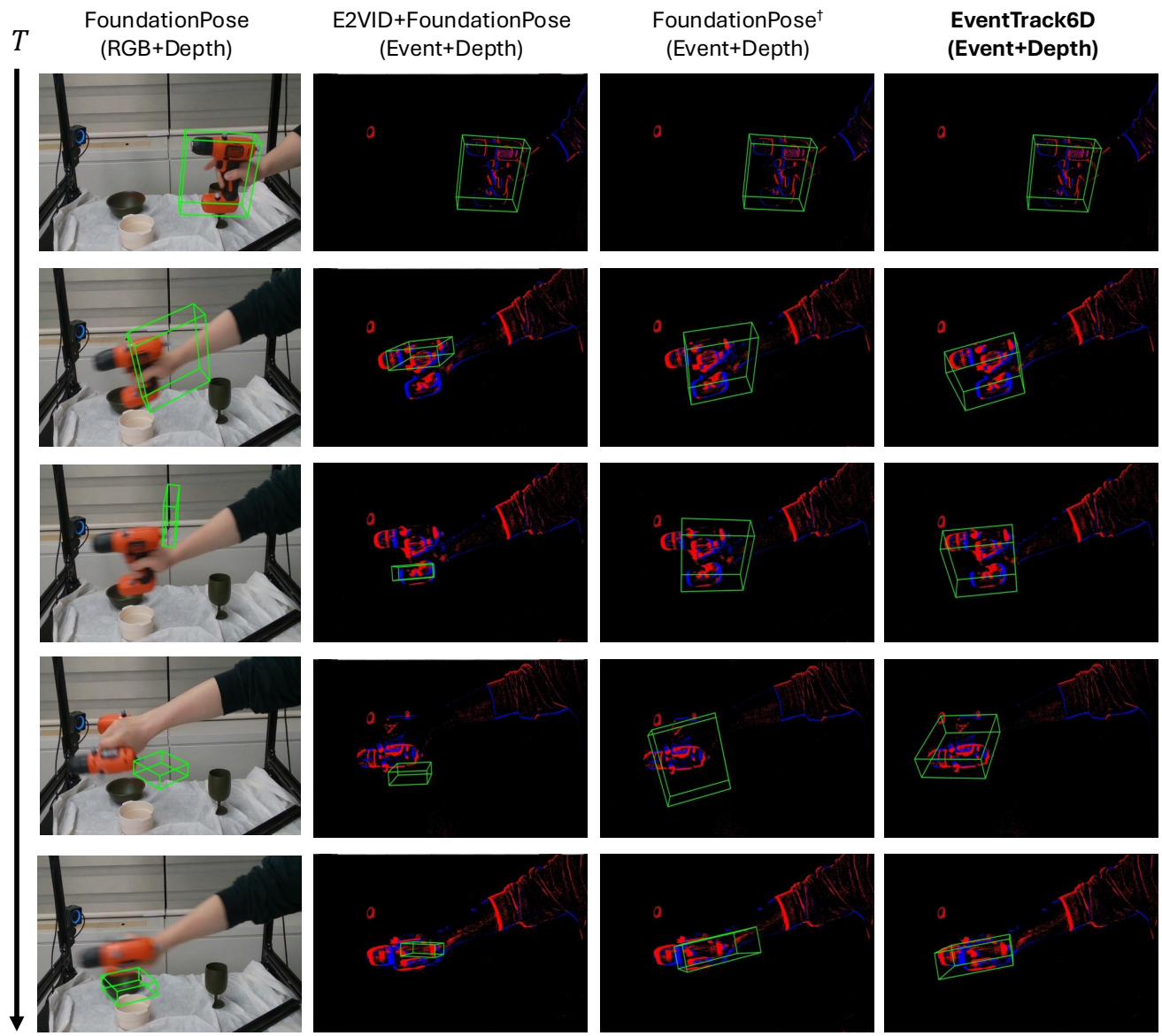

Figure 8. Qualitative comparison on the Event6D drill object sequence. Although the event-based methods operate at intervals corresponding to 120 FPS, all visualizations are presented at the RGB frame rate of 30 FPS for consistency. † denotes that the model is trained with event inputs on the EventBlender6D dataset.

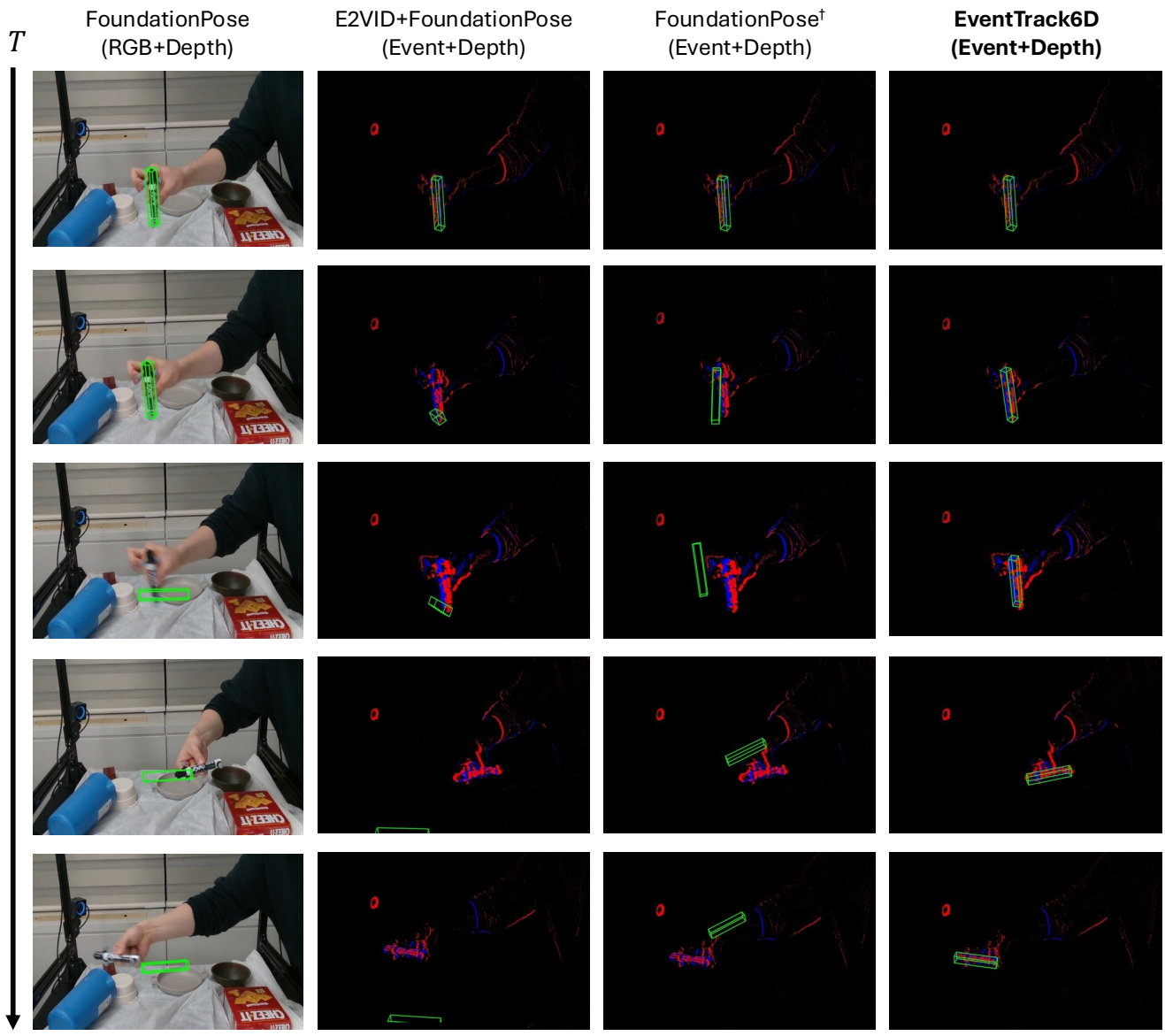

Figure 9. Qualitative comparison on the Event6D marker object sequence. Although the event-based methods operate at intervals corresponding to 120 FPS, all visualizations are presented at the RGB frame rate of 30 FPS for consistency.† denotes that the model is trained with event inputs on the EventBlender6D dataset.