Structural Information Leakage in Event-Based Camera Streams

Without Explicit Reconstruction

Ilinca Mocanu
Supervisor(s): Nergis Tömen, Tunahan Parlayici
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Ilinca Mocanu
Final project course: CSE3000 Research Project
Thesis committee: Nergis Tömen, Tunahan Parlayici, Ricardo Marroquim

Abstract

Event-based cameras are often considered more privacy preserving than conventional RGB cameras because they don't capture full image frames, colour, or texture. Nonetheless, their raw event streams might still encode structural information about the recorded scene. This paper questions this assumption and investigates this privacy concern experimentally by converting raw events into direct event representations and evaluating whether machine-learning models can recover semantic, spatial, and motion structure without explicit image reconstruction. Three forms of leakage are studied: semantic leakage through segmentation, spatial layout leakage through depth estimation, and motion leakage through optical flow estimation. All experiments are mainly based on DSEC dataset. As an extension dataset, PEDRo is used to investigate human-specific semantic leakage. The segmentation experiments show that semantic leakage is present but uneven: large and persistent driving scene regions such as road and background are recovered more reliably than sparse human regions in DSEC, while PEDRo shows clearer leakage of approximate human location through human-box segmentation. The depth estimation experiment shows that event representations preserve enough geometric information for a pretrained model to recover coarse scene depth. The optical flow experiment further outlines that event streams preserve recoverable motion information, since a pretrained event-based model can estimate dense motion patterns from the data. These findings highlight that event cameras don't guarantee privacy by sensor design alone and that privacy in event-based vision depends on the representation, temporal window, task, model, and dataset. The segmentation code used in this project is available at https://github.com/ilincamaria03/event_camera_segmentation.

1 Introduction

Event-based cameras are becoming a popular sensing technology in computer vision due to their advantages over conventional cameras. Compared to normal RGB cameras, they don't capture full image frames synchronously but instead report brightness changes [1]. This makes event cameras suitable for applications that require low motion blur, high temporal resolution [2] and high-contrast scenes [3]. For these reasons, they are more and more studied for autonomous driving [4], camera-surveillance [1], and even human-computer interaction [5, 2].

At first sight, event cameras may appear more privacy-preserving than standard RGB cameras. Since they don't directly report color, texture, or full image frames, they seem to avoid some of the most obvious privacy risks of conventional video [1]. A raw event stream is also harder for a human observer to interpret than a normal camera image [6]. Nonetheless, this doesn't automatically mean that event-camera data is private. The same events that make these sensors useful for vision tasks may also encode information about the surrounding scene. Moving objects produce contours, humans can generate recognizable silhouettes, and the structure of roads, buildings, vehicles, or outdoor spaces may still be visible after events are accumulated into a representation [7].

Conventional visual-privacy research has already shown that suppressing detailed appearance doesn't eliminate privacy risk, because people can still be identified or characterized from non-facial cues such as clothes, height, gait, posture, and contextual scene information [8, 9, 10, 11]. Event-camera research suggests an analogous risk. Du et al. [12] explicitly distinguish reconstruction attacks from recognition attacks on neuromorphic vision sensors and note that event-based recognition can reveal privacy-relevant information even when no clear grayscale image is reconstructed.

Existing event-camera research has focused mainly on improving visual performance in tasks such as segmentation [13], depth estimation [14], optical flow [15, 16], reconstruction [3], and recognition [17]. These results show that event data can support rich visual inference, but the privacy implications of this capability are less directly studied. This project therefore investigates event-camera privacy from the perspective of structural information leakage. In this paper, **structural scene information** refers to visual information that describes the organization of a scene, including object contours, human location, human contours, spatial depth structure, and motion patterns. **The main research question is:** *What types of structural scene information can be inferred from raw event-based data without explicit image reconstruction?*

To answer this research question, this paper studies three forms of structural information leakage in event-camera data. First, semantic leakage is evaluated by training segmentation models on different event representations, as shown in Figure 1, and testing whether object classes and human regions can be recovered. Second, spatial layout leakage is investigated through depth estimation, using pretrained event-based depth estimation to assess whether scene structure and relative object distance can be inferred. Third, motion leakage is considered through optical flow, which tests whether event streams preserve information about the movement of objects and scene regions, using another pretrained optical flow estimation model. Together, these experiments examine whether privacy risks remain even when event data is not explicitly reconstructed into conventional images.

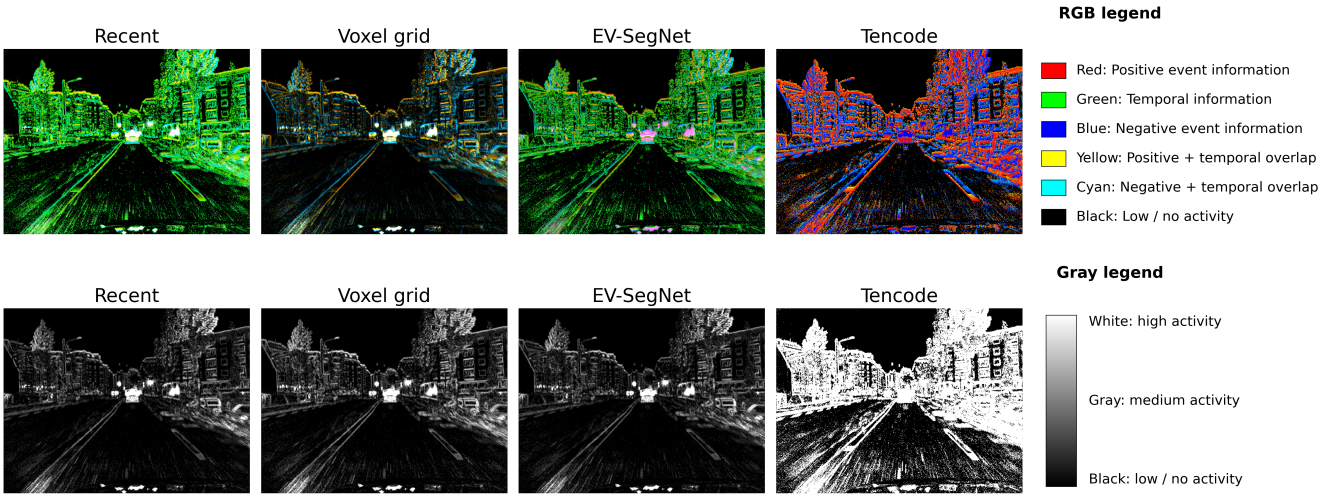

Figure 1: **This figure is only a visualization of the representation tensors; the RGB colors are not real event-camera colors.** Qualitative visualization of four event representations on the same 50 ms DSEC event slice. The top row shows RGB-like visualizations and the bottom row shows grayscale activity maps. In the RGB row, red denotes positive-polarity event information and blue denotes the same, but negative. The green channel visualizes temporal information, but its exact meaning depends on the representation: for Recent, it is the normalized most recent timestamp at each pixel and polarity; for the voxel grid, it is a weighted sum of temporal bins, with later bins weighted more strongly; for EV-SegNet, it is derived from the mean timestamp channels; and for Tencode, it is the representation's timestamp channel. Mixed colors indicate overlap between polarity and temporal information. In the grayscale row, brighter pixels indicate higher event activity and black pixels indicate low or no activity.

The remainder of this paper is organized as follows. Section 2 presents the background and related work on event representations, privacy claims, and event-based vision tasks. Section 3 describes the methodology used to evaluate structural information leakage from event representations and the utilized experimental setup, including the datasets, model choices, and evaluation metrics. Section 4 presents and analyzes the results for semantic leakage, spatial layout leakage, and motion leakage. Section 5 discusses responsible research considerations, including ethical implications, dataset limitations, and reproducibility. Finally, Section 6 concludes the paper by summarizing the main findings of the study.

2 Related work

2.1 Event Representations

Following Gehrig et al. [18], an event representation can be understood as a grid-like tensor T obtained by mapping an asynchronous set of events $E = \{(x_k, y_k, t_k, p_k)\}_{k=1}^{N}$, where each event encodes spatial position, timestamp, and polarity, into a format compatible with CNN-based vision models. The *event-count* representation uses a single channel that counts the number of events at each pixel, preserving only the spatial distribution of event activity [19]. The *polarity histogram* representation uses two channels, one for positive events and one for negative events, so it preserves whether brightness increased or decreased but still collapses the temporal order inside the event window [13]. The *Recent* representation extends the polarity histogram with two additional channels that store the normalized timestamp of the most recent positive and negative event at each pixel, adding local temporal recency information [13]. The *voxel-grid* representation divides the event window into temporal bins and accumulates events within each bin, preserving more temporal structure than count-based representations [14]. The *EV-SegNet* representation uses six channels: positive and negative event histograms, the mean timestamp for each polarity, and the timestamp standard deviation for each polarity, combining event density with the temporal distribution of events [13]. Finally, *Tencode*, used by DepthAnyEvent, is an RGB-like event representation where the red and blue channels encode positive and negative polarities, while the green channel stores the event timestamp relative to the event-slice duration, making the event data compatible with image-pretrained depth models [14].

The choice of event representation is therefore both an implementation detail and a task-dependent design decision. Gehrig et al. [18] show that event representations can be learned together with the downstream task network and that this improves performance on tasks such as optical flow estimation and object recognition.

Similarly, Alonso and Murillo [13] propose an event representation specifically for semantic segmentation, which in this paper will be called Ev-SegNet representation, and show that it performs better than simpler event encodings for this task. For optical flow, Wu et al. [15] use a discretized event voxel grid and process event bins sequentially, due to the fact that optical flow estimation depends on the spatiotemporal traces left by moving events over time. For depth estimation, DepthAnyEvent uses Tencode because it preserves polarity and timing in a three-channel format compatible with image-pretrained depth models [14]. This suggests that different representations expose different parts of the event stream: segmentation benefits from representations that preserve spatial contours and local temporal structure, optical flow benefits from temporally binned representations that retain motion traces, and depth estimation benefits from representations that encode polarity and timing in an image-compatible form.

2.2 Event-based Applications

Previous work in event-based applications suggests that event streams can contain structural scene information even before explicit image reconstruction. Alonso and Murillo [13] show that semantic segmentation can be learned from event-camera data alone by using event representations as input and producing segmentation outputs on driving scenes. This is directly relevant to scene layout leakage, because segmentation requires the model to recover object boundaries from the event stream. Nevertheless, their work evaluates segmentation as a useful computer-vision task, not as a privacy risk, and it doesn't compare representations in terms of how much structural information they expose.

Human-related applications provide a second indication that event data can preserve privacy relevant structure. Ahmad et al. [1] show that person re-identification is possible using event-camera representations, suggesting that event streams can retain human-shape cues such as edges, contours, and motion patterns. This is relevant to silhouette leakage, but the focus of their work is identity recognition across views rather than a systematic analysis of whether human presence or body shape is recoverable from different event representations.

Depth estimation provides another example of structural information that can be inferred from event streams. Hidalgo-Carrió et al. [20] show that dense monocular depth can be predicted from events alone, indicating that event data can preserve geometric information about the surrounding scene. DepthAnyEvent further shows that event-based monocular depth estimation can benefit from image pretrained foundation models by converting events into the Tencode representation and using cross-modal distillation from image-based depth models [14]. These works are relevant for structural leakage because depth estimation recovers object boundaries and also exposes spatial layout and scene geometry. Nonetheless, their goal is to improve depth prediction performance, not to evaluate depth recovery from event representations.

For motion structure, Zhu et al. [16] show that event streams can be used to predict optical flow and egomotion. More recent optical flow work also relies on temporally structured event inputs, such as voxel-grid event bins, because optical flow depends on the spatiotemporal traces left by moving events [15]. These works are important for understanding motion leakage, but they are mainly designed to maximize motion estimation accuracy and not to evaluate privacy exposure.

Together, these studies show that event streams can support semantic segmentation, human recognition, motion estimation and depth estimation. The remaining gap is that these capabilities are usually studied as computer vision applications, while privacy leakage from the intermediate event representations themselves is not systematically evaluated. This study addresses that gap by treating downstream performance as a probe for representation level leakage: if a model can infer human silhouettes, scene layout, motion, or depth from an event representation, then that representation exposes corresponding structural information even without explicit reconstruction.

2.3 Privacy in Event-based Vision

Privacy in event-based vision has been studied mainly through identity protection, event-to-image reconstruction, and anonymization. Although event cameras only record brightness changes and are difficult for humans to interpret directly, this doesn't make their output private by default. Reconstruction methods such as E2VID show that intensity video can be recovered from event streams, making event data visually interpretable again and allowing standard computer-vision pipelines to be applied to the reconstructed output [3]. More recent privacy work therefore treats event-to-image conversion itself as a possible attack surface: Kim et al. [6] consider client-server event-based localization and identify privacy risks such as inversion, weight-swapping, and reverse-engineering attacks during event-to-image reconstruction.

Anonymization work addresses these risks more directly. Ahmad et al. [21] propose event anonymization to reduce identity-revealing reconstructions while preserving downstream utility. Nonetheless, Bendig et al. [22] argue that preventing human-readable reconstruction is not sufficient, because neural networks may still recover identity

relevant information from anonymized event data. Their AnonyNoise method therefore targets re-identification directly by adding learnable data-dependent noise, and evaluates robustness against both image reconstruction and inversion attacks. Similarly, Adra and Dugelay [23] frame event-to-video reconstruction as a privacy risk and propose E2PRIV to anonymize faces during reconstruction while keeping the videos useful for action recognition.

These works highlight that event cameras shouldn't be treated as automatically privacy-preserving, but leave a different question less explored: *what structural scene information is already exposed by event representations before any standard image or video is reconstructed?* This project addresses that gap by evaluating whether task-specific event representations, including voxel grids, Recent, EV-SegNet-style representations, and Tencode, expose human silhouettes, semantic scene layout, motion cues, or depth-related structure directly from the representation itself.

3 Methodology

This work evaluates whether event-camera data can leak privacy-relevant information using three different experiments. The first experiment is divided into two sub-experiments. The first sub-experiment studies **general semantic segmentation leakage**: whether different event representations preserve enough information to recover object and scene categories such as road, vegetation, vehicles, and humans. If an event representation allows a model to recover surrounding objects and scene structure, it may also reveal contextual information about where a person is, what kind of environment is being recorded, and what activities may be taking place. The second sub-experiment studies **human focused segmentation leakage**: whether event representations preserve enough information to localize humans or human-related regions. This experiment directly evaluates whether people can be detected or spatially localized from event data.

The second experiment is a depth estimation experiment and it studies **spatial layout leakage**: whether event data can support monocular depth estimation, revealing the three-dimensional structure of the scene. Finally, the third experiment is an optical flow experiment and it studies **motion leakage**: whether event data preserves enough spatiotemporal information to estimate the direction and magnitude of motion in the scene.

All experiments use synchronous event representations as input to neural networks. This is necessary because event cameras produce asynchronous streams of events rather than standard image frames. For each task, events are accumulated over fixed temporal windows and converted into dense tensor representations.

3.1 Segmentation Experiment

3.1.1 Event Representations

The segmentation experiment compares three event representations: recent timestamp encoding, voxel grid, and the six-channel EV-SegNet-style representation [13]. This limited selection is motivated by prior work on event-based segmentation and event representation learning. EV-SegNet shows that event data can support semantic segmentation when encoded into dense image-like tensors, and reports that its proposed representation performs better than simpler dense event encodings for this task [13]. Voxel-grid representations are also widely used in CNN-based event-vision pipelines because they preserve the temporal distribution of events within the integration window [16, 18]. Recent timestamp-based encodings, the inspiration for the EVSegNet representation, are included because they retain local recency information at each pixel and polarity, which is useful for representing object boundaries and motion-induced structure [16]. Choosing these representations makes the segmentation experiment a worst-case scenario for leakage, meaning that these representations are more likely to contain inferable information.

All representations are generated from the same event window for a given sample and are resized to the same input resolution before being passed to the segmentation model. This keeps the input size and model architecture fixed, so that differences in performance can be attributed mainly to the information preserved by each representation. For more information about the exact implementation, check the Appendix.

3.1.2 Datasets

The segmentation experiment uses two event-camera datasets: DSEC and PEDRo.

DSEC [24] is used for semantic segmentation in driving scenes. Although DSEC-Semantic provides labels for a larger set of DSEC sequences, this experiment uses a selected subset of five Zurich sequences: `zurich_00`, `zurich_06`, and `zurich_07` are used for training, `zurich_08` is used for validation, and `zurich_13` is used for testing. This avoids mixing frames from the same driving sequence across training and evaluation.

The original semantic labels [25] are mapped to a smaller set of urban categories inspired by EV-SegNet [13]: flat, background, object, vegetation, human, and vehicle. In addition to the multiclass setup, a human-focused

binary setup is used in which the classes are reduced to background and human. This makes it possible to evaluate both general scene understanding and explicit human related leakage.

PEDRo [26] is used as a second dataset focused on person detection in robotic scenarios. The predefined train, validation, and test folders are used. Since PEDRo provides bounding-box annotations rather than pixel-level human masks (see example in Fig. 2), the bounding boxes are converted into binary box masks. Therefore, the PEDRo experiment should be interpreted as *human box-mask localization*, not as precise human silhouette segmentation.

For both datasets, the data is split into training, validation, and test partitions. The validation set is used for model selection, while the final reported metrics are computed on the test set. The same splits, preprocessing procedure, and evaluation protocol are kept fixed across event representations to make the comparison fair.

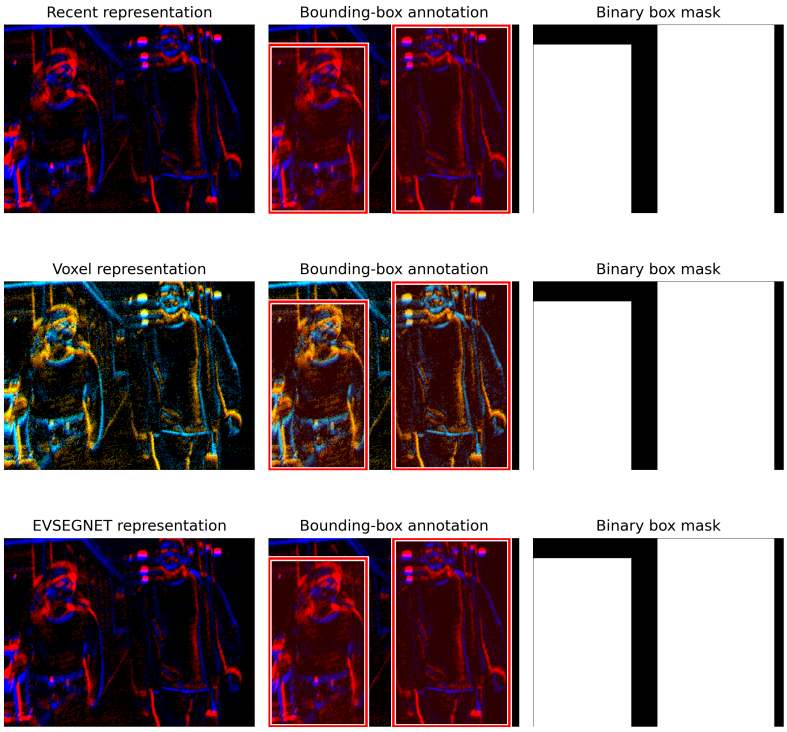

Figure 2: **Visualization of PEDRo event representations and labels.** Each row shows the same 40 ms event window represented as Recent, voxel grid, or EV-SegNet. The colours are only visualization channels, not reconstructed RGB appearance: red and blue indicate positive and negative polarity event information, while green encodes temporal information in a representation-specific way (see Fig. 1) The middle column shows the PEDRo bounding-box annotations, and the right column shows the filled binary box mask used as the human segmentation target.

3.1.3 Model

The segmentation model is inspired by EV-SegNet [13], which showed that event representations can be used as input to an encoder-decoder CNN for semantic segmentation. This makes EV-SegNet a suitable reference for this experiment, since the goal is also to evaluate how much semantic information can be recovered from event representations. Nonetheless, the model used in this work isn't an exact reimplementation of EV-SegNet, but a model with a lighter architecture to keep training feasible across several event representations and repeated runs.

The model follows the same general encoder-decoder principle: the encoder extracts progressively more abstract features from the input representation and the decoder upsamples these features into a dense pixel-level segmentation map. Compared with the full Xception-based EV-SegNet architecture, the model used here is smaller: it uses simplified Xception-style downsampling blocks, a reduced channel width, a compact decoder, and a lightweight multi-scale context block based on dilated convolutions. Skip connections are used between encoder and decoder stages to preserve spatial detail, which is important for detecting small regions such as humans or human-box areas.

3.1.4 Training and Evaluation Protocol

For DSEC, both the multiclass setup and the human-focused setup use the full selected split without frame-level filtering or oversampling. This means that the human-focused experiment is evaluated under the original class

imbalance of the dataset. Class imbalance is handled through class weighting, and dice loss is added for the binary human segmentation setup. For PEDRo, the XML bounding-box annotations are converted into binary box masks, so the PEDRo task is interpreted more as human localization rather than precise silhouette segmentation.

Each representation is evaluated with the same three random seeds: 42, 43, and 44. These seeds control stochastic parts of training such as weight initialization, data shuffling, and data augmentation. Results are therefore reported as mean and standard deviation across seeds.

Different event-window lengths are evaluated at test time. For DSEC, models are trained with 50 ms event windows and evaluated with 10 ms, 50 ms, and 250 ms windows. For PEDRo, models are trained with 40 ms windows and evaluated with 10 ms, 20 ms, and 40 ms windows. Shorter windows contain less event information and therefore test how much can be inferred from a small amount of activity. Longer windows accumulate more events and can reveal more structure, but may also blur together motion over time. Evaluating multiple windows therefore helps measure how privacy leakage changes with the amount of temporal information available. More technical information can be found in the Appendix.

3.1.5 Experimental Setup

The segmentation experiments were implemented in Python using PyTorch and were run in a Kaggle GPU environment with NVIDIA T4 GPUs available. Training was executed on CUDA with automatic mixed precision enabled. The recorded software environment used Python 3.12.13, PyTorch 2.10.0+cu128, NumPy 2.4.6, CUDA 12.8, and cuDNN 9.10.2.

3.1.6 Evaluation Metrics

The segmentation experiment uses pixel accuracy, mean Intersection over Union (mIoU), and focused IoU per class. These metrics are computed from the predicted segmentation mask and the ground-truth mask.

Pixel accuracy is the fraction of correctly classified pixels. Although it is easy to interpret, it isn't sufficient on its own because most pixels often belong to the background. A model can obtain high accuracy even when performing poorly on smaller but privacy-relevant classes, such as humans.

Intersection over Union (IoU) measures the overlap between the predicted region and the ground-truth region for a class: $IoU = \frac{TP}{TP+FP+FN}$, where TP is the number of correctly predicted pixels of the class, FP is the number of pixels incorrectly predicted as that class, and FN is the number of pixels of that class missed by the model. IoU is more informative than accuracy for segmentation because it penalizes both false detections and missed regions.

The experiments report focused IoU per class. Focus IoU refers to the IoU of the class or group of classes most relevant to the experiment. In the DSEC multi-class setup, the focus classes are the six EV-SegNet-style urban categories, which are: flat (road and pavement), background (construction and sky), object, vegetation, human, vehicle [13]. In the human-focused DSEC setup, the focus class is the human class, so focused IoU corresponds to human IoU. In the PEDRo setup, the equivalent metric is human-box IoU, because the target masks are generated from bounding-box annotations rather than precise pixel-level human silhouettes.

For the multiclass DSEC experiment, mIoU is used as the main metric. It is computed by calculating the IoU for each class and then averaging over all classes. This gives a more balanced view of segmentation performance than pixel accuracy, especially when some classes occupy much larger image areas than others.

For the privacy-focused experiments, human IoU or human-box IoU is the most important metric. A higher value means that the event representation preserves more information about the location of people. Therefore, in this work, human-focused IoU is interpreted as the main indicator of human-related privacy leakage.

3.2 Depth Estimation Experiment

3.2.1 Event Representation

The depth estimation experiment uses the input representation expected by the pretrained Depth AnyEvent model [14]. This experiment keeps the representation fixed to Tencode (described in section 2). In the Depth AnyEvent pipeline, this representation is used to reduce the gap between event data and the RGB-like input format expected by the pretrained image-based depth model. The paper's selected time slicing window is of $50ms$.

3.2.2 Dataset

The depth estimation experiment uses DSEC, which provides event streams from stereo driving sequences together with ground-truth disparity labels. In this experiment, the evaluated DSEC sequences are taken from the

train split and include `interlaken_00_f`, `interlaken_00_g`, `thun_00_a`, `zurich_city_05_a`, `zurich_city_05_b`, `zurich_city_06_a`, `zurich_city_07_a`, `zurich_city_08_a`, `zurich_city_09_d`, and `zurich_city_10_b`.

In DSEC, the ground-truth disparity is obtained using LiDAR-based processing and filtered to remove outliers. Since the pretrained depth model, described in 3.2.3, predicts depth rather than disparity, the DSEC disparity labels are converted into depth using the stereo calibration parameters. This allows the predicted depth maps to be compared with ground-truth scene geometry.

3.2.3 Model

The depth estimation experiment uses the pretrained Depth AnyEvent-R model. Depth AnyEvent is an event-based monocular depth estimation model. Their pipeline converts event streams into Tencode representations and uses a vision foundation model adapted to the event domain. The recurrent variant, DepthAnyEvent-R, extends this model by integrating information from previous event stacks through recurrent modules.

DepthAnyEvent-R is chosen in this project because event-camera data is naturally sequential. A single event slice may contain only sparse information, especially in regions with little motion or few brightness changes. By maintaining information from previous event stacks, the recurrent model can use temporal context instead of estimating depth from each slice in isolation. Choosing the stronger pretrained variant makes the experiment closer to a worst-case privacy analysis, meaning that if DepthAnyEvent-R can infer meaningful depth from events, then the representation preserves information about the three-dimensional layout of the scene.

Given a Tencode event representation as input, DepthAnyEvent-R predicts a dense depth map. The prediction is then compared with the DSEC ground-truth depth obtained from its disparity labels.

3.2.4 Experimental Setup

The depth estimation experiment uses the released pretrained DepthAnyEvent-R checkpoint without additional training or fine-tuning. The experiment was run locally on a Lenovo Legion 5 laptop with an AMD Ryzen 7 5800H CPU, 32 GB RAM, and an NVIDIA GeForce RTX 3070 Laptop GPU, running Windows 11. The software environment used PyTorch 2.5.1+cu124, Torchvision 0.20.1+cu124, Torchaudio 2.5.1+cu124, NumPy 2.1.2, OpenCV 4.12.0, h5py 3.14.0, hdf5plugin 5.1.0, Kornia 0.8.1, and Matplotlib 3.10.6. The CUDA-enabled PyTorch build used CUDA 12.4 with cuDNN 9.1.0.70. Since the model is used only for evaluation, training hyperparameters doesn't apply. Reproducibility only depends on the released checkpoint, the Tencode preprocessing pipeline, the DSEC sequences used for evaluation, and the software environment.

3.2.5 Evaluation Metrics

Depth estimation is evaluated using standard monocular depth metrics: Absolute Relative Error $AbsRel$, Squared Relative Error $SqRel$, Root Mean Squared Error $RMSE$, logarithmic RMSE $RMSE_{log}$, scale-invariant logarithmic error $SIlog$, and threshold accuracies δ_1, δ_2, and δ_3. Let d_i be the predicted depth at pixel i, d_i^* the ground-truth depth, and N the number of valid pixels. $AbsRel$ computes the mean relative absolute error, $\frac{1}{N}\sum_i \frac{|d_i - d_i^*|}{d_i^*}$, while $SqRel$ gives more weight to large mistakes by using $\frac{1}{N}\sum_i \frac{(d_i - d_i^*)^2}{d_i^*}$. $RMSE$ measures the average absolute depth error in metres after squaring the errors, and $RMSE_{log}$ applies the same idea in logarithmic depth space. $SIlog$ also operates in log-depth space, but reduces the influence of global scale differences between prediction and ground truth. The threshold accuracies measure the percentage of pixels for which the prediction is close to the ground truth, using $\max(d_i/d_i^*, d_i^*/d_i) < 1.25$, 1.25^2, and 1.25^3 for δ_1, δ_2, and δ_3, respectively. Lower values are better for the error metrics, while higher values are better for the threshold accuracies.

These metrics quantify different aspects of depth leakage. RMSE and mean depth error indicate the absolute size of depth mistakes. Abs Rel and Sq Rel measure errors relative to the true depth. The δ metrics measure the fraction of pixels for which the predicted depth is close to the ground truth.

3.3 Optical Flow Estimation Experiment

3.3.1 Event Representation

The optical flow experiment uses the event representation expected by the pretrained IDNet model [15], which is the voxel-grid representation. In IDNet, the event stream is discretized into temporal bins, so that the model receives a dense spatiotemporal representation of recent event activity. This representation preserves both where events occur

and how they are distributed over time, which is necessary for estimating motion from the event stream. IDNet uses event windows of $100ms$ represented with temporal event bins.

3.3.2 Dataset

The optical flow experiment is evaluated on DSEC-Flow, the optical flow benchmark derived from the DSEC driving dataset [27]. DSEC provides high-resolution event-camera recordings in driving scenes, and DSEC-Flow defines the task of estimating forward optical flow for the left event camera at selected timestamps. Optical flow describes the apparent two-dimensional motion of pixels in the image plane between two time points [15].

This experiment uses the DSEC-Flow test sequences required by the IDNet evaluation setup: `interlaken_00_b`, `interlaken_01_a`, `thun_01_a`, `thun_01_b`, `zurich_city_12_a`, `zurich_city_14_c`, and `zurich_city_15_a`. For each sequence, the pretrained IDNet model predicts forward optical flow at the timestamps specified by the DSEC-Flow benchmark. The predictions are then saved in the required DSEC-Flow submission format.

3.3.3 Model

The optical flow experiment uses the pretrained IDNet model [15]. IDNet is an event-based optical flow network designed to estimate dense optical flow from event traces without explicitly constructing correlation volumes by exploiting the spatiotemporal traces produced by events during motion. The model uses a recurrent network to process event bins and predicts optical flow from the residual motion visible in the event representation.

The pretrained IDNet model is applied directly to the DSEC event sequences. For each selected timestamp, the corresponding event window is converted into the voxel-grid representation expected by the model. The model then predicts a dense forward optical flow field for the left event camera. The generated flow predictions are saved in the DSEC-Flow submission format and evaluated using the DSEC-Flow benchmark protocol [27].

The central mechanism of IDNet is iterative deblurring. Starting from an initial flow estimate, the event representation is deblurred, according to the current estimate. The network then predicts a residual flow update from the deblurred events. Repeating this process allows the model to refine the flow estimate over several iterations. This is useful for event data because motion appears directly as event traces over time, and deblurring these traces can make the remaining motion easier for the model to estimate.

This model is used as a pretrained model without additional training or fine-tuning. In this case, the experiment evaluates the motion information that can be recovered by an already trained event-based optical flow pipeline. If IDNet can infer accurate optical flow from the event stream, then it can be deducted that event data preserves information about movement in the scene.

3.3.4 Evaluation Protocol & Experimental Setup

No retraining or fine-tuning is performed in this experiment. This means that optimizer settings, learning-rate schedules, training epochs, and random seeds don't apply. The reproducibility of this experiment depends mainly on the released pretrained checkpoint, the DSEC input sequences, the event preprocessing used by IDNet, and the benchmark evaluation protocol.

The optical flow experiment uses the released pretrained IDNet checkpoint without additional training or fine-tuning. The experiment was run locally on a Lenovo Legion 5 laptop with an AMD Ryzen 7 5800H CPU, 32 GB RAM, and an NVIDIA GeForce RTX 3070 Laptop GPU, running Windows 11. The software environment followed the official IDNet conda environment, using Python 3.8, PyTorch 1.13.1, and CUDA 11.7 through the `pytorch-cuda=11.7` package. Additional dependencies included Torchvision, scikit-image, NumPy/Numba-related processing tools, pandas, h5py, hdf5plugin, OpenCV, Hydra, Matplotlib, tqdm, and imageio.

3.3.5 Evaluation Metrics

The optical flow experiment is evaluated using the metrics reported by the DSEC-Flow benchmark: Endpoint Error (EPE), Angular Error (AE), and n-pixel error (nPE). Let $\hat{\mathbf{f}}_i = (\hat{u}_i, \hat{v}_i)$ be the predicted optical flow vector at pixel i, and let $\mathbf{f}_i^* = (u_i^*, v_i^*)$ be the ground-truth optical flow vector. Endpoint Error measures the Euclidean distance between the predicted and ground-truth flow vectors: $EPE_i = \sqrt{(\hat{u}_i - u_i^*)^2 + (\hat{v}_i - v_i^*)^2}$. The reported EPE is the average over all valid ground-truth pixels. Lower EPE indicates that the predicted motion vector is closer to the ground truth.

Angular Error measures the angular difference between the predicted and ground-truth flow vectors. This captures whether the predicted motion direction is correct, even when the magnitude differs. The nPE metrics

report the percentage of valid pixels for which the endpoint error is greater than n pixels, typically for $n = 1$, 2, and 3. These metrics show how often the model makes errors larger than a given pixel threshold.

For the privacy analysis, optical flow accuracy is interpreted as motion leakage. A lower EPE, lower AE, and lower nPE indicate that the event stream contains enough information for the model to recover scene motion more accurately.

4 Results and Discussion

4.1 Segmentation Experiment

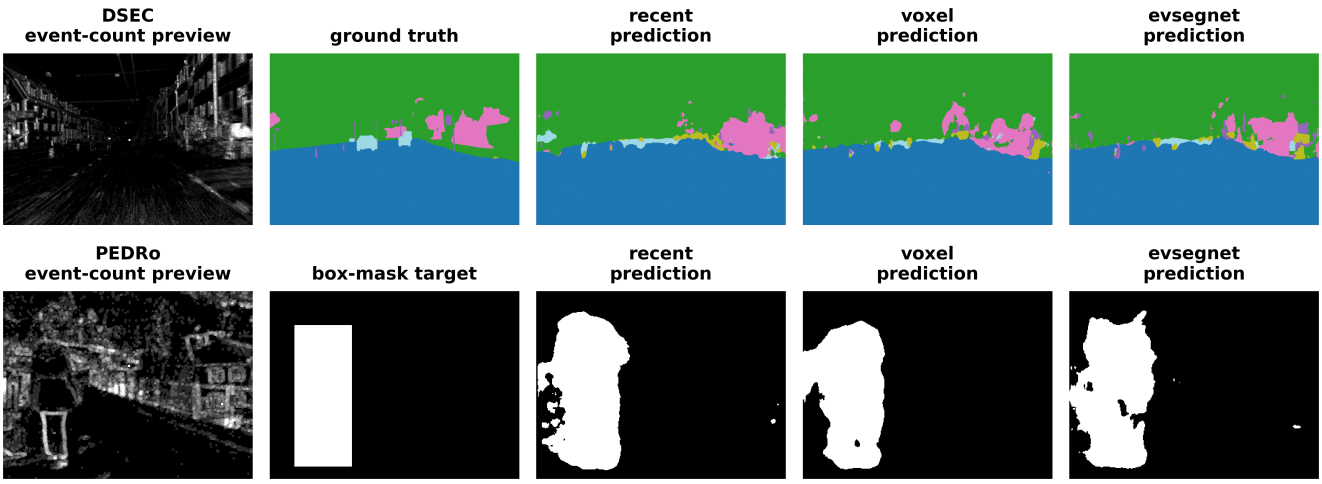

Figure 3: Qualitative segmentation examples for DSEC and PEDRo. The top row shows a DSEC multiclass segmentation example, with an event-count preview, the ground-truth semantic mask, and predictions from the Recent, voxel-grid, and EV-SegNet representations. The bottom row shows a PEDRo human-box segmentation example, with an event-count preview, the binary box-mask target, and the corresponding predictions from the same representations. The event-count previews are visualization inputs only. The DSEC example illustrates semantic scene layout recovery, while the PEDRo example illustrates approximate human location recovery.

Table 1: DSEC multiclass segmentation results. Values are mean ± standard deviation over seeds 42, 43, and 44. Bold indicates the strongest segmentation score for each metric and time window; green bold indicates the weakest leakage score.

Representation	Time window	mIoU ↑	Acc. ↑
Recent	10 ms	**0.385 ± 0.023**	**0.854 ± 0.020**
Recent	50 ms	**0.407 ± 0.007**	**0.880 ± 0.001**
Recent	250 ms	0.354 ± 0.007	**0.844 ± 0.001**
Voxel grid	10 ms	0.386 ± 0.013	0.861 ± 0.004
Voxel grid	50 ms	0.410 ± 0.010	**0.887 ± 0.002**
Voxel grid	250 ms	**0.362 ± 0.010**	**0.861 ± 0.003**
EVSegNet	10 ms	**0.398 ± 0.013**	**0.870 ± 0.011**
EVSegNet	50 ms	**0.415 ± 0.002**	0.886 ± 0.003
EVSegNet	250 ms	**0.353 ± 0.012**	0.849 ± 0.011

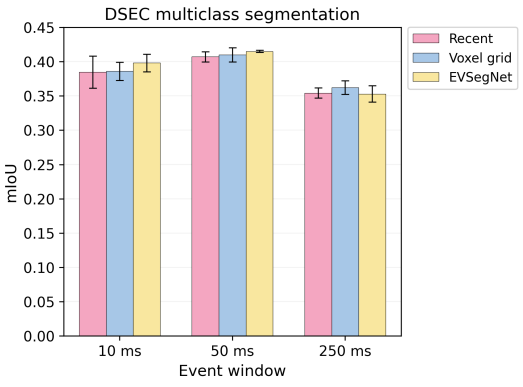

Figure 4: DSEC multiclass segmentation performance across event-window lengths. Each bar shows the mean mIoU over seeds 42, 43, and 44 for one event representation, and the error bars show one standard deviation.

The DSEC experiment was first used to evaluate whether event representations preserve semantic information about the driving scene. Table 1 reports the aggregate multiclass segmentation results, while Table 2 reports the class-level IoU values. Table 3 reports the binary human-only segmentation results. All aggregate values are reported as mean

± standard deviation over seeds 42, 43, and 44. Since mIoU, class IoU, human IoU, and accuracy are recovery metrics, higher values indicate stronger information recovery from the event representation.

For the DSEC multiclass task, the strongest semantic recovery is mainly obtained by EVSegNet and voxel grids. EVSegNet reaches the highest mIoU at 10 ms and 50 ms, while voxel grids obtain the highest mIoU at 250 ms. This pattern suggests that semantic leakage is strongest when the representation preserves more than only the most recent event at each pixel. EVSegNet combines event counts with timestamp statistics, so it gives the model both spatial activity and a compact summary of temporal variation. Voxel grids preserve temporal bins directly, which may become more useful at longer windows because motion traces are distributed across time. Recent is generally weaker because it keeps only local recency information, which is enough to preserve contours, but less informative than the richer temporal summaries used by EVSegNet and voxel grids.

The per-class IoU values in Table 2 show that the leakage is not equally distributed across semantic categories. Flat and background regions are recovered much more reliably than smaller or less frequent classes. Across representations and time windows, flat-region IoU remains high, roughly between 0.887 and 0.936, and background IoU remains between 0.784 and 0.822. In contrast, object, vegetation, human, and vehicle IoUs are much lower. The human class is especially weak in the multiclass setup, with IoU values between 0.018 and 0.032. Therefore, the DSEC multiclass experiment mainly shows leakage of coarse scene layout, road structure, and background structure, rather than reliable human localisation.

Table 2: Per-class IoU for DSEC multiclass segmentation. Values are means over seeds 42, 43, and 44. Bold indicates the strongest class IoU for each time window; green bold indicates the weakest leakage score.

Representation	Time window	Flat	Background	Object	Vegetation	Human	Vehicle
Recent	10 ms	**0.913**	**0.784**	0.148	0.284	0.027	0.154
Recent	50 ms	0.931	**0.813**	**0.155**	0.345	**0.021**	**0.176**
Recent	250 ms	0.889	**0.792**	0.107	0.197	**0.018**	**0.123**
Voxel grid	10 ms	0.916	0.797	**0.147**	**0.280**	**0.032**	**0.144**
Voxel grid	50 ms	**0.936**	**0.822**	0.158	**0.333**	**0.031**	**0.181**
Voxel grid	250 ms	**0.900**	**0.811**	**0.098**	**0.205**	**0.025**	**0.135**
EVSegNet	10 ms	**0.920**	**0.805**	**0.166**	**0.302**	**0.025**	**0.171**
EVSegNet	50 ms	**0.930**	0.821	**0.173**	**0.366**	0.024	**0.176**
EVSegNet	250 ms	**0.887**	0.804	**0.122**	**0.159**	0.023	**0.123**

The DSEC binary human-only results in Table 3 are much less informative for privacy analysis. Human IoU remains close to zero for all representations and time windows, even when pixel accuracy is very high. For example, voxel grids obtain the strongest human IoU, but only reach at most 0.027 ± 0.034 at 10 ms. At the same time, all representations obtain accuracies close to one. This mismatch shows that accuracy is dominated by background pixels and is therefore not a reliable indicator of human localization leakage in this setup.

Table 3: DSEC human-only segmentation results. Values are mean ± standard deviation over seeds 42, 43, and 44. Bold indicates the strongest segmentation score for each metric and time window; green bold indicates the weakest leakage score.

Representation	Time window	Human IoU ↑	Acc. ↑
Recent	10 ms	**0.009 ± 0.015**	**0.999 ± 0.001**
Recent	50 ms	**0.007 ± 0.013**	**0.999 ± 0.000**
Recent	250 ms	**0.000 ± 0.000**	**1.000 ± 0.000**
Voxel grid	10 ms	**0.027 ± 0.034**	**0.995 ± 0.007**
Voxel grid	50 ms	**0.023 ± 0.028**	**0.994 ± 0.007**
Voxel grid	250 ms	**0.024 ± 0.032**	**0.996 ± 0.005**
EVSegNet	10 ms	**0.009 ± 0.008**	0.997 ± 0.003
EVSegNet	50 ms	0.016 ± 0.015	0.998 ± 0.002
EVSegNet	250 ms	0.002 ± 0.001	0.999 ± 0.001

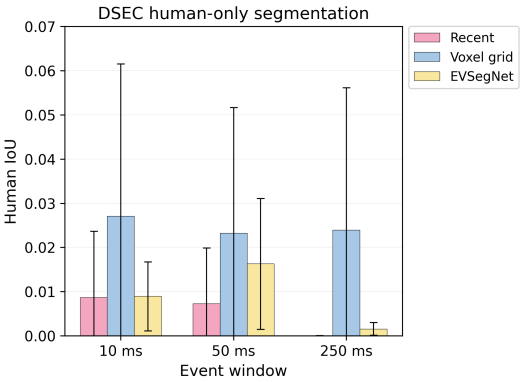

Figure 5: DSEC human-only segmentation performance across event-window lengths. Each bar shows the mean human IoU over seeds 42, 43, and 44 for one event representation, and the error bars show one standard deviation.

The high standard deviations for human IoU, relative to the mean values, further suggest that the DSEC human-

only task is unstable. The human class is too sparse in the selected DSEC split for this experiment to provide a strong conclusion about human privacy leakage. Therefore, the DSEC segmentation results should be interpreted mainly as evidence of semantic scene layout leakage. The human-specific privacy question is addressed more directly in the PEDRo experiment, which is designed around person annotations and is therefore more suitable for evaluating human location leakage.

The PEDRo results in Table 4 show a clear difference between the three evaluated event representations. Across all time windows, EVSegNet obtains the highest human-box IoU and accuracy, while the voxel-grid representation obtains the weakest scores. Recent performs between these two representations and remains close to EVSegNet, especially at shorter windows. This indicates that EVSegNet preserves the most human-related information in this setup, while the voxel grid leaks the least human-location information according to the evaluated human-box IoU metric.

The effect of the time window is also representation-dependent. Human-box IoU increases for all three representations as the window becomes longer, which suggests that accumulating events over a longer period gives the model more complete spatial information about the person. Nonetheless, the increase is strongest for the voxel grid: it improves from 0.377 at 10 ms to 0.535 at 40 ms. This suggests that the voxel grid needs enough temporal accumulation before it becomes useful for human-box segmentation. Recent and EVSegNet already perform relatively well at 10 ms and improve more moderately with longer windows.

From a privacy perspective, these results suggest that longer temporal windows increase human location leakage in PEDRo, because they allow the model to recover the human box more accurately. The lowest observed leakage is obtained with the voxel grid at 10 ms, while the strongest human-box recovery is obtained with EVSegNet at 40 ms. Therefore, the PEDRo results show a privacy-utility trade-off: representations and windows that improve human-box segmentation utility also expose more information about the approximate presence and location of people.

Table 4: PEDRo human-box segmentation results. Values are mean ± standard deviation over seeds 42, 43, and 44. Bold indicates the strongest segmentation score for each metric and time window; green bold indicates the weakest leakage score.

Representation	Time window	Human IoU ↑	Acc. ↑
Recent	10 ms	0.523 ± 0.017	0.908 ± 0.002
Recent	20 ms	0.548 ± 0.015	0.909 ± 0.002
Recent	40 ms	0.556 ± 0.011	0.907 ± 0.002
Voxel grid	10 ms	**0.377 ± 0.006**	**0.858 ± 0.004**
Voxel grid	20 ms	**0.469 ± 0.006**	**0.886 ± 0.003**
Voxel grid	40 ms	**0.535 ± 0.006**	**0.902 ± 0.004**
EVSegNet	10 ms	**0.529 ± 0.014**	**0.911 ± 0.002**
EVSegNet	20 ms	**0.560 ± 0.003**	**0.913 ± 0.004**
EVSegNet	40 ms	**0.573 ± 0.008**	**0.913 ± 0.007**

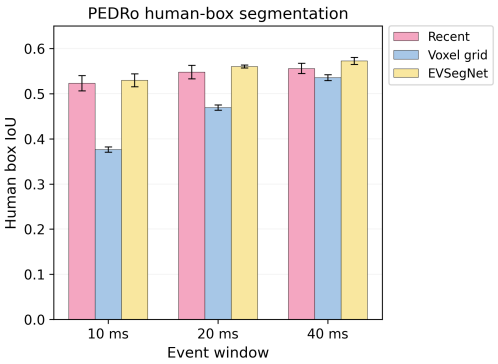

Figure 6: PEDRo human-box segmentation performance across event-window lengths. Each bar shows the mean Human IoU over seeds 42, 43, and 44 for one event representation, and the error bars show one standard deviation.

Accuracy is less informative than human-box IoU in this experiment. All representations reach relatively high accuracy values because the background occupies a large part of each binary mask. A model can therefore obtain high pixel accuracy even when it segments the human-box region less precisely. For this reason, human-box IoU is the more relevant metric for evaluating human location leakage in the PEDRo experiment.

These results should still be interpreted cautiously. PEDRo provides clearer evidence of human location leakage than the DSEC human-only experiment, but the target is a filled bounding box rather than a precise human silhouette. The experiment therefore measures whether event representations allow approximate human localization estimation, not whether they reveal body shape or fine-grained appearance.

The dataset level statistics in Table 5 make the difference between the DSEC and PEDRo human experiments more concrete. The difference between the DSEC and PEDRo human-binary results shows why the PEDRo experiment is needed. In DSEC, the human class is very sparse, and the binary human-only task produces human IoU values close to zero for all representations. This means that DSEC is useful for analyzing broader semantic scene layout leakage, but it isn't sufficient for drawing strong conclusions about human-location leakage. PEDRo addresses this limitation because it is centered on people and provides more direct person annotations, which

leads to substantially higher human IoU values and clearer differences between representations and time windows. Nonetheless, the target types are not equivalent: DSEC uses semantic human pixels, while PEDRo uses bounding boxes converted into filled box masks. Since these boxes also include background around the person, the PEDRo experiment measures leakage of approximate human location and not precise body shape or appearance.

Table 5: Human-target coverage in the evaluated DSEC and PEDRo splits. DSEC uses semantic human pixels, while PEDRo uses filled box-mask pixels generated from bounding-box annotations.

Split	Target type	Samples	Human samples	Human samples (%)	Human pixels / all pixels (%)	Median human area if present (%)
DSEC train	semantic human pixels	3925	2236	56.97	0.09	0.08
DSEC val	semantic human pixels	787	364	46.25	0.02	0.03
DSEC test	semantic human pixels	379	161	42.48	0.04	0.04
PEDRo train	filled human box-mask pixels	19228	19228	100.00	11.58	8.17
PEDRo val	filled human box-mask pixels	3950	3950	100.00	14.16	11.44
PEDRo test	filled human box-mask pixels	3823	3822	99.97	14.63	11.80

4.2 Depth Estimation Experiment

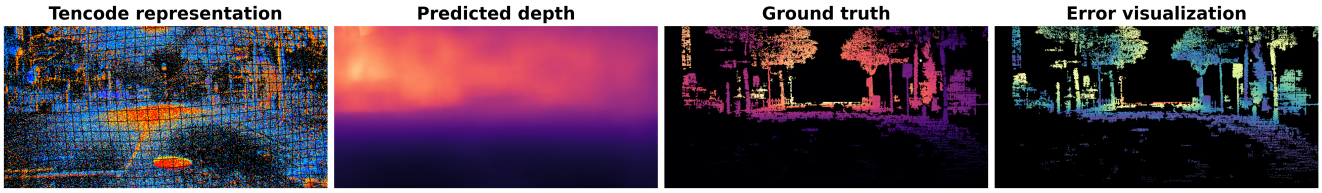

Figure 7: Qualitative depth estimation example for a single validation sample from a DSEC driving sequence using DepthAnyEvent-R. From left to right: Tencode event representation, predicted depth visualization, DSEC ground-truth depth visualization, and absolute-error visualization. The input is a Tencode RGB-like event representation. The prediction is shown as a dense depth estimate, while the ground truth is semi-dense; black regions mainly indicate invalid or unavailable depth values. The error visualization shows the absolute difference between prediction and ground truth where valid depth is available.

The results in Table 6 show that event data preserves recoverable geometric information. The mean AbsRel is 0.203, meaning that the predicted depth differs from the ground truth by roughly 20.3% on average relative to the true depth. Similarly, the mean RMSE is 8.853 m and the mean absolute error is 6.359 m, which indicate that metric depth is still noisy. Nonetheless, these errors should be interpreted in the context of outdoor driving scenes, where depth ranges can be large and distant regions naturally increase absolute error.

The threshold metrics further show that the predictions contain meaningful depth structure. The mean δ_1 is 0.665, meaning that about two thirds of valid pixels are predicted within the strictest threshold. The higher thresholds are even stronger: $\delta_2 = 0.916$ and $\delta_3 = 0.975$. This means that most pixels are still placed within a broader but meaningful depth range. Therefore, even when the exact metric distance is imperfect, the model often recovers the correct coarse scene geometry, such as which regions are near or far away.

Overall, the scores aren't accurate enough to claim perfect metric reconstruction, but they are strong enough to show that depth cues remain available in the event stream. This supports the broader conclusion that event cameras shouldn't be treated as inherently privacy preserving, since downstream models can still recover spatial structure from their outputs.

The qualitative example in Figure 7 supports this interpretation. The prediction does not reproduce the ground truth perfectly, and the error map shows visible local mistakes. Still, the predicted map preserves a recognizable depth structure of the driving scene. This means that the Tencode event representation can support inference about the three-dimensional arrangement of the scene when processed by a capable pretrained depth model.

The sequence-level variation in Table 6 further shows that depth leakage is scene-dependent. Some sequences allow stronger recovery of depth structure, while others produce higher errors. This prevents a stronger claim that all event streams expose spatial layout equally.

This experiment also has an important methodological limit. Since only one pretrained model and one depth representation are evaluated, the results don't isolate whether Tencode leaks more spatial information than other

event representations. The experiment is therefore best understood as a strong-model leakage probe. It shows what can be recovered from event data when a capable event depth model is available, rather than providing a complete comparison of all possible privacy preserving event representations.

Table 6: Depth estimation results on the evaluated DSEC sequences. Bold indicates the strongest depth estimation score; green bold indicates the weakest depth estimation score.

Sequence	AbsRel↓	SqRel↓	RMSE↓	$\text{RMSE}_{\log}$ ↓	SIlog↓	Mean err.↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
interlaken_00_f	0.183	**0.062**	8.537	**0.223**	**0.052**	6.306	0.694	**0.953**	**0.991**
interlaken_00_g	0.190	0.063	7.973	0.257	0.074	5.768	0.680	0.931	0.979
thun_00_a	0.192	0.072	8.123	0.254	0.070	**5.415**	0.695	0.920	0.976
zurich_city_05_a	0.192	0.064	8.372	0.251	0.066	5.722	0.678	0.925	0.979
zurich_city_05_b	0.198	0.076	8.443	0.258	0.069	5.930	0.676	0.914	0.975
zurich_city_06_a	**0.261**	**0.140**	9.783	**0.334**	**0.121**	6.967	**0.580**	**0.843**	**0.942**
zurich_city_07_a	**0.179**	0.065	8.101	0.241	0.063	5.804	**0.719**	0.935	0.980
zurich_city_08_a	0.190	0.067	**7.871**	0.248	0.065	5.449	0.688	0.926	0.980
zurich_city_09_d	0.215	0.088	10.562	0.263	0.072	8.028	0.632	0.914	0.980
zurich_city_10_b	0.230	0.110	**10.764**	0.289	0.088	**8.204**	0.613	0.895	0.971
Average	0.203	0.081	8.853	0.262	0.074	6.359	0.665	0.916	0.975

4.3 Optical flow Estimation Experiment

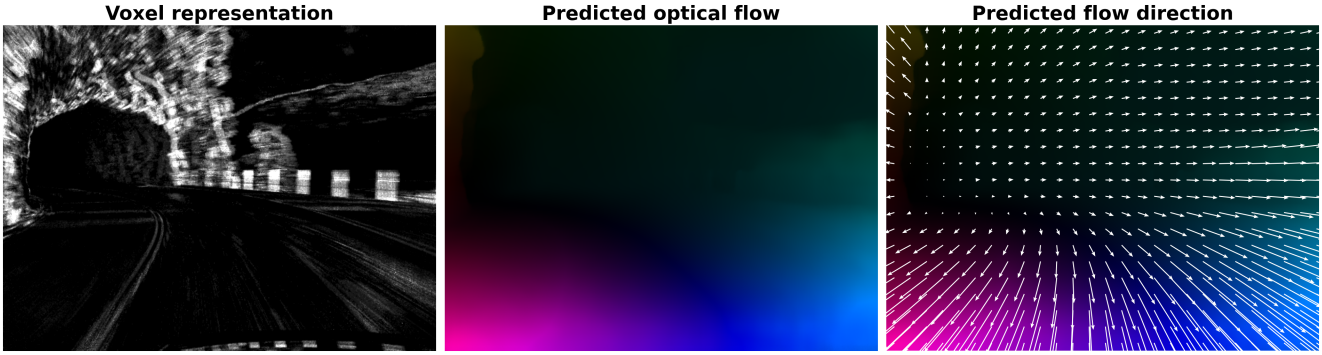

Figure 8: Qualitative optical flow result for a sample from the DSEC sequence `interlaken_00_b`. From left to right: grayscale visualization of the 15-bin voxel representation used as input to the pretrained IDNet model; decoded dense optical flow prediction, where colour encodes the predicted motion direction and magnitude; sparsely sampled flow vectors on the same prediction

The results in Table 7 report the optical flow performance of IDNet on the DSEC test sequences. Since all metrics are error-based, lower values indicate better performance. On average, IDNet obtains an EPE of 0.771 and an angular error of 3.000, showing that event data contains enough temporal and motion information to support accurate optical flow estimation. Optical flow evaluates whether the event stream preserves information about how scene regions move over time. This is important because motion can reveal information that is not captured by static scene structure, such as trajectories, relative movement, and dynamic interaction between the camera and surrounding objects.

The results show that this motion information is recoverable from the voxel representation used by IDNet. The qualitative example in Figure 8 also makes the privacy implication visible: the input is not a conventional RGB image, but the predicted flow still forms a dense motion field with coherent direction patterns. This means that the event stream preserves temporal structure that can be transformed into interpretable motion information by a pretrained event-based model.

At the same time, the results should not be interpreted as evidence of perfect motion recovery. The differences between sequences results in Table 7 show that motion leakage varies across scenes. This variation is expected because event-based optical flow depends on event density, camera motion, object motion, and the visibility of spatiotemporal traces in the event stream. The main privacy conclusion is that dense motion cues can be inferred from event data without reconstructing RGB frames with variable performance.

Table 7: DSEC optical flow results using IDNet. Arrows indicate the preferred direction for better optical flow estimation. Black bold indicates the strongest sequence-level result for each metric, while green bold indicates the weakest result, corresponding to the lowest motion leakage.

Sequence	1PE ↓	2PE ↓	3PE ↓	EPE ↓	AE ↓
interlaken_00_b	**18.351**	**8.279**	**5.319**	**1.367**	**2.383**
interlaken_01_a	14.225	5.209	2.894	0.833	2.445
thun_01_a	8.668	2.924	1.781	0.623	2.833
thun_01_b	8.519	2.674	1.466	0.608	2.442
zurich_city_12_a	11.805	2.259	**1.097**	0.617	**4.946**
zurich_city_14_c	17.022	6.054	2.361	0.703	3.771
zurich_city_15_a	**8.121**	**2.113**	1.112	**0.585**	2.801
Average	12.102	4.028	2.274	0.771	3.000

4.4 Cross-experiment Summary

Taken together, the three experiments suggest that event streams can preserve multiple complementary forms of structural information, although the strength of the evidence differs across tasks. The segmentation experiment provides moderate evidence of semantic leakage. In DSEC, the model mainly recovers large and persistent scene regions such as road and background, while recovery for smaller classes , such as human, remains weak. This can be a limitation of the built model and doesn't necessarily mean that a better leakage is unfeasible. Therefore, the DSEC segmentation results should be interpreted as evidence of coarse semantic scene-layout leakage. The PEDRo results give clearer evidence of human-location leakage, but only at the level of filled bounding-box masks, not precise body shape or identity.

The depth and optical-flow experiments show stronger recovery of non-semantic structure, but under a different condition: both use pretrained models designed specifically for their tasks. The depth experiment shows that the Tencode representation, when processed by DepthAnyEvent-R, preserves enough geometric information for coarse spatial layout estimation, even though the predictions don't recover exact distances in metres. The optical-flow experiment shows that the voxel representation, when processed by IDNet, preserves temporal motion traces that can be converted into dense motion fields. These results indicate that capable task-specific models can recover spatial and motion information from event data without image reconstruction.

The combined results therefore support a cautious conclusion about event-camera privacy. Event cameras reduce some appearance-based cues because they don't directly record RGB frames, colour, or texture. Nevertheless, the evaluated pipelines still recover several forms of structural information from event representations. Privacy in event-based vision should therefore be treated as a property of the full processing pipeline, including the representation, temporal window, model, task, and dataset, rather than as a guarantee provided by the sensor alone.

5 Responsible Research

5.1 Ethics

The main ethical concern of this project is that the same results that help understand privacy risks could also be used to recover information from event camera data. This is important because event cameras are increasingly used in settings such as autonomous driving, surveillance, robotics, and human-computer interaction. If event streams are stored, shared, or processed by external systems, they may reveal more information than expected.

This work doesn't try to identify people, reconstruct faces, perform person re-identification, or infer personal attributes. The human related experiments are limited to segmentation tasks. In DSEC, the human class is evaluated through semantic segmentation and in PEDRo, bounding boxes are converted into filled binary masks, so the experiment measures approximate human location leakage rather than detailed body shape, identity, or appearance.

All experiments use existing public datasets, and no new recordings or personal data were collected. This means that the main privacy implication of this work is that results show that even without RGB images, event data can approximately reveal where people are, how the scene is structured, how far objects are from the camera, and how scene regions move.

5.2 Bias

The conclusions of this project are limited by the datasets used. Almost all experiments are based on DSEC, which contains outdoor driving scenes. This means that the results are mainly valid for road environments and shouldn't be directly generalized to indoor monitoring, wearable cameras, or other event-camera applications. These settings might have different motion patterns, distances, object sizes, lighting conditions, and human activity.

The DSEC human experiments are also affected by class imbalance. Human pixels occupy only a very small part of the evaluated DSEC split. This explains why the DSEC human centered results have human IoU values close to zero while pixel accuracy remains close to one. In this case, high accuracy mostly means that the model predicts the dominant background class correctly. Therefore, DSEC is more useful for studying general scene-layout leakage than for drawing strong conclusions about human location leakage.

PEDRo reduces this limitation because it is centered on person detection and contains person annotations in almost all samples. Nonetheless, PEDRo uses bounding boxes rather than precise human masks. After converting these boxes into filled masks, the target also includes background around the person. For this reason, PEDRo supports conclusions about approximate human location leakage, but not about precise human silhouette leakage.

There is also a bias in the choice of representations and models. The selected event representations and pretrained models are already known to work well for event-based vision tasks. This is intentional, because the project studies what information can be recovered under capable downstream processing. This means that the results shouldn't be interpreted as the leakage level of every possible event-camera system, because shorter windows, weaker models, or more privacy-oriented representations might expose less information.

5.3 Reproducibility

To make the experiments reproducible, the methodology (see section 3) describes the datasets, selected sequences, event representations, temporal windows, models, training settings, evaluation metrics, and software environments. For the segmentation experiments, the same train, validation, and test splits are used across representations. The same model architecture, optimization settings, and evaluation metrics are also kept fixed.

The segmentation experiments are repeated with three random seeds: 42, 43, and 44. These seeds control random parts of training such as initialization, data shuffling, and augmentation. The reported segmentation results are therefore shown as mean and standard deviation over these runs.

The depth and optical-flow experiments use released pretrained models without additional training or fine-tuning. For depth estimation, reproducibility mainly depends on the DepthAnyEvent-R checkpoint, the selected DSEC sequences, and the software environment. For optical flow, reproducibility mainly depends on the IDNet checkpoint, the DSEC-Flow input sequences, and the benchmark evaluation protocol.

Some small differences may still happen because of GPU behavior, CUDA versions, library versions, or nondeterministic operations in deep-learning frameworks. Still, the main comparison variables are kept fixed, and the segmentation code used in this project is made available in the project repository.

5.4 AI Usage

ChatGPT-5.5 was used during the project as a support tool. For coding, it was used to adapt publicly available code to a more simplified version, generate methods based on explicitly specified ideas, help debug errors, simplify parts of the implementation, generate or improve visualization scripts, and writing documentation. For writing, it was used to proofread paragraphs, suggest clearer phrasing based on draft ideas, and help improve the structure of some sections.

The AI output was used as a suggestion and not as a source of experimental evidence. The implementation decisions, used datasets, metrics and representations weren't prescribed by AI, but were inspired from related works. The datasets, experimental setup, metrics, results, tables, and figures were checked manually against the implemented code and the actual outputs. The AI did not collect data, label data, run experiments independently, or decide which results to report. The final design choices, interpretations, and claims remain the responsibility of the author.

6 Conclusion

This paper investigated what types of structural scene information can be inferred from event-based camera data without explicitly reconstructing standard images. The results show that event cameras should **not** be considered

inherently privacy-preserving. Even though event streams don't directly contain RGB frames, they can still preserve information that is useful for semantic segmentation, depth estimation, and optical flow estimation.

The segmentation experiments show that semantic leakage is present but uneven. On DSEC, large and persistent scene regions such as road, background, and vehicles are recovered more reliably than humans. The binary DSEC human experiment is less conclusive because the human class is very sparse, leading to human IoU values close to zero despite high pixel accuracy. The PEDRo experiment gives clearer evidence of human-location leakage: human box masks can be recovered from event representations, especially when longer temporal windows are used.

The depth estimation experiment shows that event data also preserves spatial layout information. The predicted depth maps are not perfect metric reconstructions, but they recover enough coarse scene geometry to indicate leakage of distance and layout cues. The optical flow experiment further shows that event streams preserve motion information, since a pretrained event-based model can estimate dense motion patterns from the data.

Overall, the main conclusion is that privacy leakage in event-based vision is representation and task-dependent. Some representations and longer time windows preserve more useful information for downstream models, which also means they may leak more privacy-relevant structure. Future work should evaluate more datasets, more event representations, and stronger privacy attacks, including reconstruction-free attacks on human identity, behavior, and activity. It should also investigate privacy preserving event representations that reduce semantic, spatial, and motion leakage while keeping enough utility for safe computer-vision applications.

References

[1] Shafiq Ahmad et al. "Event-driven re-id: A new benchmark and method towards privacy-preserving person re-identification". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2022, pp. 459–468. DOI: 10.1109/WACVW54805.2022.00052.

[2] Mira Adra et al. "Event-based solutions for human-centered applications: a comprehensive review". In: *Frontiers in Signal Processing* 5 (2025), p. 1585242. DOI: 10.3389/frsip.2025.1585242.

[3] Henri Rebecq et al. "Events-to-video: Bringing modern computer vision to event cameras". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3857–3866. DOI: 10.1109/CVPR.2019.00398.

[4] Bingquan Zhou and Jie Jiang. "Deep event-based object detection in autonomous driving: a survey". In: *arXiv preprint arXiv:2405.03995* (2024). DOI: 10.48550/arXiv.2405.03995.

[5] Jiahui Yuan et al. "Event-based head pose estimation: Benchmark and method". In: *European Conference on Computer Vision*. Springer. 2024, pp. 191–208. DOI: 10.1007/978-3-031-72633-0_11.

[6] Junho Kim et al. "Privacy-preserving visual localization with event cameras". In: *IEEE Transactions on Image Processing* 34 (2025), pp. 6215–6230. DOI: 10.1109/TIP.2025.3607640.

[7] Nicholas FY Chen. "Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 644–653. DOI: 10.1109/CVPRW.2018.00107.

[8] José Ramón Padilla-López, Alexandros Andre Chaaraoui, and Francisco Flórez-Revuelta. "Visual privacy protection methods: A survey". In: *Expert Systems with Applications* 42.9 (2015), pp. 4177–4195. DOI: 10.1016/j.eswa.2015.01.041.

[9] Andy Catruna, Adrian Cosma, and Emilian Radoi. "Gaitpt: Skeletons are all you need for gait recognition". In: *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*. IEEE. 2024, pp. 1–10. DOI: 10.1109/FG59268.2024.10581947.

[10] Jan Malte Hilgefort, Daniel Arp, and Konrad Rieck. "Spying through virtual backgrounds of video calls". In: *Proceedings of the 14th ACM workshop on artificial intelligence and security*. 2021, pp. 135–144. DOI: 10.1145/3474369.3486870.

[11] Robert Templeman et al. "PlaceRaider: Virtual theft in physical spaces with smartphones". In: *arXiv preprint arXiv:1209.5982* (2012). DOI: 10.48550/arXiv.1209.5982.

[12] Bowen Du et al. "Event encryption for neuromorphic vision sensors: Framework, algorithm, and evaluation". In: *Sensors* 21.13 (2021), p. 4320. DOI: 10.3390/s21134320.

[13] Inigo Alonso and Ana C Murillo. "EV-SegNet: Semantic segmentation for event-based cameras". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 1624–1633. DOI: 10.1109/CVPRW.2019.00205.

[14] Luca Bartolomei et al. "Depth AnyEvent: A Cross-Modal Distillation Paradigm for Event-Based Monocular Depth Estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025, pp. 19669–19678. DOI: 10.1109/ICCV51701.2025.01829.

[15] Yilun Wu, Federico Paredes-Vallés, and Guido CHE De Croon. "Lightweight event-based optical flow estimation via iterative deblurring". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 14708–14715. DOI: 10.1109/ICRA57147.2024.10610353.

[16] Alex Zihao Zhu et al. "Unsupervised event-based learning of optical flow, depth, and egomotion". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 989–997. DOI: 10.1109/CVPR.2019.00108.

[17] Luming Wang et al. "Egoevgesture: Gesture recognition based on egocentric event camera". In: *arXiv preprint arXiv:2503.12419* (2025). DOI: 10.48550/arXiv.2503.12419.

[18] Daniel Gehrig et al. "End-to-end learning of representations for asynchronous event-based data". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 5633–5643. DOI: 10.1109/ICCV.2019.00573.

[19] WeiJie Bai et al. "Accurate and efficient frame-based event representation for AER object recognition". In: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2022, pp. 1–6. DOI: 10.1109/IJCNN55064.2022.9892070.

[20] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. "Learning monocular dense depth from events". In: *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 534–542. DOI: 10.1109/3DV50981.2020.00063.

[21] Shafiq Ahmad, Pietro Morerio, and Alessio Del Bue. "Event anonymization: Privacy-preserving person re-identification and pose estimation in event-based vision". In: *IEEE Access* 12 (2024), pp. 66964–66980. DOI: 10.1109/ACCESS.2024.3399539.

[22] Katharina Bendig et al. "Anonynoise: anonymizing event data with smart noise to outsmart re-identification and preserve privacy". In: *Proceedings of the Winter Conference on Applications of Computer Vision*. 2025, pp. 3159–3161.

[23] Mira Adra and Jean-Luc Dugelay. "E2PRIV: Privacy-Preserving Event-to-Video Reconstruction with Face Anonymization". In: *2025 13th International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2025, pp. 1–6. DOI: 10.1109/IWBF63717.2025.11113401.

[24] Mathias Gehrig et al. "Dsec: A stereo event camera dataset for driving scenarios". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4947–4954. DOI: 10.1109/LRA.2021.3068942.

[25] Zhaoning Sun et al. "Ess: Learning event-based semantic segmentation from still images". In: *European Conference on Computer Vision*. Springer. 2022, pp. 341–357. DOI: 10.1007/978-3-031-19830-4_20.

[26] Chiara Boretti et al. "Pedro: an event-based dataset for person detection in robotics". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 4065–4070. DOI: 10.1109/CVPRW59228.2023.00426.

[27] Mathias Gehrig et al. "E-raft: Dense optical flow from event cameras". In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 197–206. DOI: 10.1109/3DV53792.2021.00030.

A More information about Representations Implementation

The voxel-grid representation is implemented with separate temporal bins for positive and negative events. With five temporal bins, this results in a ten-channel tensor: five channels for positive events and five channels for negative events. This design was used to preserve polarity-specific information and to avoid cancellation between positive and negative events within the same temporal bin. Although this differs from more compact signed voxel-grid formulations [16, 14], it is consistent with the goal of evaluating semantic leakage under a high-information representation. In this setting, preserving polarity separately gives the segmentation model access to more of the structure contained in the event stream.

The recent timestamp representation stores polarity-specific event counts together with the most recent event timestamp at each pixel. This allows the model to use both where events occurred and how recently they occurred. The EV-SegNet-style representation stores polarity-specific event counts together with timestamp statistics, resulting in a six-channel tensor that summarizes both event density and temporal structure.

B More information about the Training Protocol

All segmentation models are trained with the Adam optimizer, an initial learning rate of 10^{-4}, and a polynomial learning-rate decay schedule. Training is performed for up to 12 epochs with a micro-batch size of 8, one gradient accumulation step, and an effective batch size of 8. Validation is performed once per epoch. Early stopping is applied with a patience of four validation evaluations and a minimum improvement threshold of 0.002.

During training, the validation set is used only for model selection. The best checkpoint is selected according to the validation metric that matches the experiment goal: mean Intersection over Union for the DSEC multiclass setup and human IoU for the human-focused setups. The test set is then used only for the final evaluation.