Event Quality Score (EQS): Assessing the Realism of Simulated Event Camera Streams via Distances in Latent Space

Kaustav Chanda, Aayush Atul Verma, Arpitsinh Vaghela, Yezhou Yang, Bharatesh Chakravarthi
Arizona State University
{kchanda3, averma90, avaghel3, yz.yang, bshettah}@asu.edu

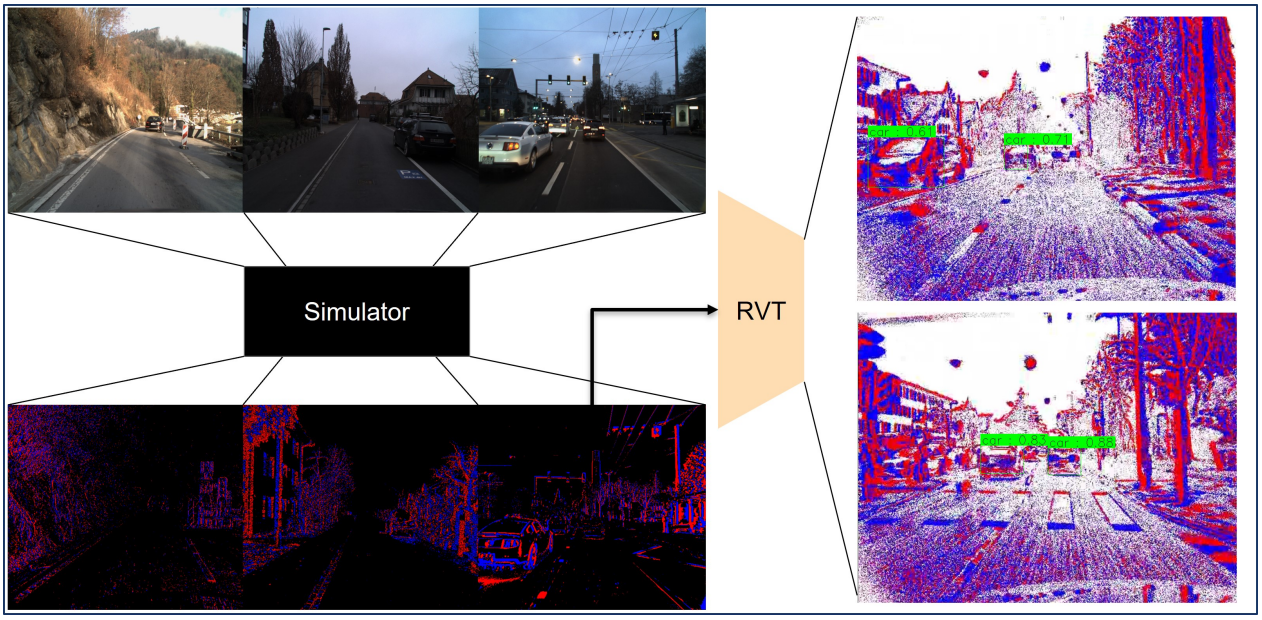

Figure 1. Schematic representation of a recurrent vision transformer (RVT) model trained on simulated event data and evaluated on real event streams.

Abstract

Event cameras promise a paradigm shift in vision sensing with their low latency, high dynamic range, and asynchronous nature of events. Unfortunately, the scarcity of high-quality labeled datasets hinders their widespread adoption in deep learning-driven computer vision. To mitigate this, several simulators have been proposed to generate synthetic event data for training models for detection and estimation tasks. However, the fundamentally different sensor design of event cameras compared to traditional frame-based cameras poses a challenge for accurate simulation. As a result, most simulated data fail to mimic data captured by real event cameras. Inspired by existing work on using deep features for image comparison, we introduce event quality score (EQS), a quality metric that utilizes activations of the RVT architecture. Through sim-to-real experiments on the DSEC driving dataset, it is shown that a higher EQS implies improved generalization to real-world data after training on simulated events. Thus, optimizing for EQS can lead to developing more realistic event camera simulators, effectively reducing the simulation gap. EQS is available at https://github.com/eventbasedvision/EQS .

1. Introduction

Event cameras [5] are neuromorphic sensors inspired by the human retina, where rods and cones activate independently based on the light stimulus received in real-time. Analogous to this mode of operation, event cameras asynchronously register pixel-level changes in the scene with precise timestamps in the order of microseconds. This results in a spatially sparse but temporally dense data stream. Additionally,

the absence of a fixed exposure time frame makes these sensors resilient to drastic changes in lighting conditions with a dynamic range in the region of 120 dB [27]. These characteristics make them ideal for vision applications in challenging tasks such as object detection in low-light [4, 25, 30], over-exposure [12], high-speed motor control [29, 34], and autonomous landing and flight [36, 45].

A single pixel of an event camera consists of three components [27]: a photo receptor circuit, a differencing circuit, and a comparator. The photo-receptor circuit converts photocurrent logarithmically into voltage, and the difference between these values is compared against global thresholds in the comparator to determine when to fire an event. Through this process, shot noise from photons, thermal noise from transistors, and other non-idealities in clocking and quantization [9] impart a stochastic nature to the data captured by event cameras that exacerbates the challenge for models trained on simulated data to generalize.

One of the key enabling factors for the deep learning revolution in computer vision has been the availability of large-scale image datasets captured using widely available sensors in cameras and smartphones. Due to their limited adoption across the industry, there is an acute scarcity of large, high-quality datasets captured using these sensors. To mitigate this limitation, several event camera simulators have been proposed [3, 7, 14, 16, 20, 28, 32]. These simulators have been utilized both for generating new datasets such as SEVD [1] as well as event-based counterparts of existing image datasets such as Event-KITTI [26], Yelan−Syn [48], ESfP−Synthetic [33], N−EPIC Kitchen and N−ImageNet [23]. Although these datasets have motivated much of the initial research in event vision, the efficiency of models trained using synthetic datasets on real-world data has not been thoroughly investigated.

The primary difficulty in characterizing event streams arises from how event data is structured. An event stream is an array of $\langle x, y, p, t \rangle$ tuples, each indicating that the log intensity at the pixel location (x, y) during the timestamp t changed by a factor greater than $pol \times C$ where C is a constant threshold for a given scene. Therefore, existing approaches convert event streams into an image-like $2D$ representation before applying well-known image similarity metrics such as structural similarity (SSIM) [46] or peak signal-to-noise ratio (PSNR). SSIM and PSNR have been extensively studied, and it has been shown in [43] that not only are they unsuitable for directly comparing event frames but also produce misleading results for grayscale frames reconstructed from event streams with encoder-decoder models. Crucially, no measurement framework exists that computes a differentiable similarity score directly between two event streams without relying on an intermediate representation.

The LPIPS [47] metric proposed for image comparison has been shown to alleviate some of the aforementioned shortcomings of traditional similarity metrics. The authors show that the internal activations of networks trained for high-level tasks such as classification are capable of capturing the subtle differences in input data, such as blurriness, and that these activations correlate strongly with human perception. [39] utilized the LPIPS loss for training a UNet [40] architecture for reconstructing grayscale images from event frames, indicating that the activations of a VGG [41] model can serve as an effective loss function. [43] showed that the noise model used in simulated event data has a significant impact on model performance and that a simple zero-mean Gaussian noise to simulate background events and a few uniformly selected 'hot' pixels improved model performance for grayscale reconstruction from simulated events. However, these approaches focus solely on the accuracy of reconstructed image frames. Our framework directly evaluates the quality of event streams analogous to the approach used by [47] for frames without converting to a $2D$ image-like representation first.

The proposed *event quality score (EQS)* measure takes two event streams of arbitrary length as input and converts them to a 4-dimensional tensor following the same approach as [11] and passes them as input to a recurrent vision transformer network [11] pre-trained on event data for object detection. The activations from the first three convolution layers from both input event tensors are compared to yield a quality score that indicates how closely the two event streams match. If performed between simulated and real event streams, we show that this distance is strongly correlated with the simulation gap. The key contributions can be summarized as follows:

1. Propose EQS, which, to the best of our knowledge, is the first quantitative and fully differentiable metric that works directly on raw event streams and computes a distance measure.
2. Show that the EQS metric correlates strongly with how well a model trained on simulated data generalizes to real-world data.
3. Demonstrate the importance of bringing simulated events closer to real data, both qualitatively as well as quantitatively in terms of model performance.

The rest of the paper is structured as follows. Sec. 2.1 provides an overview of existing event camera simulators and the methodology adopted by each for generating event streams. Sec. 2.2 discusses prior works utilizing deep features for analyzing image data. Sec. 3 highlights the significance of the simulation gap by examining the drop in performance when testing on real data after training with simulated data. The theoretical details regarding how EQS is computed are explained in Sec. 4, and the Sec. 5 presents empirical results and subsequent analysis for the DSEC [12] dataset.

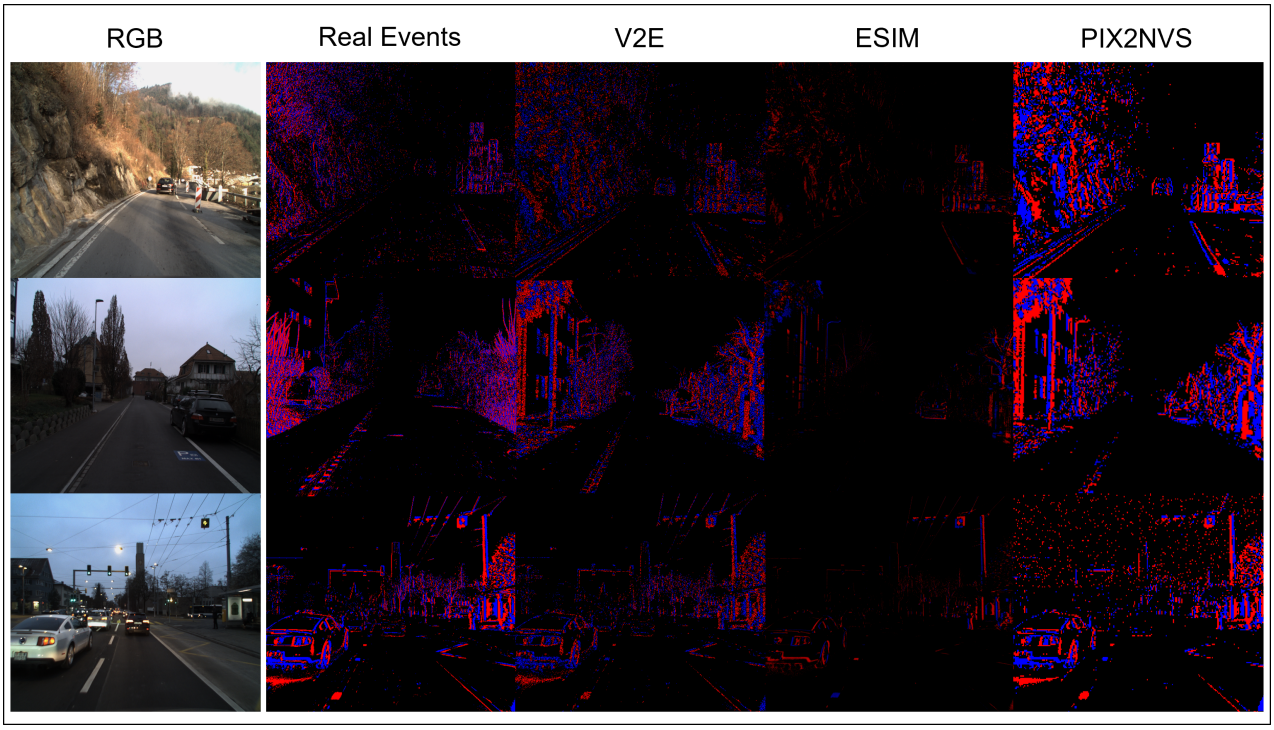

Figure 2. Simulated events from the V2E [16], ESIM [38], and PIX2NVS [3] event camera simulators and real event camera data visualized as frames where red pixels indicate negative polarity and blue pixels indicate positive polarity.

2. Related Work

Initial work on event cameras generally utilized the DVS and DAVIS cameras, followed by more sophisticated sensors developed by Prophesee [37], iniVation [17], and Lucid Vision Labs [31]. Early research in event vision was dominated primarily by algorithms to create feature representations of event data. Representations such as time surfaces [24] and HATS [42] underscored much of the initial progress in creating machine learning models with event data. As these sensors became more widely available, real-world datasets such as DSEC [12], MVSEC [49] and 1Mpx [35], along with simulated datasets like SEVD [1] and eTraM [44] facilitated training and fine-tuning of larger models on event data.

2.1. Event Camera Simulators

Existing event camera simulators generally employ one of the following strategies: (1) Render entire $3D$ scenes in engines such as Blender, Unreal Engine [8] or CARLA [7], place a virtual event camera in the scene and sample events [38] [20] or frames [2] at frequencies similar to those of real event cameras to generate a set of events or (2) Use video or image sequences as input and generate event streams by thresholding log intensity differences [16] [3]. Recently, [50] proposed an end-to-end event generation model, and [13, 14, 20] developed physics-aware simulators by parameterizing the event generation process.

The ESIM [38] simulator takes a set of grayscale images as input and performs linear interpolation on the log-intensity values at each pixel to reconstruct a piecewise linear approximation of the underlying signal and adaptively samples this interpolated signal to simulate events. In contrast to previous works such as [32], which performed uniform sampling on this interpolated signal to generate events, ESIM samples frames adaptively based on the estimated motion field map or optical flow between sampled images. Effectively, it samples events at a higher frequency from sequences with fast motion and lower frequency when the image signal varies slowly. In the absence of ground truth camera trajectories, the adaptive sampling scheme based on pixel displacement detailed in [38] may be used, wherein the next sampling time t_{k+1} from t_k is chosen as:

$$t_{k+1} = t_k + \lambda_v \left| \mathcal{V}(x; t_k) \right|_m^{-1}$$

where $|\mathcal{V}| = \max_{x \in \Omega} |\mathcal{V}(x; t_k)|$ is the maximum magnitude of the motion field at time t_k and $\lambda_v = 0.5$ in the setup detailed in [38].

Instead of utilizing the motion between frames, the V2E [16] simulator seeks to realistically model the photon noise inherent in event cameras using a DVS pixel model that includes temporal noise, leak events, and a finite intensity-dependent bandwidth. The V2E toolbox converts RGB

Simulator	Open Source	Timestamp Sampling Method	Methodology
ESIM [38]	✓	Adaptive	Renders synthetic images and uses adaptive sampling for event timing based on brightness changes. Models contrast threshold noise.
Vid2E [10]	✓	Adaptive (via ESIM)	Extends ESIM by using SuperSloMo for temporal interpolation.
V2E [16]	✓	Frame	Incorporates bandwidth limitations, leak events, and Poisson shot noise for simulation.
CARLA [7]	✓	Frame	Incorporates uniform noise to pixel differences between consecutive image frames from the rendered $3D$ scene.
Prophesee Video to Events [37]	✗	Frame	Generates event streams using pixel-level interpolation.
PIX2NVS [3]	✓	Frame	Simulates events based on log-intensity or contrast-enhanced differences between successive frames.
DVS-Voltmeter [28]	✓	Circuit-based	Models voltage noise, photon noise, and temperature noise based on the circuitry of event cameras.
ADV2E [19]	✗	Stochastic	Simulates DVS circuit behaviors using continuity sampling and analogue low-pass filtering.
PECS [14]	✗	N/A	Works on $3D$ scenes in Blender. Models lens behavior, multispectral rendering, and photocurrent generation using ray tracing.
ICNS [20]	✓	Gaussian Noise	Performs latency estimation using a delayed filter and light-dependent time constant.
EventGAN [50]	✓	N/A	Learning-based simulator that utilizes Generative Adversarial Networks (GANs) to produce event frames.
Gazebo DVS Plugin [21]	✓	Frame	Threholds difference between consecutive frames of a video stream captured from sensors in the Gazebo $3D$ simulation.

Table 1. Summary of existing event camera simulators.

video frames to events by (1). Converting color to luma and (2). Optionally interpolating the luma frames to increase the temporal resolution of the video, (3). Performing a linear to logarithmic mapping by using a linear mapping for luma values $L < 20$ and a logarithmic scale for larger values. This arises from the observation that DVS pixel bandwidth is proportional to intensity for low photocurrents. This is followed by (4). The event generation model, where it is assumed that each pixel has a memorized brightness value L_m, and the number of events $N_e = \lfloor \frac{L_{lp} - L_m}{\theta} \rfloor$ where θ varies with a Gaussian distribution with a value drawn from $0.3 + \mathcal{N}(0, \sigma_\theta)$ with $\sigma_\theta = 0.03$. The V2E simulator also models *'hot pixels'*, which fire events even in the absence of intensity change and spontaneous ON events called *'leak noise events'* by continuously decreasing the value of L_m by a random amount at each pixel location.

Another relatively early approach, the pixel-to-NVS simulator (PIX2NVS) [3] generates event tuples $E_e = \langle x_e, y_e, t_e, P_e \rangle$ from a sequence of pixel-domain video frames $F_0, F_1, ..., F_N$. The RGB pixel values are first converted to luminance values using $y_{i,j} = 0.299R_{i,j} + 0.587G_{i,j} + 0.114B_{i,j}$, which are then converted to log-intensity using a log-linear mapping:

$$l_{i,j} = \begin{cases} y_{i,j}, & y_{i,j} \leq T_{\log} \\ \ln(y_{i,j}), & y_{i,j} > T_{\log} \end{cases}$$

where T_{log} threshold controls the switch between linear and log mapping. For lin-log intensity, it is set to a value approximately 10% of the maximum value. For event generation, the difference between luminance values and the average of the weighted neighborhood of luminance values in the previous frame is calculated, and the polarity is determined by the sign of this difference:

$$d_{i,j} = l_{i,j} - \frac{\sum_{p=0}^{1} l_{i+2p-1,j}[n-1] + \sum_{p=0}^{1} l_{i,j+2p-1}[n-1]}{4}$$

$$P_e = \begin{cases} ON, & \text{sgn}(d_{i,j}) = 1 \\ OFF, & \text{sgn}(d_{i,j}) = -1 \end{cases}$$

The corresponding event is generated if $|d_{i,j}|$ exceeds a fixed threshold and is added to the set of events between the two consecutive frames considered. The event data produced by each of the simulators is qualitatively compared in Fig. 2.

2.2. Deep Feature Spaces

In contrast to traditional frame-based data, event data are characteristically sparse in the spatial domain and much denser in the temporal domain. Thus, effective feature representations must capture both the spatial and temporal aspects of the event stream. The recently proposed recurrent vision transformer (RVT) [11] mixes spatial and temporal features, utilizing transformer layers for spatial feature extraction and recurrent neural networks for temporal feature extraction. Although only activations from the initial convolution layers have been taken for quality evaluation, it should be noted that the LSTM hidden states may also potentially be utilized for a stronger emphasis on temporal features.

It has been shown that feature space division [22] occurs when deep convolutional neural networks are trained on classification tasks, which leads to the activations diverging for each target class after training. It is reasonable to assume that the output of the penultimate layer would refer to a vector that captures the properties of the input in abstract dimensions, such as perceptual quality [47], aesthetics [18], or faithfulness to the text prompt [6]. More recently, this approach has been utilized to estimate the distances between the distributions of real images and those generated by an algorithm. The classical approach for this is the Frechet Inception Distance [15], which estimates the distance between a distribution of Inception-v3 features of real and generated images.

3. The Motivation

A qualitative visualization of real and simulated events is shown in Fig. 2. The visualization parameters were kept

constant for each event stream to aid a fair comparison, despite each simulator generating a different number of events for the same input image sequence. Qualitatively, it can be seen that the nature of noise in the real event camera data is significantly different from that of the artificial noise added by each simulator. It should be noted that the ESIM output appears darker due to a lower number of events generated at the same pixel position compared to the V2E and PIX2NVS simulators.

By keeping the number of events plotted constant, the objective is to bring out the differences in how the events are distributed in the simulated event streams compared to that of the Prophesee Gen 3.1 camera. We hypothesize that these differences in the noise models between real and simulated events cause deep learning models trained on simulated event data to overfit to the noise, which results in sub-optimal features being extracted when tested on real data. This is substantiated by establishing a baseline performance gap by training an RVT-small model on synthetic data produced by each simulator and testing on the actual DSEC test set. These results are summarized in Tab. 3. Notably, the higher error on the simulated test set for the model trained on PIX2NVS data indicates that the simulated data fails to represent objects of interest in the synthetic event stream. From the mAP scores reported in Tab. 3, it can be seen that the accuracy drops by around 50% when transitioning from simulated to real data for V2E and ESIM, and a much larger gap of about 69% in the case of PIX2NVS. Intuitively, a good event quality measure should be able to detect this and assign a lower score to the event stream generated by this simulator. It is demonstrated in Sec. 5 that the corresponding EQS values agree with this notion.

4. Event Quality Score (EQS)

To work around the severe lack of high-quality event camera data for common tasks in computer vision, such as pose estimation and object detection, several simulators have been proposed, such as V2E [16], ESIM [38], and end-to-end models like EventGAN [50] have been proposed. The quality of the data generated by such methods is usually demonstrated indirectly by showcasing the accuracy of the model trained on simulated data on some downstream task. However, this evaluation is also hindered by the lack of labeled real event camera data. To the best of our knowledge, no numerical metric exists that directly computes a similarity score between two sets of raw events. This brings us to the question: *How to meaningfully measure the distance between two event streams?*

The evaluation metric proposed works directly on the generated event streams. The benefits of directly comparing event streams are twofold: (1) Both spatial and temporal information can be considered in the evaluation as we do not convert events to an image-like representation, and (2) the metric becomes independent of the downstream task, such as object detection or action recognition. The features extracted by a recurrent vision transformer network are used to perform this computation. The mean cosine similarity between the extracted features from each input event stream is taken as the similarity score.

Let $E = \{e_{t_1}, e_{t_2}, ..., e_{t_n}\}$ represent a stream of events where each event $e_{t_k} = (x_k, y_k, t_k, p_k)$ is a tuple with polarity $p_k \in \{1, -1\}$ and $|t_n - t_1| \simeq \Delta T$. An event tensor representation is created from event streams in a manner similar to [11]. T temporal bins are created and the positive and negative polarity events that occur within a fixed time interval δt are accumulated separately to create a tensor X_1 with shape $(2T, H, W)$ where H and W are the height and width of the event camera sensor. This tensor representation is compatible with $2D$ convolutions and can be taken as input to any model with convolution operations, such as Yolo or recurrent vision transformer. The overall architecture for our framework is schematically shown in Fig. 3.

4.1. Latent Feature Similarity (LFS) Block

First, event tensors are created for real and simulated event data using the method described in Sec. 4 and each tensor is passed to a pre-trained, deep convolutional neural network. Our approach is built on the hypothesis that the sim-to-real gap for event-camera data can be captured in the representational space of the network's activations. The feature maps generated by the convolution layers of the model are extracted and the cosine distance across spatial and temporal features of the network are computed. Although the proposed method can be adapted for all the activations across a given model, we limit our analysis to convolution layers for interpretability.

Both real and simulated data are passed through a pre-trained recurrent vision transformer network, leveraging both its spatial and temporal feature extraction capabilities. Specifically, the LFS block takes these two sets of activations to compute a similarity score. We extract the feature maps produced by each convolution block and create a $1D$ tensor across depth channels by averaging over patches in the spatial domain. For the RVT model, the choice of LFS patch size for each scale is discussed in Sec. 5. Let Z_i be the output of the self-attention block for the i^{th} RVT block in the original formulation from [11]. As shown in Fig. 3 the RVT block at each scale i computes:

$$(h_i, c_i) = ConvLSTM(Z_i, h_{i-1}, c_{i-1})$$

$$o_i = \sigma(Conv(Z_i) + Conv(h_i))$$

$$X_{i+1} = h_i + Conv(\sigma(Conv(LN(h_t))))$$

where h_i and c_i represent the hidden state and cell states respectively, LN denotes layer normalization and o_i is the activation output from the i^{th} scale of the RVT block. Let $o_{i_{e1}}$

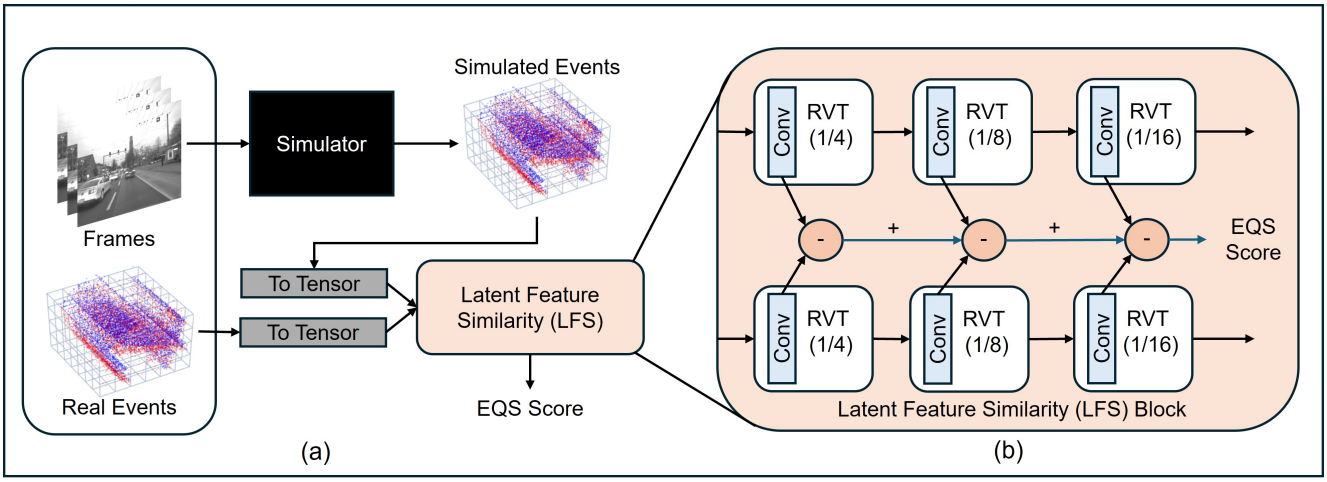

Figure 3. (a). Dataset with frames and their corresponding synchronized event streams. Simulated events are generated using frames as input, and both sets of events are composed into tensors as discussed in Sec. 4. (b). For each scale in the RVT model, cosine distances are calculated between the feature maps from corresponding convolution layers and aggregated to obtain the final score.

and $o_{i_{e2}}$ represent the activations from the i_{th} scale of the RVT block for two event streams $e1$ and $e2$ respectively. For each scale, a fixed patch p_i is taken as described in Sec. 5. For each non-overlapping patch, the following vector is created

$$V_{ei} = \left\{ \frac{1}{||p_i||} \sum_{p_i} o_{i_{ei}[p_i]} \right\}$$

such that $||V_{ei}||$ is the number of channels in $o_{i_{e1}}$. Thus, $\mathbf{V_{ei}}$ becomes a $1D$ vector for each patch. Taking the vectors V_{e1} and V_{e2}, cosine similarity is calculated as usual and subtracted from 1 to produce a distance metric for each patch.

$$CS_{p_i} = 1 - \frac{\mathbf{V_{e1}} \cdot \mathbf{V_{e2}}}{||\mathbf{V_{e1}}|| ||\mathbf{V_{e2}}|}$$

The CS_{p_i} scores are averaged across the spatial dimension to obtain a quality score that quantifies the divergence between the two sets of events in latent feature space. It is worth noting that any bounded distance metric can be used for this purpose, but a comprehensive analysis of which distance measure best captures the simulation gap is beyond the scope of this paper.

5. Experiments

For a set of synchronized RGB images and event streams, the DSEC [12] dataset was selected. DSEC is a stereo dataset consisting of a pair of Prophesee Gen3.1 event cameras with a resolution of 640×480 px synchronized with RGB cameras with 1440×1080 px resolution and a frame rate of 20 Hz. The RGB frames are converted to grayscale and taken as input to each simulator for generating event streams. For the events generated by the following simulators: ESIM [38], V2E [16], and PIX2NVS [3], we compute the EQS between the simulated and real event streams.

The DSEC dataset provides RGB frames along with the corresponding synchronized event data. This is important for evaluating event simulation accuracy as simulated event datasets are generated with ESIM [38], V2E [16] and the Pix2NVS [3] simulators, and the EQS is computed between each set of simulated event streams and real event camera data. Notably, DSEC is also a stereo dataset, but for the experiments covered here, the data from one set of sensors is sufficient. To evaluate the fidelity of these simulated datasets, the quality score for the entire DSEC dataset and the same for the first 300 frames of five representative sequences is shown in order to highlight the inter-sequence variation. The results are summarized in Tab. 2. As shown in Fig. 3, we utilize the activations from the conv layers in the first three RVT blocks that downscale the spatial dimension by factors of $1/4$, $1/8$, and $1/16$, respectively. For each feature map, the spatial dimension is divided into equal chunks of (3×3) patches with zero padding if necessary and the cosine similarity score is averaged across all patches. We report the EQS for each simulator over each sequence, where the mean across all sequences can be considered to be the representative score for each simulator.

5.1. Evaluating Generalization Performance

All sequences from the DSEC dataset are run through each simulator and corresponding train, test, and validation sets are generated. The RVT-small [11] model is trained from scratch on this data and tested on the original DSEC test set. Tab. 3 summarizes the performance of each simulator on the simulated and real test sets from the DSEC dataset. Qualitatively observing the data samples generated by each sim-

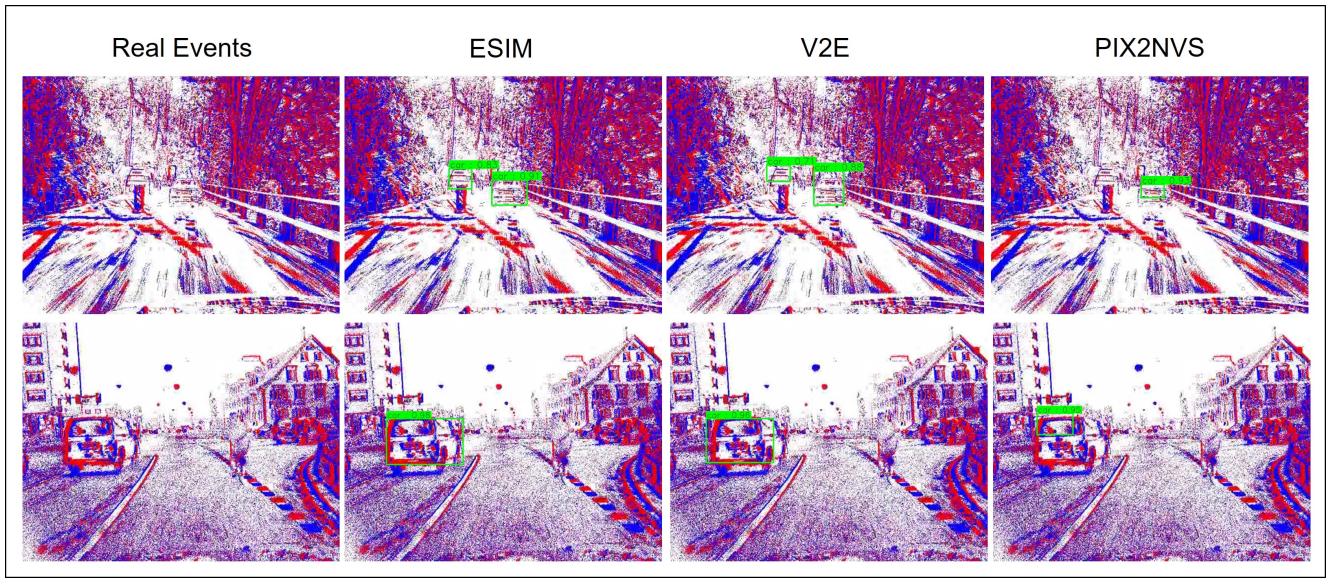

Figure 4. Detection output samples for RVT-small model on real event streams after training on simulated datasets generated using the ESIM, V2E, and PIX2NVS simulators.

DSEC Sequence	EQS		
	V2E	ESIM	Pix2NVS
interlaken_00_c	0.66	0.72	0.44
zurich_city_01_e	0.83	0.88	0.52
zurich_city_04_f	0.74	0.79	0.61
zurich_city_14_b	0.81	0.91	0.74
thun_01_a	0.75	0.84	0.69
Average	0.758	**0.828**	0.599
DSEC Average	0.773	**0.866**	0.501

Table 2. The EQS for each simulator over the first 300 frames of 5 randomly chosen sequences and across the entire DSEC [12] dataset

Simulator	DSEC Test Accuracy (mAP)	
	Simulated	Real
V2E	37.1	14.3
ESIM	**34.4**	**18.6**
PIX2NVS	17.1	05.4

Table 3. Comparing performance of the RVT-small model on the original DSEC test set after training on event data simulated from frames

ulator in Fig. 2 as well as the EQS generated by our quality metric, we can see that the ESIM simulator achieves significantly superior performance on the DSEC test set alongside a higher EQS. This shows that the ESIM simulator is better at mimicking the noise model found in the data gathered by the Prophesee sensor compared to V2E and PIX2NVS, as predicted by the higher EQS in Tab. 2.

6. Conclusion and Future Work

Although a wide range of simulators have been proposed that either generate event-based data from frames or render a virtual $3D$ environment, current simulators are restricted by the simplicity of their event-generation algorithms that largely ignore the nuances of actual event generation. To the best of our knowledge, EQS is the first fully differentiable metric that directly measures how closely a simulated event camera stream mimics real data. Effectively, the EQS quantifies the distance between two raw event camera streams, analogous to the KL-divergence loss commonly used to train generative models. We train a detection model on simulated event data and evaluate it on a real-world driving dataset to demonstrate that EQS is tightly correlated to the model's generalization capabilities to real-world data. Therefore, we anticipate that this can potentially be used as a loss function to optimize machine learning models that take image frames as input and generate corresponding event streams that are close in distribution to real event data. Thus, the EQS can be used to guide the development of the next generation of event-camera simulators, which has the potential to mitigate the difficulties caused by the lack of large labeled event-camera datasets for specific applications in computer vision.

7. Acknowledgment

This research is sponsored by NSF Pathways to Enable Open-Source Ecosystems grant (#2303748) and the Partnerships for Innovation grant (#2329780). We thank data collection support from the Institute of Automated Mobility and computing support from ASU research computing.

References

[1] Manideep Reddy Aliminati, Bharatesh Chakravarthi, Aayush Atul Verma, Arpitsinh Vaghela, Hua Wei, Xuesong Zhou, and Yezhou Yang. Sevd: Synthetic event-based vision dataset for ego and fixed traffic perception, 2024. 2, 3

[2] Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles, 2021. 3

[3] Yin Bi and Yiannis Andreopoulos. Pix2nvs: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1990–1994, 2017. 2, 3, 4, 6

[4] Bharatesh Chakravarthi, M Manoj Kumar, and BN Pavan Kumar. Event-based sensing for improved traffic detection and tracking in intelligent transport systems toward sustainable mobility. In *International Conference on Interdisciplinary Approaches in Civil Engineering for Sustainable Development*, pages 83–95. Springer, 2023. 2

[5] Bharatesh Chakravarthi, Aayush Atul Verma, Kostas Daniilidis, Cornelia Fermuller, and Yezhou Yang. Recent event camera innovations: A survey. *arXiv preprint arXiv:2408.13627*, 2024. 1

[6] Hong-You Chen, Zhengfeng Lai, Haotian Zhang, Xinze Wang, Marcin Eichner, Keen You, Meng Cao, Bowen Zhang, Yinfei Yang, and Zhe Gan. Contrastive localized language-image pre-training, 2025. 4

[7] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. 2, 3, 4

[8] Epic Games. Unreal engine. 3

[9] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. 2

[10] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras, 2020. 4

[11] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras, 2023. 2, 4, 5, 6

[12] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios, 2021. 2, 3, 6, 7

[13] Daxin Gu, Jia Li, Lin Zhu, Yu Zhang, and Jimmy S. Ren. Reliable event generation with invertible conditional normalizing flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):927–943, 2024. 3

[14] Haiqian Han, Jiacheng Lyu, Jianing Li, Henglu Wei, Cheng Li, Yajing Wei, Shu Chen, and Xiangyang Ji. Physical-based event camera simulator. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 2, 3, 4

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 4

[16] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events, 2021. 2, 3, 4, 5, 6

[17] Inc Inivation. Inivation. https://inivation.com/solutions/cameras/. 3

[18] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation, 2024. 4

[19] Xiao Jiang, Fei Zhou, and Jiongzhi Lin. Adv2e: Bridging the gap between analogue circuit and discrete frames in the video-to-events simulator, 2024. 4

[20] Damien Joubert, Alexandre Marcireau, Nic Ralph, Andrew Jolley, André van Schaik, and Gregory Cohen. Event camera simulator improvements via characterized parameters. *Frontiers in Neuroscience*, 15:702765, 2021. 2, 3, 4

[21] J. Kaiser, J. C. V. Tieck, C. Hubschneider, P. Wolf, M. Weber, M. Hoff, A. Friedrich, K. Wojtasik, A. Roennau, R. Kohlhaas, R. Dillmann, and J. M. Zöllner. Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks. In *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 127–134, 2016. 4

[22] Ioannis Kansizoglou, Loukas Bampis, and Antonios Gasteratos. Deep feature space: A geometrical perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6823–6838, 2022. 4

[23] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras, 2022. 2

[24] Xavier Lagorce, G. Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39: 1346–1359, 2017. 3

[25] Hebei Li, Jin Wang, Jiahui Yuan, Yue Li, Wenming Weng, Yansong Peng, Yueyi Zhang, Zhiwei Xiong, and Xiaoyan Sun. Event-assisted low-light video object segmentation, 2024. 2

[26] Zichen Liang, Hu Cao, Chu Yang, Zikai Zhang, and Guang Chen. Global-local feature aggregation for event-based object detection on eventkitti. In *2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–7, 2022. 2

[27] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal con-

trast vision sensor. *IEEE Journal of Solid-State Circuits*, 43 (2):566–576, 2008. 2

[28] Songnan Lin, Ye Ma, Zhenhua Guo, and Bihan Wen. Dvs-voltmeter: Stochastic process-based event simulator for dynamic vision sensors. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022. 2, 4

[29] Bingde Liu, Chang Xu, Wen Yang, Huai Yu, and Lei Yu. Motion robust high-speed light-weighted object detection with event camera. *IEEE Transactions on Instrumentation and Measurement*, 72:1–13, 2023. 2

[30] Haoyue Liu, Shihan Peng, Lin Zhu, Yi Chang, Hanyu Zhou, and Luxin Yan. Seeing motion at nighttime with an event camera, 2024. 2

[31] Inc Lucid vision labs. Lucid vision labs. https://thinklucid.com/triton2-evs/. 3

[32] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36:142–149, 2017. 2, 3

[33] Manasi Muglikar, Leonard Bauersfeld, Diederik Moeys, and Davide Scaramuzza. Event-based shape from polarization. In *IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 2

[34] Manasi Muglikar, Siddharth Somasundaram, Akshat Dave, Edoardo Charbon, Ramesh Raskar, and Davide Scaramuzza. Event cameras meet spads for high-speed, low-bandwidth imaging, 2024. 2

[35] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera, 2020. 3

[36] Bas J. Pijnacker Hordijk, Kirk Y. W. Scheper, and Guido C. H. E. de Croon. Vertical landing for micro air vehicles using event-based optical flow. *Journal of Field Robotics*, 35(1): 69–90, 2017. 2

[37] Inc Prophesee. Prophesee. https://www.prophesee.ai/event-based-sensors/. 3, 4

[38] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Proceedings of Machine Learning Research*, 2018. 3, 4, 5, 6

[39] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras, 2019. 2

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2

[42] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification, 2018. 3

[43] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. *Reducing the Sim-to-Real Gap for Event Cameras*, pages 534–549. 2020. 2

[44] Aayush Atul Verma, Bharatesh Chakravarthi, Arpitsinh Vaghela, Hua Wei, and Yezhou Yang. etram: Event-based traffic monitoring dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22637–22646, 2024. 3

[45] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3 (2):994–1001, 2018. 2

[46] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 2

[47] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2, 4

[48] Zhongyang Zhang, Kaidong Chai, Haowen Yu, Ramzi Majaj, Francesca Walsh, Edward Wang, Upal Mahbub, Hava Siegelmann, Donghyun Kim, and Tauhidur Rahman. Neuromorphic high-frequency 3d dancing pose estimation in dynamic environment. *Neurocomputing*, 547:126388, 2023. 2

[49] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3 (3):2032–2039, 2018. 3

[50] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras. In *2021 IEEE International Conference on Computational Photography, ICCP 2021*, 2021. 3, 4, 5