Supplementary Material: SIS-Challenge: Event-based Spatio-temporal Instance Segmentation Challenge at the CVPR 2025 Event-based Vision Workshop

Friedhelm Hamann[1], Emil Mededovic[2], Fabian Gülhan[2], Yuli Wu[2], Johannes Stegmaier[2], Jing He[3], Yiqing Wang[3], Kexin Zhang[3], Lingling Li[3], Licheng Jiao[3], Mengru Ma[3], Hongxiang Huang[4], Yuhao Yan[5], Hongwei Ren[4], Xiaopeng Lin[4], Yulong Huang[4], Bojun Cheng[4], Se Hyun Lee[6], Gyu Sung Ham[6], Kanghan Oh[6], Gi Hyun Lim[6], Boxuan Yang[7], Bowen Du[7], and Guillermo Gallego[1]

[1] TU Berlin, SCIoI, ECDF, [2] RWTH Aachen, [3] Xidian University,

[4] Hong Kong University of Science and Technology,

[5] Sun Yat-sen University, [6] Wonkwang University, [7] Tongji University.

Overview

This supplementary material provides detailed descriptions of the methods employed by the 4th and 5th place teams in the SIS-Challenge. Due to space constraints in the main paper, these contributions are presented here in full detail.

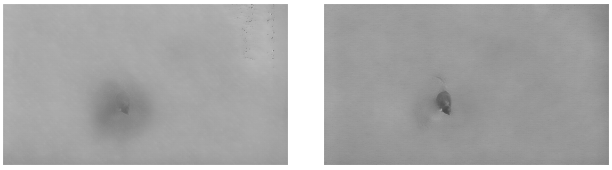
6. Additional Challenge Teams and Methods

6.1. Team 4: Shlee

6.1.1. Description

To address the challenge [6], the team specifically focuses on handling background noise in existing event-based instance segmentation methods. Event cameras inherently produce various types of noise, but here they focus on background noise, especially that caused by nearby light sources, which degrade performance in tasks such as object detection, tracking, and segmentation [5, 9]. In E2VID [12], during voxelization, the number of events in each voxel directly influences the quality of the reconstructed image. To mitigate this issue, the team employs a two-component Gaussian mixture model [3] to separate noisy events from informative ones, resulting in low- and high-frequency clusters. The filtered events are then converted into reconstructed images via E2VID. These images are passed to an object detector to produce denoised intensity images.

These images are used to fine-tune a YOLOv8 [7] detector, which outputs bounding boxes for each object. The resulting bounding boxes serve as prompts for the SAM2 [11] model, which predicts instance segmentation masks. Finally, an XMem [2] tracker is applied to link object instances across frames, producing a unified spatio-temporal segmentation and tracking output.



(a) Baseline Reconstruction (b) Team's Reconstruction

Figure 1. (*Team 4*). Comparison between the baseline frame (a) and the reconstructed frame (b).

6.1.2. Implementation Details

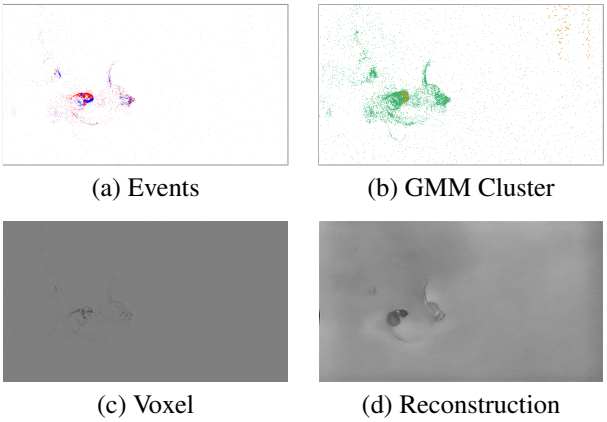
Data processing begins by identifying, for each image frame, the first event whose timestamp matches that frame. The team then applies a fixed-size event count window of 30,000 events centered on this event, collecting events both before and after to form the input set. This event subset is then passed to their event-count-based GMM clustering ($k = 2$), which generates a low-frequency cluster and a high-frequency cluster, with means μ_1 and μ_2, respectively. They compute the absolute difference of these means,

$$\Delta\mu = |\mu_1 - \mu_2|, \tag{1}$$

and compare it to a threshold $\tau = 2.5$. In other words:

$$\text{Voxel} = \begin{cases} \text{Select a low frequency cluster,} & \Delta\mu \geq \tau, \\ \text{Select both clusters,} & \Delta\mu < \tau. \end{cases} \tag{2}$$

After selecting clusters according to this rule, they voxelize the chosen events and feed them into the E2VID model for reconstruction, as shown in Fig. 2.



(a) Events (b) GMM Cluster

(c) Voxel (d) Reconstruction

Figure 2. (*Team 4*). Illustration of (a) events, (b) GMM clustering result, (c) voxelized events, and (d) final reconstructed frame.

The detector is a YOLOv8n model initialized with MouseSIS YOLO e2vid pretrained weights and fine-tuned on those reconstructed frames. Training runs for 300 epochs with batch size of 32, an initial learning rate of 0.001, a final learning rate of 0.0001, and weight decay is 5×10^{-4} on two RTX 3090 GPUs.

The team adopts SAM2 as a segmenter. YOLO-generated bounding boxes are provided as prompts to SAM2 to generate a binary mask for each frame. The resulting masks are produced once and utilized for downstream processing without updating the detector, as shown in Fig. 3.

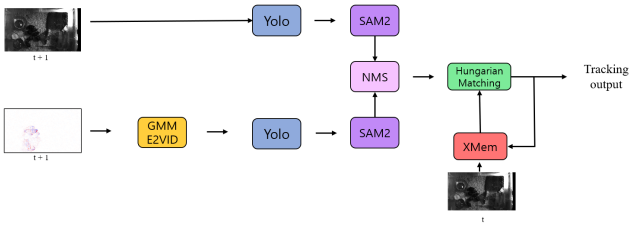


Figure 3. *Team 4*. Architecture overview of the proposed method.

6.1.3. Results

The method achieves a combined HOTA of 54.01 on the test set. Detailed per-sequence results are shown in Table 1.

6.2. Team 5: vivien

6.2.1. Description

Space-time Instance Segmentation (SIS) is crucial for detailed behavioral analysis, particularly in studies involving laboratory animals like mice. The MouseSIS dataset [6], used in this challenge, provides rich multi-modal data from grayscale frames and event-based cameras, whose advantages have been extensively surveyed [4]. The official ModelMixSort baseline [6] employs a tracking-by-detection paradigm inspired by methods like SORT [1], integrating YOLOv8 for object detection, SAM for instance segmentation and XMemSort for tracking.

Sequence	MOTA↑	IDF1↑	HOTA↑	DetA↑	AssA↑
10	0.754	0.647	0.536	0.619	0.470
16	0.592	0.718	0.501	0.533	0.471
22	0.425	0.568	0.387	0.442	0.343
26	0.242	0.438	0.404	0.415	0.408
28	0.544	0.756	0.576	0.530	0.626
32	0.913	0.954	0.741	0.728	0.756
Combined	0.606	0.677	0.540	0.549	0.535

Table 1. (*Team 4*). Per-sequence and combined results for MOTA, IDF1, HOTA, DetA, and AssA.

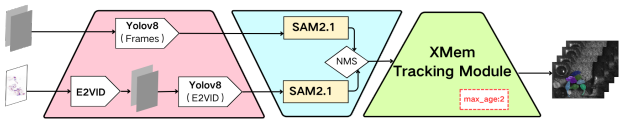


Figure 4. *Team 5*. Overall pipeline of the method, highlighting modifications to the ModelMixSort baseline.

The team's work was motivated by the opportunity to enhance several aspects of this baseline:

- **Segmentation Model and Mask Quality:** The baseline SAM implementation relies on the 'transformers' library and 'facebook/sam-vit-huge' [8], selecting a mask from multiple predictions based on IoU scores. The team hypothesized that utilizing a more recent SAM variant, SAM2.1, through the 'ultralytics.SAM' implementation with its direct mask output, could offer improved segmentation accuracy and potentially more consistent mask quality.
- **Tracking Persistence:** The XMemSort tracker in the baseline has a max_age of 1, meaning a track is terminated if it's not matched for more than a single frame. For dynamic mouse movements, this might be too aggressive, leading to premature track termination and increased ID switches. The team aimed to optimize this parameter for better tracking continuity.

6.2.2. Implementation Details

The method adheres to the tracking-by-detection framework established by ModelMixSort. The overall pipeline is illustrated in Fig. 4.

The key stages are:

1. **Data Preprocessing:** Event streams are converted into intensity frames using E2VID [12].
2. **Dual-Path Detection & Segmentation:**
   - YOLOv8 [7] detects mice in grayscale frames and

E2VID frames, providing bounding box proposals.
   - SAM2.1 [14] takes these bounding boxes as prompts to generate precise instance segmentation masks for each detected mouse.
3. **Detection Fusion:** Masks from the grayscale and E2VID pathways are merged using Non-Maximum Suppression (NMS).
4. **Tracking:** The fused instance masks are fed into the XMemSort tracker (an adaptation of XMem [2] with a SORT-like management logic).
5. **Output:** The final output consists of per-frame instance segmentation masks with consistent temporal IDs.

Data Preprocessing. The primary preprocessing step involves converting the raw event data from the DVS camera into a sequence of intensity frames. This is achieved using the E2VID model [12], as provided in the baseline framework. These E2VID-reconstructed frames are then synchronized with the standard grayscale video frames for subsequent processing. No other significant modifications were made to the data preprocessing stage of the baseline.

Detection Module (with SAM2.1). The detection module is responsible for identifying and segmenting individual mice in each frame from both grayscale and E2VID modalities.

Initial Object Detection with YOLOv8. Following the baseline, the team employs two instances of the YOLOv8 object detector:

- One detector is applied to the standard grayscale frames.
- Another detector is applied to the E2VID-reconstructed frames.

Both detectors are pre-trained and produce bounding boxes corresponding to potential mouse instances. These boxes serve as prompts for the subsequent segmentation stage.

Accurate Instance Segmentation with SAM2.1. A core modification in the approach is the change in the SAM implementation and model. The baseline utilizes 'facebook/sam-vit-huge' [8], selecting the final mask from multiple predictions based on SAM's internally predicted IoU scores.

The team replaced this with the 'ultralytics.SAM' implementation. The YOLOv8-generated bounding boxes are directly used as prompts for the 'predict' method of this SAM2.1 model, which then outputs the final segmentation masks. This change aims to leverage the potentially improved segmentation performance and a more streamlined mask generation process of the 'ultralytics.SAM' library with SAM2.1.

Tracking Module (with Refined XMem configuration). The tracking module associates the fused instance masks across consecutive frames, assigning consistent temporal IDs. The team uses the XMemSort tracker, which integrates the XMem video object segmentation model with a SORT-like track management system.

Core Tracking Mechanism. XMemSort takes the set of unique instance masks from the detection fusion stage as input for each frame. It maintains a set of active tracks and attempts to associate new detections with existing tracks based on mask IoU. For matched tracks, XMem updates the mask. For unmatched detections, new tracks are initialized.

Parameter Optimization. The primary optimization in the tracking module involves the max˙age parameter of XMemSort. This parameter defines the maximum number of consecutive frames a track can remain unmatched before it is terminated.

- **Baseline** max_age**:** The baseline configuration sets tracker:max_age to 1.
- **Team's** max_age**:** The team increased this value to **2** in their configuration.

The rationale for increasing max˙age from 1 to 2 is to provide slightly more persistence to tracks. With max˙age: 1, a track is deleted if it is not matched in the very next frame. By changing it to 2, a track can survive one missed frame and potentially be re-associated in the subsequent frame. This can be beneficial for handling very brief occlusions or momentary detection failures, thus improving track continuity for the dynamic movements of mice.

The min˙hits parameter (number of consecutive matches required to activate a track) was kept at 3, and the iou˙threshold for matching detections to tracks within XMemSort (configured under tracker:iou˙threshold) was kept at 0.3, consistent with the baseline.

Inference Procedure

1. The script loads the configuration from the specified YAML file, which defines data paths, model paths, and key parameters like NMS IoU threshold and tracker settings.
2. For each sequence in the specified split :
   - Grayscale frames are loaded from HDF5 files.
   - E2VID frames are pre-loaded from their respective directories.
   - The SamYoloDetector (utilizing YOLOv8 and the team's SAM2.1 setup) is instantiated separately for grayscale and E2VID data.
   - The XMemSort tracker is initialized with the team's optimized parameters (max_age: 2, min_hits: 3, iou_threshold: 0.3).
   - Per frame, detections are obtained from both grayscale and E2VID paths using the respective detectors.
   - These detections are fused using NMS.
   - The fused masks are passed to the XMemSort tracker, which updates track states and assigns IDs.
   - Results (segmentation masks and track IDs per frame)

are formatted into the COCO-like JSON structure.

3. Finally, results from all processed sequences are aggregated into a single final_results.json file for submission and evaluation.
4. Visualization of predictions per frame was enabled during processing for debugging and qualitative assessment.

6.2.3. Results

Experimental Setup

- **Dataset:** The experiments were conducted on the MouseSIS Challenge Dataset [6].
- **Evaluation Metrics:** Performance was evaluated using standard multi-object tracking metrics, including HOTA [10], CLEAR MOT [10], and IDF1 [13].
- **Software Environment:** The implementation is based on Python. Key libraries include PyTorch for deep learning models, Ultralytics (for YOLOv8 and SAM2.1), OpenCV for image processing, NumPy for numerical operations, h5py for data loading, and PyYAML for configuration management. All experiments were run on a Linux operating system.
- **Hardware:** Experiments were conducted on a system equipped with a 14-core Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz, 90 GB of RAM, and a single NVIDIA RTX 3090 GPU with 24GB VRAM.

Main Challenge Performance. The team's performance on the official test set of the SIS Challenge is reported in Tab. 1.

Ablation Studies and Parameter Analysis. All ablation studies were conducted on the combined MouseSIS validation sequences (03, 04, 12, 25).

Impact of SAM2.1 vs. Original SAM: To isolate the effect of upgrading the segmentation model, the team compared the original SAM with their modification using SAM2.1 (ultralytics.SAM) while keeping other key parameters consistent with the original baseline configuration. The results are shown in Tab. 2. The introduction of SAM2.1 led to substantial improvements across all major metrics, with HOTA increasing by 24.44% and IDF1 by 28.00%. This highlights the significant benefit of the newer segmentation model and the team's chosen implementation for this task.

Configuration	HOTA↑	MOTA↑	IDF1↑
Baseline (SAM)	0.45	0.62	0.50
Team's (SAM2.1)	**0.56**	**0.74**	**0.64**
Improvement (%)	+24.44	+19.35	+28.00

Table 2. (*Team 5*). Ablation Study: Original SAM vs. SAM2.1 (Validation Set, Combined).

Sensitivity to XMemSort's max_age **Parameter**: The team investigated the impact of the XMemSort's max_age parameter. These experiments were conducted using SAM2.1. The baseline configuration for max_age was 1. The team tested max_age values of 2 and 3.

max_age	HOTA↑	AssA↑	IDF1↑	MOTA↑	IDSW↓
1	0.557	0.512	0.638	0.740	96
2	0.560	0.516	0.657	0.742	80
3	**0.583**	**0.557**	**0.693**	**0.744**	**62**

Table 3. (*Team 5*). Sensitivity Analysis: XMemSort max_age (Validation Set, Combined).

Increasing max_age from 1 to 2 showed a slight improvement in HOTA and IDF1, and a reduction in IDSW. Further increasing max_age to 3 yielded the best validation performance. This suggests that allowing tracks to persist for a slightly longer duration (2–3 frames) when unmatched is beneficial for this dataset, likely by better handling brief occlusions or misdetections without losing track identity. The team's final submitted model used max_age=2 as a balance, though max_age=3 showed superior validation results.

References

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 3464–3468, 2016.

[2] Ho Kei Cheng and Alexander G Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 640–658, 2022.

[3] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[4] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1):154–180, 2022.

[5] Shasha Guo and Tobi Delbruck. Low cost and latency event camera background activity denoising. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):785–795, 2023.

[6] Friedhelm Hamann, Hanxiong Li, Paul Mieske, Lars Lewejohann, and Guillermo Gallego. MouseSIS: A frames-and-events dataset for space-time instance segmentation of mice. In *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, pages 156–173, 2024.

[7] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and

Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3992–4003. IEEE, 2023.

[9] Gi Hyun Lim and Se Hyun Lee. Labelling a stereo event dataset in indoor scenes for segmentation tasks. In *IEEE 21st Int. Conf. Ubiquitous Robots (UR)*, pages 619–623, 2024.

[10] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021.

[11] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[12] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43 (6):1964–1980, 2021.

[13] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 17–35, 2016.

[14] Ultralytics. Ultralytics SAM. https://github.com/ultralytics/ultralytics, 2023-2024.