# Learned Event-based Visual Perception for Improved Space Object Detection

Nikolaus Salvatore
Pacific Defense Solutions
535 Lipoa Parkway, Kihei, Hawaii
nikolaus.salvatore@centauricorp.com

Justin Fletcher
United States Space Force
550 Lipoa Parkway, Kihei, Hawai'i
justin.fletcher.14.ctr@us.af.mil

## Abstract

*The detection of dim artificial Earth satellites using ground-based electro-optical sensors, particularly in the presence of background light, is technologically challenging. This perceptual task is foundational to our understanding of the space environment, and grows in importance as the number, variety, and dynamism of space objects increases. We present a hybrid image- and event-based architecture that leverages dynamic vision sensing technology to detect resident space objects in geosynchronous Earth orbit. Given the asynchronous, one-dimensional image data supplied by a dynamic vision sensor, our architecture applies conventional image feature extractors to integrated, two-dimensional frames in conjunction with point-cloud feature extractors, such as PointNet, in order to increase detection performance for dim objects in scenes with high background activity. In addition, an end-to-end event-based imaging simulator is developed to both produce data for model training as well as approximate the optimal sensor parameters for event-based sensing in the context of electro-optical telescope imagery. Experimental results confirm that the inclusion of point-cloud feature extractors increases recall for dim objects in the high-background regime.*

## 1. Introduction

Use of the near-Earth space environment is essential for many commercial, government, and scientific endeavors. Consequently, the population of resident space objects (RSOs) has grown dramatically over the past decade, necessitating the development of new methods to accurately perceive objects in the space environment at scale. This is challenging even for large, bright objects under ideal observing conditions; for dim objects in sub-optimal collection conditions (i.e., near the moon, through deep atmospheric turbulence) new visual perception approaches are even more crucial. While the apparent visual magnitude (MV), an astronomical measure of the brightness of space objects, of the brightest satellites can reach up to +5 MV, small and low reflectance satellites often have apparent magnitudes as low as +15 MV, well outside the range visible to the naked eye [10][20]. Specialized radar and laser ranging equipment has been used to detect small orbiting targets, but optical solutions provide a versatile, power efficient, and cost effective approach to space domain awareness tasks [23]. However, due to the significant background noise present in charge-coupled device (CCD) sensors as well as the presence of high brightness objects such as stars, sophisticated algorithms are necessary to differentiate between light-sources with little apparent difference in shape or size. Many classical approaches both identify classes of space-based objects and filter background noise data by relying on the temporal information between consecutively captured images [7][29][26]. Nonetheless, the detection of satellites in cislunar orbit and daylight continues to represent a significant hurdle for optical systems, primarily due to ambient lighting having much greater apparent magnitude than potential targets.

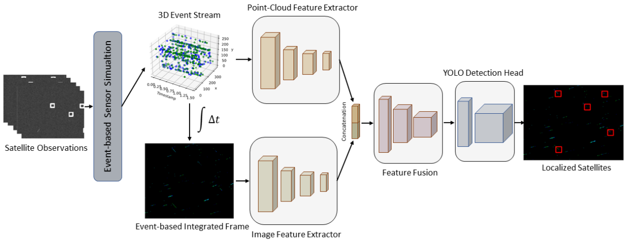

Figure 1. Event-based Satellite Detection Framework

The primary focus of this work is leveraging the unique properties of event-based cameras to enhance the detection of dim RSOs. Also sometimes referred to as dynamic vision sensors, event-based cameras are a class of optical imager in which each pixel responds asynchronously to logarithmic changes in luminance. Rather than synchronized images, event-based cameras generate binary events only at pixels that register a significant change in luminance beyond calibrated threshold values. Given the asynchronous response of the pixels, image data is relayed in a one-dimensional event stream with the form $(x, y, t, p)$, where $x$ and $y$ indicate the two-dimensional coordinates of the pixel in camera space, $t$ represents an assigned timestamp, and $p$ is a positive or negative polarity flag indicating an increase or decrease in luminance, respectively. These sensors, originally developed to model the function of the human eye, are separated into unique categories based on the type of intensity change that is primarily detected, including: spatial contrast, spatial differencing, and temporal contrast [4]. The overall benefit is that the resulting sensors have high dynamic range, high temporal resolution, and sampling rate in the MHz range, while maintaining low power consumption and relatively low data-rate. More mature versions of event-based sensors have begun development over the past few years that have improved resolution and include synchronous frame outputs, albeit with greater noise and far less dynamic range [15]. Some preliminary work has already demonstrated the effectiveness of using event-based sensors for space-based imaging tasks, showing that these sensors can successfully detect Low-Earth Orbit (LEO) and Geosynchronous Earth Orbit (GEO) objects under both nighttime and daytime conditions [3]. The ability of event-based cameras to detect exceedingly small differences in luminance, especially with respect to objects of varying speed, suggests that these sensors could be ideal for detecting dim, but relatively high-speed satellites in conditions too difficult for conventional sensors. Despite the low resolution and significant noise present in many current event-based cameras, these issues are already being addressed by newer versions of hardware that present an even more attractive option for satellite detection.

In this work, we present a framework for both generating synthetic, event-based training data and detecting RSOs in the context of event-based optical telescope imagery. Training data is generated via event-based vision simulation in conjunction with a previously established space scene simulator, SatSim [5]. Both SatSim and the event-based simulator developed in this work allow the varying multiple telescope and sensor parameters, enabling heuristic optimization of parameters for space detection tasks with the intent to extend these findings to physical hardware in future work. Lastly, RSO detection with event-based data is accomplished through a hybridized YOLO architecture utilizing feature fusion between image and point-cloud feature extraction backbones. This hybridized architecture is evaluated using multiple feature extraction backbones and with space scenes of widely varying conditions. While the simulation pipeline established in this work is focused on RSO detection, the hybridized architecture proposed has object detection applications beyond the scope of this work.

## 2. Related Work

The hybrid architecture proposed in this work approaches the satellite localization task using a YOLO-based detection scheme. ”You Only Look Once”, or YOLO, is a now well-known approach to object detection that involves the regression of object bounding boxes from prediction volumes derived from a feature extraction backbone and YOLO detection head [18]. The subsequent updated architecture, YOLOv3, adopts an anchor-based approach that predicts offsets to predefined anchors for better overall accuracy, especially for smaller target objects [19]. For our purposes, YOLO’s simpler, unified detection approach is more easily extensible to multiple sensor modalities and avoids the object proposal modules required by many other architectures. Furthermore, a YOLOv3 architecture has already demonstrated superior detection performance on telescope imagery as compared to more traditional means of satellite detection based on background subtraction and intensity thresholding [5][30]. This previously established model for RSO detection serves as a baseline of comparison for our own hybridized architectures’ performance.

While the image feature extraction backbone used in the original YOLOv3 architecture, DarkNet53, was shown to be effective for satellite detection in [5], a conventional image feature extraction backbone is not immediately suitable for event-based data. While it is possible to create two-dimensional images by integrating over a range of timestamps, flattening the event stream removes potentially critical temporal data. Alternatively, the event stream can be treated as a three-dimensional point cloud, with the timestamp of event defining a third spatial dimension. This representation naturally lends itself to feature extraction using geometric deep learning architectures such as PointNet. While the PointNet architecture extracts only global features from point cloud embeddings, the subsequent PointNet++ architecture introduced hierarchical k-nearest neighbors clustering in conjunction with PointNet to extract local spatial information [16][17]. Other more recent models have also been developed, such as PointCNN, PointConv, among others, which extract local features by approximating three-dimensional convolution operations with varying degrees of permutation invariance [27][9].

The unconventional image data produced by event-based sensors has necessitated modifications to standard computer vision techniques, as well as entirely new methods of object detection and tracking. The earliest methods of detection relied on event-based reformulations of standard techniques such as the Hough transform [12] and Harris corner detector [24], while more recent methods, such as the hierarchy of time-ordered surfaces (HOTS) algorithm, exploit the temporal information of generated events to perform both detection and identification of objects [8]. Several deep learning models have also been developed, which differ in their treatment of event-based data as either a sequence of two-dimensional image data or as a single three-dimensional point-cloud, though both also deal with relatively sparse numbers of events. ”You Only Look at Events”, or YOLE, is a variant of YOLO modified with leaky and asynchronous layers intended to capture temporal changes within the two-dimensional context [1]. Conversely, EventNet is an event-based PointNet, including a temporal encoding with each event and implementing a look-up table approach to perform real-time feature extraction [21]. For all of these methods, however, one of the greatest challenges to event-based deep learning methods is the general lack of available training data. While some event-based versions of well-known data sets do exist [22][13], many of these are produced by shifting conventional two-dimensional images to produce three-dimensional representations. For our purposes, these datasets lack a suitable model of event-based noise and feature objects that have far more distinctive spatial characteristics compared to satellites on starfield backgrounds.

## 3. Approach

To accomplish our satellite detection task, this work is separated into three distinct contributions. Firstly, an event-based simulation framework is established that attempts to emulate the function of the underlying event-based hardware. Secondly, tunable sensor parameters are heuristically optimized with respect to a proposed event-based signal-to-noise ratio metric in order to both increase target detection performance and inform future hardware platforms. Lastly, a YOLO-based object detection model utilizing feature fusion between extracted image and event stream features is established and evaluated on both generated satellite imagery and a small, real dataset.

### 3.1. Dataset Simulation

In order to produce event-based satellite imagery, we leverage the Tensorflow-based SatSim space scene simulator. Developed in [5] for augmenting datasets with GEO satellite targets, SatSim allows for the simulation of electro-optical telescope imagery with a range of varied hardware parameters and scene conditions. The simulator supports generating target satellites of varying apparent magnitudes and velocities in addition to simulated sensor noise, background magnitude due to ambient lighting, and background starfields. Previous approaches to simulating event stream data largely rely on linear extrapolation on the intensity between subsequent images, i.e. producing events by variations of frame differencing. In physical hardware, each pixel in an event-based sensor integrates logarithmic current generated from incident light, producing an event of either positive or negative polarity if this integrated current rises above or falls below a given threshold respectively. To emulate this behavior, simulators will generate a sequence of events based on the difference of log-intensity in each frame, with some simulators generating a single event and others encoding the magnitude of the difference in the number or relative timing of these events.

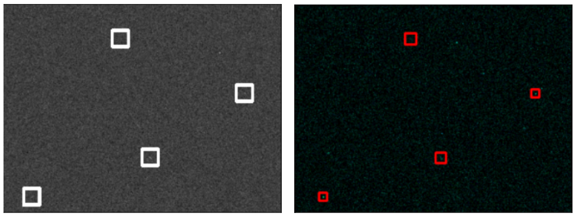

Figure 2. Example SatSim scene with conventional CCD image (left) and event-based integrated frame (right). Marked targets have apparent magnitude 15 MV with background apparent magnitude of 19 MV.

Given the extremely high temporal resolution of events captured in hardware (on the $\mu$s scale), these simulation methods may not fully express the temporal information present in collected data, which could be critical for training deep learning based detection techniques. However, SatSim produces scene images directly using the apparent magnitude of scene components, e.g. satellite targets, background, noise, etc., and the corresponding time interval between simulated frames in order to generate the raw photoelectron counts as they might be observed on a sensor focal plane array.

$$PE_{target} = 10^{\frac{mv_{zeropoint} - mv_{target}}{2.5}} \tag{1a}$$

$$PE_{background} = 10^{\frac{mv_{zeropoint} - mv_{background}}{2.5}} \\ * (iFoV_y * 3600 * iFoV_x * 3600) * \Delta t \tag{1b}$$

$$PE_{noise} = \mathcal{N}(0,\, 1) * \sqrt{PE^2_{readnoise} + PE^2_{elec.noise}} \tag{1c}$$

Equations 1a-1c define the generation of raw photoelectron counts for both target satellites ($PE_{target}$) and background activity ($PE_{background}$) as a function of several parameters pertaining to both the simulated sensor and scene. $mv_{target}$ and $mv_{background}$ refer to the apparent magnitude of satellite targets and sky background respectively, while $mv_{zeropoint}$ indicates the minimum apparent magnitude that produces one photoelectron per second on the sensor focal plane array. $iFoV_y$ and $iFoV_x$ both refer to the instantaneous field-of-view of the sensor, i.e. the angle of visible sky to which the pixel is sensitive, multiplied by 3600 to convert from degrees to arc seconds, while $\Delta t$ represents the exposure time between subsequent frames. Finally, the photoelectrons attributed to noise ($PE_{noise}$) are sampled from a normal distribution ($\mathcal{N}(0,\, 1)$) and then scaled by the root-mean-square of read and electronic noise expected on the focal plane array. Using the fine control over the time interval between frames, we can more closely emulate the physical behavior of the event-based sensor by implementing an integration and discharge behavior. The generation of events is also mitigated by a refractory period, i.e. a minimum time interval before a new event can be generated at a given pixel.

$$\Delta C = logPE(t) - C_t e^{-\lambda \Delta t} \tag{2a}$$

$$p_k = \begin{cases} +1 & \Delta C > \Theta^{+} and \Delta t > t_{refractory} \\ -1 & \Delta C < \Theta^{-} and \Delta t > t_{refractory} \\ 0 & otherwise \end{cases} \tag{2b}$$

$$C_t = \begin{cases} logPE(t) + C_t e^{-\lambda \Delta t} & p_k = 0 \\ 0 & p_k \neq 0 \end{cases} \tag{2c}$$

Equations 2a-2c describe the generation of events from the photoelectron counts produced by SatSim. In Equation 2a, the contrast at a given pixel, in terms of photoelectrons generated, is first transformed to a logarithmic scale ($logPE(t)$), after which the accumulated charge ($C_t$) is subtracted from it to determine the change between time steps. This accumulated charge decays at a rate of $\lambda$, which is a parameter chosen before simulation time. This change in accumulated threshold is subsequently compared against contrast thresholds, $\Theta^{+}$ and $\Theta^{-}$, to determine if a positive or negative polarity event should be generated as shown in Equation 2b. Finally, as indicated in Equation 2c, the accumulated charge is updated if no event was generated or discharged to 0 if otherwise. A small, normally distributed offset is added to the positive and negative threshold for each pixel [6], which represents an inherent error in setting the contrast threshold of pixels. This offset, defined in Equation 3, is applied equally to the positive and negative polarity thresholds for each pixel $k$ such that all pixels in the sensor array will have unique thresholds.

$$\Theta_k^{(+/-)} = \Theta_k^{(+/-)} + \mathcal{N}(0,\, 0.3) \tag{3}$$

Using SatSim in conjunction with this event-based simulation, we can generate event-based space scene data by generating a large sequence of space images with extremely small (15-100$\mu$s) intervals between frames. In order to produce a two-dimensional representation for parallel image processing, we then integrate over the entire time period of an event-based sample, resulting in an image of shape $H \times W \times 2$, with the event polarities separated into the two image channels. Equation 4 from [11] details the general method for integrating events to create conventional frames, which is maintained across many works using event-based image data,

$$log\hat{I}(u;t) = logI(u;0) + \sum_{0<t_k \leq t} p_k C \delta(u - u_k)\delta(t - t_k), \tag{4}$$

where $u = (x, y)$, the $x$ and $y$ image coordinates of the integrated frame, $I$ is the corresponding pixel intensity, $p_k$ represents the event polarity (+/-1), and $C$ is the quantization interval of intensity for each event. As indicated in Equations 2a - 2c, the quantized pixel intensity used in previous works is replaced with the raw photoelectron counts generated by SatSim. Figure 2 depicts a typical SatSim scene and an integrated event-stream image yielded from event-based simulation with ground-truth bounding boxes indicating satellite targets.

### 3.2. Sensor Parameter Optimization

The detectability of a target generated in SatSim scenes is directly determined by the interplay between sensor parameters, scene conditions, and the contrast threshold chosen for event generation. Since the contrast threshold for event-based simulation can also be readily modified in a real-world scenario, we choose to optimize the detectability of target objects with respect to sensor parameters and scene conditions, using contrast threshold as a control output. In this context, we introduce an event-based signal-to-noise ratio metric, similar to measures used in previous noise filtering approaches [14], with which to quantify the detectability of targets in SatSim scenes.

$$EB - SNR = 10log_{10}(\frac{E^{+}_{signal} + E^{-}_{signal}}{E^{+}_{noise} + E^{-}_{noise}})(\Delta t) \tag{5}$$

Equation 5 defines event-based signal-to-noise ratio as it is used in this work, where $E^{+}$ and $E^{-}$ represent the number of positive and negative polarity events, respectively, and $\Delta t$ refers to the window of time in which these events are accumulated. In order to produce training data with maximal detectability, we perform a heuristic optimization procedure to determine the appropriate contrast threshold given three key parameters: sensor field-of-view, exposure time (i.e. temporal window length), and apparent background magnitude of the scene. In general, field-of-view and apparent background magnitude both contribute to the number of noise events generated. Field-of-view, or instantaneous field-of-view once sensor resolution is taken into account, dictates the amount of light incident upon a given pixel from both signal and noise sources found in the scene. Background magnitude defines the intensity of ambient light, and is therefore necessarily a source of noise events. Lastly, exposure time is defined as the total light collection time associated with a generated sample, which in turn dictates the temporal window length, i.e. the time interval over which the stream of events is generated. For optimization, these three values were varied over a continuous range within expected conditions. Event-based signal-to-noise ratios were then calculated across these samples and polynomial regression used to determine optimal contrast thresholds with given parameters. Data for model training and evaluation was finally generated using uniformly, randomly sampled values for each key parameter and the empirically determined optimal contrast thresholds.

### 3.3. Model Architecture

To perform the object detection task, we introduce an ensemble network framework to incorporate features extracted from both an integrated image and raw event stream. For the prediction of bounding boxes and classes, the framework organizes extracted features into a prediction volume in the same manner as YOLOv3, using predefined anchors for localizing targets and non-maximum suppression to filter bounding boxes based on a predicted confidence score. Features extracted by both an image feature extraction backbone, e.g. DarkNet53, and a point-cloud feature extraction backbone, e.g. PointNet, are concatenated before passing through a feature fusion and YOLO detection head. Since point-cloud extraction backbones such as PointNet produce a one-dimensional embedding of the point-cloud, these features are tiled and reshaped for concatenation such that global point-cloud features are associated with each image feature. The resultant features are reduced through a series of convolutional layers to form the final YOLO prediction volume. The overall framework is depicted in Figure 3.

As previously mentioned, an event stream can be integrated to produce two-dimensional images with shape $H \times W \times 2$, where positive and negative polarity event counts are separated into two image channels. For the intended object detection task in this work, targets of interest, i.e. satellites, have very few visual features that distinguish them from both noise and non-target stars. In the conventional image context, a YOLO architecture can learn to distinguish targets via small differences in size and inferred direction as a result of the total exposure time of the collected CCD image. As exposure time is increased, relatively brighter objects will appear larger and produce streaks indicating their trajectory. However, distinguishing targets becomes exceedingly more difficult when the apparent magnitude of a target approaches the ambient background brightness. This issue represents the primary motivation for investigating the benefits of event-based sensors, and consequently the inclusion of model architectures that operate on point-cloud input.

### 3.4. Point Cloud Networks

Due to the high sample rate and temporal resolution of event-based sensors, many such sensors are capable of producing more than a million events per second, resulting in dense point-clouds far larger than those ordinarily processed by PointNet and derivative networks. Furthermore, the event streams in our application lack large continuous shapes such as in the objects that PointNet and similar networks were originally meant to classify and/or segment (see the ModelNet dataset [28]). To overcome these issues, our framework leverages both image and point-cloud feature extraction backbones to localize targets and distinguish them from stars and background activity that appear too similar for image-based object detection alone. For our evaluation, we explored three point-cloud architectures to use in tandem with YOLOv3's DarkNet backbone architecture: PointNet, PointNet++, and PointConv. While PointNet++ and PointConv both extract local features within the point-cloud via inter-layer clustering, PointNet extracts global features that are reduced by application of a global maximum pooling operation on its output. To supplement the basic version of PointNet for use with event streams in our context, we include sinusoidal positional encoding added to the output features as established in [25], in addition to replacing the global maximum pooling layer with a 1-by-1 convolutional layer to introduce a learned reduction of the global output features.

### 3.5. Training

The general process for model training follows that of [19], specific details may be found in supplementary material. A set of purely simulated training data was generated with 400,000 samples of randomized scene/target conditions and with sensor parameters chosen to resemble a Raven-class telescope [2]. Four hybrid configurations were trained and evaluated in total: a frame-based only baseline (Darknet53 only), DarkNet53 with PointNet, DarkNet53 with PointNet++, and DarkNet53 with PointConv. Since our work focuses on the improvement brought by including point-cloud models, we chose to evaluate hybrid models all including the Darknet53 backbone that been previously established for RSO detection. Each model configuration was trained with a batch size of 8 and linearly decaying learning rate of 1e-4 for 500 epochs. In terms of model input, integrated frames were resized to (224,224,2) for DarkNet53 input, down-scaled from event-based sensor dimensions of (346,260) with two channels accommodating separate polarity channels. Event stream input was either padded or truncated to a size of (10000, 4) for point-cloud model input, where the four dimensions represented are $x$ location, $y$ location, timestamp $t$, and event polarity $p$ in that order.

In addition to simulated data sets, we also had access to a small set of annotated, real data collections taken with a Prophesee Gen 3 VGA event-based sensor for validation and simulator-to-real gap analysis. Samples containing relevant satellite targets were isolated into a dataset containing 857 samples for model training and validation; conversely, samples containing stars correlated with available star catalogs were used to extrapolate magnitude values for use in simulation (full details can be found in supplementary material). In order to train the hybrid model

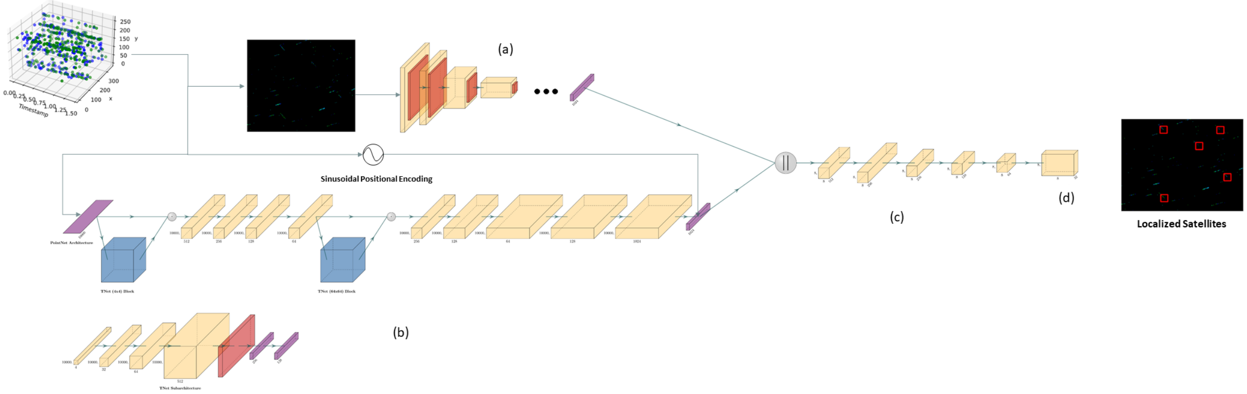

Figure 3. **A Point-Image Fusion Framework.** Our proposed framework comprises four components: (a) An image feature extractor that maps an integrated event-based image to a feature map. (b) A point-cloud feature extractor that produces a position-encoded feature map from point-clouds. (c) A feature fusion trunk that enables feature representation mixing. (d) A detection head, YOLO v3 in this work, that maps the fused feature map to a representation encoding bounding boxes.

with an adequate amount of real data, we applied a randomized, non-destructive data augmentation pipeline with both 2D augmentations (flipping, rotating, and addition of Gaussian noise) and 3D augmentations (point perturbation and point dropout). To explore the simulation to real data gap, we generated an approximately equivalent simulated dataset (20,000 samples) using simulator parameters (including temporal window length, target magnitude, target velocity, and background magnitude) extrapolated from the real samples. Model performance and simulator-to-real gap were finally analyzed by training and cross-evaluation on simulated and real datasets, as well as direct comparison of simulated and real event streams.

# 4. Experiment

We separate the task of training our event-based satellite detection framework into three stages: (1) optimization of contrast threshold, (2) simulation of real dataset equivalents, and (3) model training and evaluation. Since the detectability of targets (i.e., their event-based signal-to-noise ratio) is largely determined by the contrast threshold used during collection, the contrast threshold must be chosen with respect to sensor parameters and observation conditions before collection. Once determined, an optimal contrast threshold value, for a given sensor configuration, may be used to generate the elements of the dataset upon which we may train a model. Since not all sensor and scene parameters for the real data samples are known, approximately equivalent samples are generated using optimal values determined in simulation.

## 4.1. Empirical Contrast Threshold Optimization

To determine the optimal contrast thresholds for the simulation of an event stream, we generate SatSim event-based samples with relevant sensor parameters and scene conditions and using a full range of contrast threshold settings. A sample was generated for each unique combination of values for a total of 45,000 trials. Sample targets were generated with magnitude $BackgroundMag. - 1$ up to a maximum of +15 MV in order to prioritize the detection of dim targets, while velocities were randomly sample from a uniform distribution of U(-10,+10). Full details of the parameters used are left to supplementary material. To determine the optimal contrast thresholds in each scenario, the contrast threshold that maximized event-based signal-to-noise ratio was selected for each unique combination of FoV, exposure time, and background magnitude. A 2nd order multivariate polynomial regression was then fit to the resulting trial data, which was subsequently used to calculate contrast thresholds for dataset generation; the full result is left to supplementary material.

## 4.2. Sim-to-Real Gap

As stated previously, approximately equivalent simulated samples were generated for each annotated real-data sample obtained. Several necessary parameters for simulation were readily available from the real-data annotations, such as FoV, exposure time, target location, and target velocity. However, due to the real data collected being beyond our control, both background and satellite target magnitude, as well as the contrast threshold used by the sensor, were unavailable. In order to form some basis for comparison, these values were therefore determined via extrapolation. The magnitudes of target satellites and background were extrapolated using the known magnitudes of catalogued stars identified within the real-data samples, relating the observed event counts to the known visual magnitudes found in available star catalogs. Using the extrap-

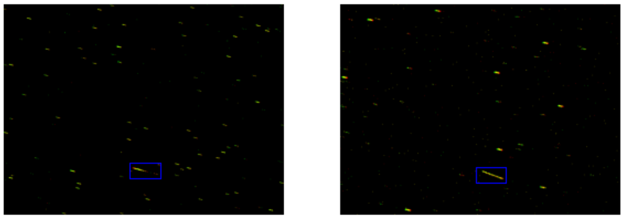

Figure 4. Real satellite collection (left) vs. simulated equivalent sample (right)

olated magnitude values, an optimal contrast threshold was finally calculated using the optimization results of the previous section. Expanded results of this extrapolation are available in supplementary material. Figure 4 shows one example of a real-data sample and the corresponding simulated sample produced by our simulation pipeline, with target satellites marked. More thorough quantitative comparison between the real and simulated event streams can be found in the supplementary material.

### 4.3. Object Detection Results

Purely simulated datasets for model training were generated using the empirically determined regression equation for contrast threshold applied to a full range of parameter values (full details available in supplementary material), while the real-data simulated equivalents were generated as previously specified. Each hybrid model configuration, including the frame-based baseline model, was trained on each of the datasets successively, then evaluated on hold-out sets taken from that dataset. To assess the simulator-to-real gap, models were trained and evaluated on either real or simulated data. Table 4 indicates the results of models with supervised pretraining performed on simulated data, followed by training and evaluation on real data. The precision, recall, maximum F1 score, and auPR results of the final evaluations are detailed in Tables 1, 2, 3, and 4, while the corresponding precision-recall curves can be found in Figure 5.

In order to visualize general trends in recall as they are related to each of the sensor parameters and scene conditions, Figure 6 includes histograms of each relevant parameter in the corresponding data set with overlaid recall values. Here temporal window length refers to the exposure time taken into account for event generation (see Equations 2a-2c). While the primary purpose of the purely simulated datasets is to assess the effects of various simulation parameters, full precision-recall results for these sets can also be found in the supplementary material.

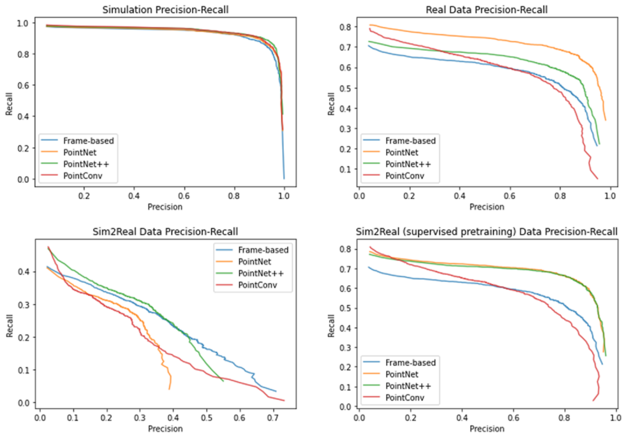

Figure 5. Precision-recall curves for all model configurations on both real and equivalent simulated data.

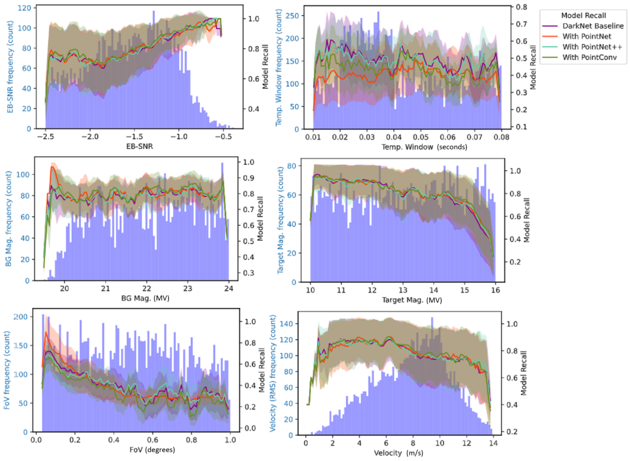

Figure 6. Maximum recall for each hybrid architecture with respect to relevant simulation parameters and conditions

## 5. Discussion and Future Work

Upon evaluation, the maximum recalls in Figure 6 show little difference in the trends of recall versus the sensor parameters and scene conditions with respect to the different model configurations. With regard to the parameters themselves, Figure 6 shows that the strongest indicator of model recall is field-of-view, though this parameter does not impact the real data results given that all real collections were made with the same field-of-view. Nonetheless, EB-SNR, to which FoV heavily contributes, is also a strong indicator of model performance and is readily calculated from event counts in real samples. Recall also exhibits a slightly parabolic trend with respect to velocity, indicating that exceptionally slow moving and fast moving targets both generate fewer events and become more difficult to detect. Despite similar trends with respect to sensor parameters/conditions, the overall precision and recall results indicated in Tables 1 through 4 show a significant increase in both maximum F1 score and area under the PR curves when incorporating the point-cloud architectures versus the frame-based only baseline. The improve-

| Hybrid Architecture | TP | FP | FN | Precision | Recall | $F_1^*$ | auPR | Conf. |
|---|---|---|---|---|---|---|---|---|
| Framed Baseline | 1960 | 210 | 271 | 0.903226 | 0.87853 | 0.890707 | 0.87822 | 0.503 |
| DarkNet-PointNet | 1971 | 188 | 260 | 0.912923 | 0.88346 | 0.89795 | 0.88121 | 0.693 |
| DarkNet-PointNet++ | 1976 | 135 | 255 | **0.936049** | **0.885701** | **0.91018** | **0.88914** | 0.768 |
| DarkNet-PointConv | 1973 | 186 | 258 | 0.913849 | 0.884357 | 0.89890 | 0.88152 | 0.859 |

Table 1. Performance comparison of hybrid frame and event stream architectures on simulated dataset

| Hybrid Architecture | TP | FP | FN | Precision | Recall | $F_1^*$ | auPR | Conf. |
|---|---|---|---|---|---|---|---|---|
| Framed Baseline | 1166 | 338 | 1019 | 0.775266 | 0.533638 | 0.6321 | 0.5297 | 0.507 |
| DarkNet-PointNet | 1441 | 214 | 744 | **0.870695** | **0.659497** | **0.750521** | **0.6746** | 0.511 |
| DarkNet-PointNet++ | 1258 | 275 | 927 | 0.820613 | 0.575744 | 0.676708 | 0.58146 | 0.594 |
| DarkNet-PointConv | 1188 | 454 | 997 | 0.723508 | 0.543707 | 0.620852 | 0.531919 | 0.908 |

Table 2. Performance comparison of hybrid frame and event stream architectures on real data collections.

| Hybrid Architecture | TP | FP | FN | Precision | Recall | $F_1^*$ | auPR | Conf. |
|---|---|---|---|---|---|---|---|---|
| Framed Baseline | 590 | 1083 | 1595 | **0.35266** | 0.27023 | 0.30585 | **0.163835** | 0.48 |
| DarkNet-PointNet | 582 | 1420 | 1603 | 0.29071 | 0.266362 | 0.278003 | 0.10790 | 0.81 |
| DarkNet-PointNet++ | 654 | 1342 | 1531 | 0.327655 | **0.299314** | **0.312844** | 0.15081 | 0.73 |
| DarkNet-PointConv | 564 | 1508 | 1621 | 0.272201 | 0.258124 | 0.264975 | 0.12851 | 0.907 |

Table 3. Sim2Real performance on real data collections with simulator trained models.

| Hybrid Architecture | TP | FP | FN | Precision | Recall | $F_1^*$ | auPR | Conf. |
|---|---|---|---|---|---|---|---|---|
| Framed Baseline | 1166 | 338 | 1019 | 0.775266 | 0.533638 | 0.6321 | 0.5297 | 0.507 |
| DarkNet-PointNet | 1415 | 282 | 770 | **0.833824** | 0.647597 | **0.7290** | **0.636029** | 0.623 |
| DarkNet-PointNet++ | 1428 | 309 | 757 | 0.822107 | **0.653547** | 0.7282 | 0.63179 | 0.678 |
| DarkNet-PointConv | 1215 | 573 | 970 | 0.67953 | 0.556064 | 0.6116 | 0.53167 | 0.906 |

Table 4. Sim2Real performance comparison with the addition of supervised pretraining on real data subsets.

ment in object detection performance (with regard to maximum F1 and auPR) seen when using the point-cloud representation is surprisingly far greater for real data samples as opposed to simulated data. The relative better performance of the baseline model on simulated data is most likely due to the far more well-defined appearance of simulated satellites with regard to generated events. Referring to Figure 4, the simulator pipeline generates a consistent number of events across the entire timeframe, resulting in a visibly larger streak that is more easily detectable by the frame-based only model.

With regard to the simulator-to-real gap, the object detection performance of models trained in simulation and evaluated on real data shows a sizable drop in performance without additional real data training. However, the reasons for this disparity can be explained by comparisons of the streams themselves (full results found in the supplementary material). Firstly, the size of the simulated event streams is far larger than that of the real data, despite the same exposure time, suggesting that the sensor contrast threshold is much larger than that determined by our optimization method. Secondly, the average EB-SNR value observed across the real data is much lower than that of the simulated data, which would significantly impact the overall performance as previously established. Finally, and perhaps most importantly for point-cloud model training, the real data exhibits significantly different ratios of positive to negative events (and greater variation) both in regard to targets and the overall event streams. Regrettably, the lack of contrast threshold information from the real dataset presents a significant issue for assessing how closely the simulation matches real data. However, we believe that the results show that the simulation still presents a useful method for experimentation on relevant event-based data, as well as improving model generalization by enabling the simulation of scenes outside the conditions found in currently available real data. Nonetheless, discrepancies between the simulated and real sensors are a necessary avenue for further research, which is also contingent upon the acquisition of new hardware and data collection opportunities.

While further architecture exploration can no doubt increase detection performance, the more pressing avenue for improving target detection is increasing target visibility. Developing a better means of tuning the event-based contrast threshold, most likely through a neural network-based control mechanism, would not only improve EB-SNR and object detection in simulation, but could also enhance visibility on actual event-based hardware with real-time adjustments. In addition, although several of the point-cloud architectures used in this work already include simple clustering and sampling of the event streams, more advanced clustering algorithms could serve as a basis for filtering noise events or possibly included in a future version of the object detection framework. Finally, while contrast thresholds could be tuned to better match the real data in simulation, a more extensive real data collection with more varied conditions, parameters, and contrast thresholds will be essential to future simulator improvements. Overall, the results found in this work provide evidence that satellite detection can be improved by incorporating temporal event-based data.

## References

[1] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[2] Ryan D Coder and Marcus J Holzinger. Multi-objective design of optical systems for space situational awareness. *Acta Astronautica*, 128:669–684, 2016.

[3] Gregory Cohen, Saeed Afshar, Brittany Morreale, Travis Bessell, Andrew Wabnitz, Mark Rutten, and André van Schaik. Event-based sensing for space situational awareness. *The Journal of the Astronautical Sciences*, 66(2):125–141, 2019.

[4] Tobi Delbrück, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429. IEEE, 2010.

[5] Justin Fletcher, Ian McQuaid, Peter Thomas, Jeremiah Sanders, and Greg Martin. Feature-based satellite detection using convolutional neural networks. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, 2019.

[6] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *arXiv preprint arXiv:1904.08405*, 2019.

[7] Sijie Kong, Jin Zhou, and Wenli Ma. Effect analysis of optical masking algorithm for geo space debris detection. *International Journal of Optics*, 2019, 2019.

[8] Xavier Lagorce, Garrick 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(7):1346–1359, 2016.

[9] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on $\chi$-transformed points. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 828–838, 2018.

[10] Gary A McCue, James G Williams, and Joan M Morford. Optical characteristics of artificial satellites. *Planetary and Space Science*, 19(8):851–868, 1971.

[11] 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(2):142–149, 2017.

[12] Zhenjiang Ni, Cécile Pacoret, Ryad Benosman, Siohoi Ieng, and Stéphane RÉGNIER*. Asynchronous event-based high speed vision for microparticle tracking. *Journal of microscopy*, 245(3):236–244, 2012.

[13] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

[14] Vandana Padala, Arindam Basu, and Garrick Orchard. A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth. *Frontiers in neuroscience*, 12:118, 2018.

[15] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck. Retinomorphic event-based vision sensors: bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102(10):1470–1484, 2014.

[16] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[17] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[19] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[20] Thomas Schildknecht. Optical surveys for space debris. *The Astronomy and Astrophysics Review*, 14(1):41–111, 2007.

[21] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. Eventnet: Asynchronous recursive event processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2019.

[22] Teresa Serrano-Gotarredona and Bernabé Linares-Barranco. Poker-dvs and mnist-dvs. their history, how they were made, and other details. *Frontiers in neuroscience*, 9:481, 2015.

[23] Rong-yu Sun and Sheng-xian Yu. Precise measurement of the light curves for space debris with wide field of view telescope. *Astrophysics and Space Science*, 364(3):1–8, 2019.

[24] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. Fast event-based harris corner detection exploiting the advantages of event-driven cameras. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4144–4149. IEEE, 2016.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[26] Gerard Vives Vallduriola, Diego Andrés Suárez Trujillo, Tim Helfers, Damien Daens, Jens Utzmann, Jean-Noel Pittet, and Nicolas Lièvre. The use of streak observations to detect space debris. *International journal of remote sensing*, 39(7):2066–2077, 2018.

[27] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[28] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d

shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[29] Jiangbo Xi, Desheng Wen, Okan K Ersoy, Hongwei Yi, Dalei Yao, Zongxi Song, and Shaobo Xi. Space debris detection in optical image sequences. *Applied optics*, 55(28):7929–7940, 2016.

[30] HKC Yee. A faint-galaxy photometry and image-analysis system. *Publications of the Astronomical Society of the Pacific*, 103(662):396, 1991.