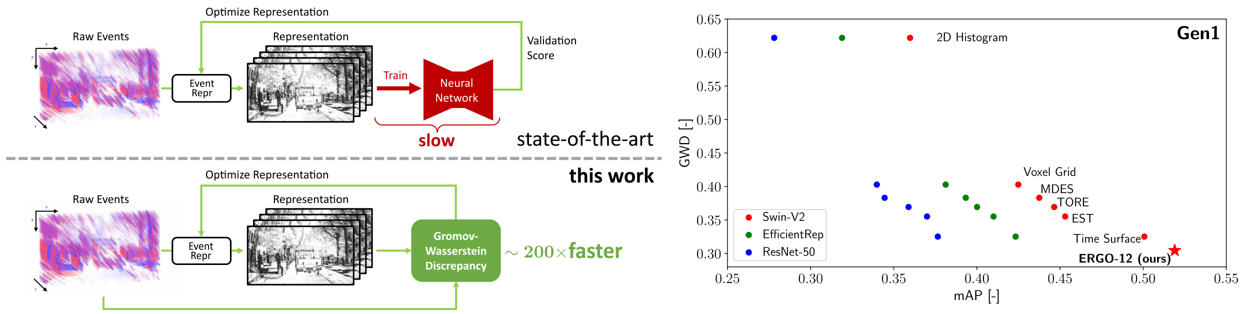
This paper has been accepted for publication at the
IEEE/CVF International Conference on Computer Vision (ICCV), Paris, 2023. ©IEEE

From Chaos Comes Order: Ordering Event Representations for Object Recognition and Detection

Nikola Zubić* Daniel Gehrig* Mathias Gehrig Davide Scaramuzza
Robotics and Perception Group, University of Zurich, Switzerland



(a) Method overview (b) GW Discrepancy vs. Performance

Figure 1: Selecting dense event representations for deep neural networks is exceedingly slow since it involves training a neural network for each representation and selecting the best one based on the validation score. In this work, we eliminate this bottleneck by selecting the representation based on the Gromov-Wasserstein Discrepancy (GWD) on the validation set (a). This metric is 200 times faster to compute and preserves the task performance ranking of event representations across multiple representations, network backbones, datasets and tasks (b, showing object detection performance on the Gen1 [9] dataset). We use it to, for the first time, perform a hyperparameter search on a large family of event representations, revealing new and powerful event representations that exceed the state-of-the-art.

Abstract

Today, state-of-the-art deep neural networks that process events first convert them into dense, grid-like input representations before using an off-the-shelf network. However, selecting the appropriate representation for the task traditionally requires training a neural network for each representation and selecting the best one based on the validation score, which is very time-consuming. This work eliminates this bottleneck by selecting representations based on the Gromov-Wasserstein Discrepancy (GWD) between raw events and their representation. It is about 200 *times faster to compute than training a neural network and preserves the task performance ranking of event representations across multiple representations, network backbones, datasets, and tasks. Thus finding representations with high task scores is equivalent to finding representations with a low GWD. We use this insight to, for the first time, perform a hyperparameter search on a large family of event representations, revealing new and powerful representations that exceed the state-of-the-art. Our optimized representations outperform existing representations by 1.7 mAP on the 1 Mpx dataset and 0.3 mAP on the Gen1 dataset, two established object detection benchmarks, and reach a 3.8% higher classification score on the mini N-ImageNet benchmark. Moreover, we outperform state-of-the-art by 2.1 mAP on Gen1 and state-of-the-art feed-forward methods by 6.0 mAP on the 1 Mpx datasets. This work opens a new unexplored field of explicit representation optimization for event-based learning.*

Multimedia Material

For open-source code please visit https://github.com/uzh-rpg/event_representation_study.

1. Introduction

Event cameras are biologically inspired vision sensors that function in a fundamentally distinct way [12]. Unlike traditional cameras that capture images at a fixed rate, these cameras measure *brightness changes* independently for each pixel, and these changes are referred to as events. The events encode the time, location, and polarity (sign) of the brightness changes. Event cameras offer several advantages over frame-based cameras, including exceptionally high temporal resolution (in the order of μs), a high

*Equal contribution

dynamic range, and low power consumption. Their numerous benefits make them attractive for a wide range of applications like robotics, autonomous vehicles, and virtual reality. However, due to their sparse and asynchronous nature, applying classical computer vision algorithms remains challenging.

Many state-of-the-art deep learning models address this challenge by converting sparse and asynchronous events into dense grid-like representations before processing them with off-the-shelf deep neural networks. By using these networks, methods like this enjoy the advantages of mature learning algorithms and network architectures, and optimized hardware, but we need to make a non-trivial choice of event representation. In fact, the computer vision and robotics fields are witnessing a surge in the number of research papers utilizing event-based vision, resulting in a plethora of new event representations being proposed. Despite this, extensive comparisons of these representations remain rare, making it unclear whether these newer representations should be adopted.

No efficient methodology exists for comparing these representations. Conventionally, comparing event representations involves training a fixed deep-learning model for each event representation separately and subsequently selecting the optimal one based on a validation score. This process is very time-intensive since it requires network training in the loop which often takes hours or days (Fig. 1 a top).

In this work, we propose a fast method to compare event representations which circumvents the need to train a neural network and instead computes the *Gromov-Wasserstein Discrepancy (GWD)* between the raw events and event representation (Fig. 1 a bottom). This metric effectively measures the distortion that is introduced through converting raw events to representations and thus puts an upper bound on the amount of information that can be accessed by downstream neural networks. We show extensive experimental evidence, that this metric preserves the task-performance ranking across a wide range of input representations for several datasets, neural network backbones and tasks (Fig. 1 b). Due to its low computational cost, we apply the GWD to, for the first time, explicitly optimize over a large family of event representations, which reveals a new and powerful representation, which we term 12-channel **E**vent **R**epresentation through **G**romov-Wasserstein **O**ptimization (ERGO-12). For the task of object detection, networks trained with these representations outperform other representations by 1.9 mAP on the 1 Mpx dataset and 0.3 mAP on Gen1, even outperforming state-of-the-art methods by 2.1 mAP on Gen1 and state-of-the-art feed-forward methods by 6.0 mAP on the 1 Mpx dataset. On object recognition, we instead find that our representation outperforms state-of-the-art representations by 3.8%. We believe that the GWD is a powerful tool that opens up a new research field that searches for optimized event representations. Our contributions are summarized as follows:

- We introduce a novel, efficient approach for comparing dense event representations using the *Gromov-Wasserstein Discrepancy (GWD)*.
- We show extensive experimental evidence that it preserves the task performance ranking of neural networks trained with these representations across datasets, neural network backbones and tasks.
- We use it to, for the first time, conduct a hyperparameter search on a vast family of event representations, unveiling novel and powerful event representations that outperform the current state-of-the-art representations on the object detection and object classification task.

2. Related Work

In the field of event-based vision, two primary groups of representations exist: sparse and dense. Methods that use sparse representations [35, 51, 23, 37] preserve the sparsity in the events but do not yet scale to more complex tasks due to a lack of specialized hardware and mature neural networks architectures. This frequently results in lower performance on downstream tasks. In contrast, dense representations [31, 48, 53, 50] offer improved performance since they can leverage mature machine learning algorithms and neural network architectures. Sparse representations, pioneered by asynchronous SNNs [35, 23, 37], are limited by the lack of specialized hardware and computationally efficient backpropagation algorithms. Point cloud encoders [47, 40, 11] have been used due to the spatio-temporal nature of event data, but can be computationally expensive and noisy. Graph neural networks [26, 46, 4, 3, 33, 10] are scalable and have achieved high performance on various vision tasks but are still less accurate than dense methods for event-based vision. In this study, we focus on dense event representations and aim to achieve better task performance by utilizing existing efficient learning algorithms that are appropriate for current hardware.

Early dense representations converted events to histograms [31], generated time surfaces [48] or combined both [52] while relying on standard neural network backbones to process them. However, these representations only capture a low-dimensional representation of events since they typically only use a few channels. Later approaches have tried to capture more event information by either computing higher-order moments [1] or stacking multiple time windows of events [53]. These methods still stack events based on fixed time windows which is problematic when the event rate becomes too large or too small and lead to the introduction of stacking based on the number of events [50]. In parallel, a bio-inspired approach led to the introduction of Time Ordered Recent Event Volumes (TORE) [2], which

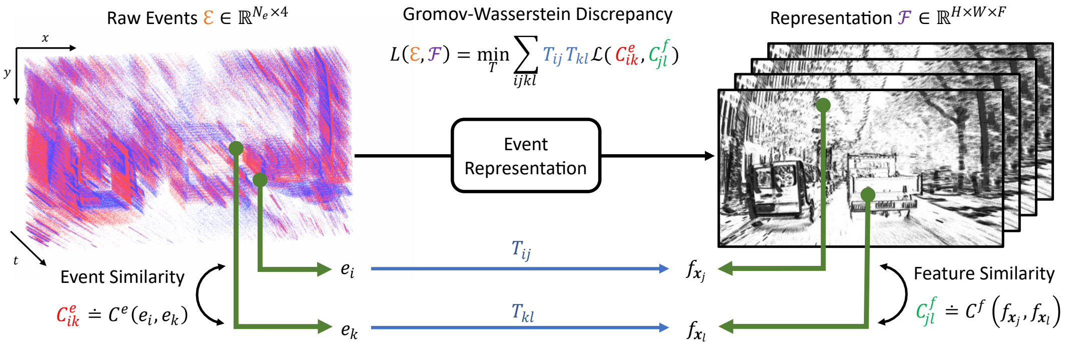

Raw Events $\mathcal{E} \in \mathbb{R}^{N_e \times 4}$
x
y
t
Gromov-Wasserstein Discrepancy
$L(\mathcal{E},\mathcal{F}) = \min_{T} \sum_{ijkl} T_{ij} T_{kl} \mathcal{L}(C^e_{ik}, C^f_{jl})$
Representation $\mathcal{F} \in \mathbb{R}^{H \times W \times F}$
Event Representation
Event Similarity
$C^e_{ik} \doteq C^e(e_i, e_k)$
e_i
e_k
T_{ij}
T_{kl}
f_{x_j}
f_{x_l}
Feature Similarity
$C^f_{jl} \doteq C^f\left(f_{x_j}, f_{x_l}\right)$

Figure 2: Overview of the Gromov-Wasserstein Discrepancy (GWD) between raw events and representations. Events $\mathcal{E}$ are converted to event representations, i.e. a set of features $\mathcal{F}$ at pixel locations $\mathbf{x}$. It is defined as the solution to an optimal transport problem which transports events pairs (e_i, e_k) to feature pairs $(f_{\mathbf{x}_j}, f_{\mathbf{x}_l})$ via transport plan T_{ij}, T_{kl}. If the transport plan preserves the similarities C^e_{ik} and C^f_{jm} between event and feature pairs, this results in a low GWD.

aggregate events into queues. However, they are slow to compute and perform similarly to existing Voxel Grids [53]. Most recently, a powerful representation was proposed by Nam et al. [34], which divides events into multiple overlapping windows that halve the number of events at each stage, which are more robust during varying scene dynamics.

Few papers study the effect of event representations on task performance. While [38] and [21] show small-scale ablation studies to select event representations, only Gehrig et al. [13] performed a large-scale investigation of event representations by training models on various inputs for multiple tasks. Their study demonstrated the advantages of splitting polarities and incorporating timestamps into representations, and it introduced a learnable representation. However, training for a single task was still computationally expensive, which limited the number of representations that could be compared. For this reason, their study did not cover a large number of representations, and in particular, did not consider different window sizes as is done in later work [50, 34], or more advanced aggregations and measurements like in [1]. Our method instead introduces an efficient metric to compare event representations that solves these limitations, allowing us to perform a search over a large family of event representations and go beyond the representations in [50, 1, 34, 14], including even non-differentiable hyperparameters.

3. Method

In this section, we will first introduce the preliminaries on computing event representations (Sec. 3.1) and then propose the metric we use to measure the discrepancy between events and their representation based on the GWD (Sec. 3.2) before concluding with Sec. 3.3 where we use Bayesian optimization to find an optimal event representation.

3.1. Preliminaries

Event cameras measure brightness changes as an asynchronous stream of *events*. Each event is triggered when the intensity $L(\mathbf{u})$ at the pixel $\mathbf{u} = (x, y)$ changes by the contrast threshold C at time t, and thus satisfies

$$p[L(\mathbf{u}, t) - L(\mathbf{u}, t - \Delta t)] = C \tag{1}$$

where $p \in \{-1, 1\}$ is the polarity of the event, and $t - \Delta t$ is the time of the last event. Within a time window ΔT, an event camera thus generates an ordered set of events $\mathcal{E} = \{e_k\}_{k=0}^{N_e - 1}$, with each event $e_k = (\mathbf{u}_k, t_k, p_k) \in \mathbb{R}^4$. To bridge the gap between asynchronous events and dense neural networks, they are usually converted to a dense event representation

$$\mathcal{R} = F(\mathcal{E}), \tag{2}$$

which has features $f_{\mathbf{x}} \doteq \mathcal{R}(\mathbf{x}) \in \mathbb{R}^{N_f}$ indexed by the integer-valued pixel location $\mathbf{x}$. The above representation thus generates a set of features $\mathcal{F} = \{f_{\mathbf{x}}\}_{\mathbf{x} \in \Omega}$, where Ω denotes the image domain and has size $|\Omega| = N_f$ with N_f being the number of pixels in the image. In what follows, we will derive a measure to quantify the distortion between events $\mathcal{E}$ and features $\mathcal{F}$ based on the GWD.

3.2. Gromov-Wasserstein Discrepancy

Converting raw events to a representation invariably distorts the events by removing important distinguishing features from the stream. We would like to measure this distortion since we expect it correlates strongly with a neural network's ability to extract features from these events. In what follows, we will derive a measure of this distortion rate based on the GWD.

We show an overview of the GWD in Fig. 2. We start by measuring the similarity between a set of events and their representation by building a soft correspondence between events e_i and features $f_{\mathbf{x}_j}$, which we denote as T_{ij}[1]. This transport plan effectively moves each event to a corresponding feature, thereby distorting the original set and destroying information. Importantly, such a plan can be interpreted as follows: We transport the events with a total weight of 1, i.e. per-event weight $\frac{1}{N_e}$ to the output features, which also need to receive a total weight of 1 or per-feature weight of $\frac{1}{N_f}$. By this construction, T_{ij} needs to satisfy $\sum_i T_{ij} = 1/N_f$ and $\sum_j T_{ij} = 1/N_e$. This means that a total weight of $1/N_e$ moves from each event, and each feature receives a total weight of $1/N_f$.

In the next step, we measure the distortion introduced by this transportation plan by considering pairwise similarities of input events and features. Let $e_i, e_k \in \mathcal{E}$ be a pair of events and $f_{\mathbf{x}_j}, f_{\mathbf{x}_l} \in \mathcal{F}$ pair of features, with similarity scores $C^e_{ik} \doteq C^e(e_i, e_k)$ and $C^f_{jl} \doteq C^f(f_{\mathbf{x}_j}, f_{\mathbf{x}_l})$ between events and features respectively. Next, consider how the transport plan T acts on these pairs: Generally, a weight T_{ij} is moved from event e_i to feature $f_{\mathbf{x}_j}$. Similarly, the weight T_{kl} is moved from event e_k to feature $f_{\mathbf{x}_l}$. Ideally, such a transport plan should preserve the similarity between pairs of source events and target features, and thus the difference in similarity scores between pairs (i, k) and (j, l) can be used as a measure of distortion. For each event pair and feature pair, we define the distortion as

$$L_{ijkl} = T_{ij} T_{kl} \mathcal{L}(C^e_{ik}, C^f_{jl}), \tag{3}$$

where $\mathcal{L}$ denotes some disparity measurement between C^e_{ik} and C^f_{jl}. Summing over all possible pairs of events and features we thus arrive at the transportation cost:

$$L(T; \mathcal{E}, \mathcal{F}) = \sum_{i,j,k,l} L_{ijkl} = \sum_{i,j,k,l} T_{ij} T_{kl} \mathcal{L}(C^e_{ik}, C^f_{jl}) \tag{4}$$

Minimizing over transport plans, we arrive at the GWD:

$$L(\mathcal{E}, \mathcal{F}) = \min_T \sum_{i,j,k,l} T_{ij} T_{kl} \mathcal{L}(C^e_{ik}, C^f_{jl}), \tag{5}$$

which can be optimized efficiently using [39]. Since the above metric is defined for a single time window of events, we may average it over multiple samples to find:

$$\text{GWD}_N = \frac{1}{N} \sum_i L(\mathcal{E}_i, \mathcal{F}_i). \tag{6}$$

GWD_N can be interpreted as an average distortion rate from raw events to event representations. In Sec. 4.2, we show that GWD_N correlates with a *NN*'s performance with that representation across network backbones, datasets, and event representations. It's also efficient to compute, taking 9 seconds for 50,000 events. In what follows, GWD_N will denote the average over N samples, while GWD denotes the average over the whole validation set.

Similarity scores and distortion function As similarity scores, we choose Gaussian radial basis functions [45] for both events and image features. In detail,

$$C^e_{ik} = e^{\frac{-\|e_i - e_k\|^2}{2h^2\sigma_e^2}}, \quad C^f_{jl} = e^{\frac{-\|f_{\mathbf{x}_j} - f_{\mathbf{x}_l}\|^2}{2h^2\sigma_f^2}} \tag{7}$$

$$\sigma_e^2 = \underset{i<j}{\text{mean}} \|e_i - e_j\|^2, \quad \sigma_f^2 = \underset{i<j}{\text{mean}} \|f_{\mathbf{x}_i} - f_{\mathbf{x}_j}\|^2. \tag{8}$$

with a bandwidth parameter $h = 0.7$. The selection of data-dependent variances normalizes the distances between pairs of events and features such that the similarity score is robust to the dimensionality of the data and the number of samples in the source and target domain. These details are discussed in [7, 45, 36]. While more complex similarity scores could be used, this simple function already achieved good results. As the distortion function, we chose the KL-Divergence i.e.

$$\mathcal{L}(C^e_{ik}, C^f_{jl}) = C^e_{ik} \log \left(C^e_{ik} / C^f_{jl} \right). \tag{9}$$

As a result, the optimization already rejects terms for which $C^e_{ik} \approx 0$, i.e. pairs of events that are far apart. We found that this property was also beneficial in improving the convergence of the optimization problem in Eq. (5)

Improving Convergence of Eq. (5). We found that three features improved convergence and speed up the optimization: (*i*) Normalization of the event coordinates and timestamps by the sensor size and time window respectively, (*ii*) Concatenation of the normalized pixel position to the image features, and (*iii*) Sparsification of image features. Both (*i*) and (*ii*) make the optimization more numerically stable. In fact, without concatenating position information to image features, randomly pixel-wise shuffled event representations would retain the same GWD, although intuitively, neural networks would have a harder time learning from such representations since they typically process nearby features together. Thus reintroducing the position removes this ambiguity and improves the convergence. Finally, (*iii*) removes image features with $\|f_{\mathbf{x}}\| = 0$ since these correspond to pixels where no events were triggered. This step significantly sped up computation by reducing the size of the pair-wise similarity score matrix C^f, with a small impact on convergence.

3.3. Optimizing over Event Representations

With a fast method to measure the effectiveness of an event representation, we can now search for the optimal representation by minimizing Eq. (6) over a space of possible

[1] In the optimal transport literature, this correspondence is also called a *transport plan*.

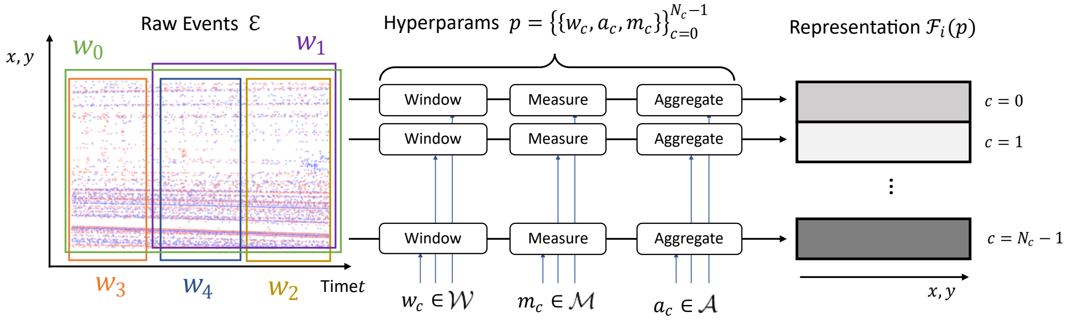


Figure 3: Overview of the hyperparameters we use to construct an event representation (right). For each channel c, we select one of several event time windows $w_c \in \mathcal{W}$ (in color, left), measurement functions $m_c \in \mathcal{M}$ (timestamp, polarity, positive timestamps, etc.), and aggregation functions $a_c \in \mathcal{A}$ (max, mean, sum, variance), resulting in $3N_c$ parameters.

representations with a set of hyperparameters p.

$$p^* = \arg\min_p \frac{1}{N} \sum_i L(\mathcal{E}_i, \mathcal{F}_i(p)). \tag{10}$$

To simplify this optimization, we first describe a very general parametrization, which defines a large family of event representations, extending the family described in [13] in a few ways. These hyperparameters are a small set of categorical variables which can subsequently be optimized using Bayesian optimization.

Parametrization of Event Representations We illustrate the hyperparameters in Fig. 3. In general, we assume an event representation comprises a stack of features, indexed by c (right side), each of which is derived from (*i*) a specific time window w_c of events, (*ii*) a specific measurement m_c of events such as polarity or timestamps, and (*iii*) a specific aggregation a_c, such as summation or averaging. We write such a representation as N_c concatenated feature maps:

$$\mathcal{R} = [\mathcal{R}_0 | \mathcal{R}_1 | ... | \mathcal{R}_{N_c - 1}] \tag{11}$$

$$\text{with } \mathcal{R}_c = a_c(m_c(w_c(\mathcal{E}))) \tag{12}$$

Here w_c is a *windowing function* which selects events within an interval, m_c is the *measurement function* which selects an event feature, and a_c is the *aggregation function*, which aggregates measurements into a single feature map.

Note, in our formulation each channel can have an independent set of parameters, different to [13], which assumes a shared aggregation and measurement function for all channels. This makes our parametrization substantially more expressive than the one in [13]. The number of (non-learnable) representations is $(|\mathcal{A}||\mathcal{M}||\mathcal{W}|)^{N_c} \approx 3.21 \times 10^{27}$, since each channel can be configured independently. Moreover, while a_c and m_c were already discussed in [13], the windowing function is a more general concept, illustrated in Fig. 3 (left). While these windows can be non-overlapping (w_3, w_4, w_2), as for Voxel Grids [53], they can also be overlapping, (w_0, w_1, w_2) as in Mixed-Density Event Stacks [34], or describe windows of a constant event count or constant time [50] In this work we allow each feature channel to select from a basis of windows $\mathcal{W} = \{[t_{0,k}, t_{1,k}]\}_{k=0}^{N_k-1}$, which can combine all types of windows unifying these concepts. In summary, a representation is parametrized by:

$$p = \{(w_c, a_c, m_c)\}_{c=0}^{N_c-1} \tag{13}$$
$$\text{with} \quad m_c \in \mathcal{M}, a_c \in \mathcal{A}, w_c \in \mathcal{W}.$$

For the sets of aggregation functions we select $\mathcal{A} = \{\text{max}, \text{sum}, \text{mean}, \text{variance}\}$ and for the measurement functions $\mathcal{M} = \{t_+, t_-, t, p, c_+, c_-, c\}$ which are most commonly used. Here c, p, and t denote event count (discarding polarity), polarity, and timestamp. The subscripts $+/-$ select only positive or negative events. For the basis of time windows, we select three equally spaced, non-overlapping windows from [53] and four overlapping windows from [34], including the global window. These are illustrated in Fig. 7 (right).

Optimization Procedure The aforementioned parametrization generates redundant combinations that can be obtained by permuting channels or selecting the same feature for different channels. To address this issue and expedite convergence, we propose a stage-wise optimization procedure. Initially, we start with a volume consisting of zeros with N_c channels and optimize over a_0, w_0, and m_0 to fill in the first channel. Next, we optimize the feature for the second channel while keeping the first fixed. With this iterative process, we incrementally fill up the representation, avoiding the selection of redundant representations and resulting in faster optimization. At each stage, we use Gryffin [18], a specialized Bayesian optimizer for categorical variables.

4. Experiments

In Sec. 4.1, we first connect the GWD and event distortion, demonstrating its behavior on several toy examples. Next, in Sec. 4.2, we showcase the correlation between the metric and task performance across multiple tasks, datasets,

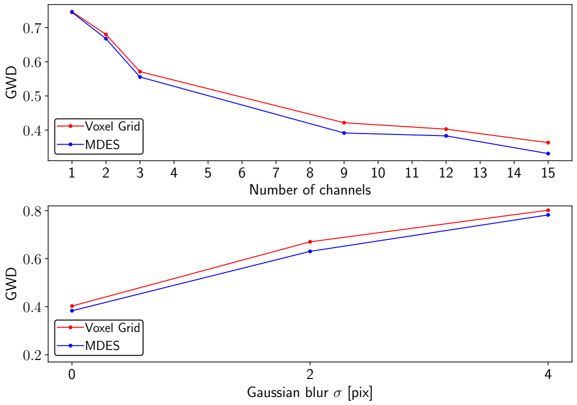


Figure 4: Validation of our metric on the Gen1 validation set. (top) Our metric for Voxel Grids [53] and Mixed-Density Event Stacks [34] with an increasing number of channels. (bottom) Effect of applying Gaussian blur with different blur kernels to the event representation.

backbones, and representations. Lastly, in Sec. 4.3, we show the outcome of Bayesian Optimization on the GWD and offer insights on the acquired event representation before comparing it against the state-of-the-art.

4.1. Toy Example

As explained in Sec. 3.2, the GWD measures the distortion rate from raw events to event representation. Two experiments were designed to test this claim: (1) analyzing the behavior of the metric when varying the number of bins in Voxel Grid [53] and Mixed-Density Event Stack (MDES) [34] representations, and (2) blurring the event representation with increasing standard deviations before measuring the GWD. We perform experiments on the validation set of Gen1[9] and report the results in Fig. 4.

Fig. 4 (top) confirms that the GWD decreases as the number of channels increases in both Voxel Grids and Mixed-Density Event Stacks, which aligns with our intuition that using more channels preserves more information from raw events, resulting in a lower distortion rate. Similarly, the bottom part illustrates that with a growing blur radius, more edges in the event representation are removed, leading to an increase in the GWD. This verifies that the GWD measures the distortion from raw events to event representations.

4.2. Object Detection

Next, we investigate the relationship between GWD and the task performance of a *NN* trained for object detection. We choose two widely used object detection datasets, the Gen1 [9] and 1 Mpx [38] Automotive Detection Dataset, which both deal with event streams featuring labeled bounding boxes for pedestrians, cars, and other vehicles. While the former has a resolution of 304×240, the latter has a resolution of 1280×720. We report the GWD computed over the validation set of each dataset and then train an off-the-shelf object detection framework based on YOLOv6 [24], pre-trained on the Microsoft Common Objects in Context (MS-COCO) [28], on different input representations as in [13]. To accommodate the varying number of channels, we replace the 3-channel input convolution with a $N_c = 12$ channel input convolution, where N_c represents the number of channels in the representation. To show the generality of the result, we also vary the detection backbone between ResNet-50 [17], EfficientRep [24] and Swin Transformer V2 [30], and report results for each.

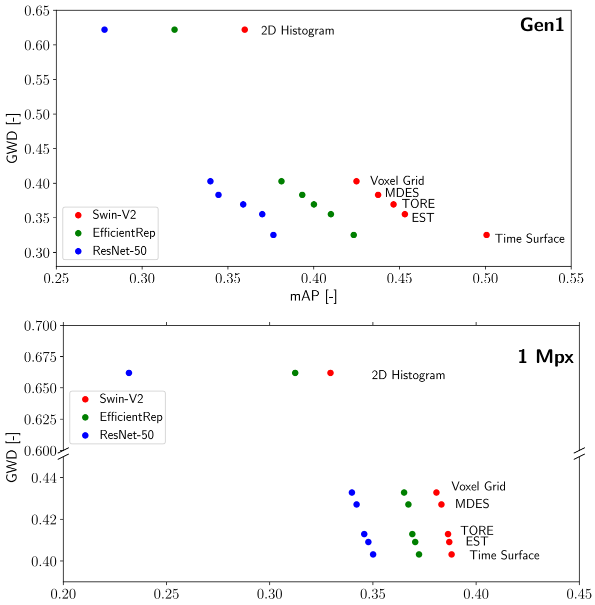


Figure 5: Correlation of the Gromov-Wasserstein Discrepancy with the mAP (higher is better) for object detection on the Gen1 [9] (top) and 1 Mpx [38] (bottom) datasets. Note the spliced y-axis on 1 Mpx, due to the high GWD of the 2D Histogram.

Representations: We test a range of common representations listed below. We compute 12 channels for each representation, except for the 2D Histogram, which has 2.

Voxel Grid: Here, the event time window is split into N_c equal, non-overlapping time windows, and then events in the same time window are aggregated by summing their polarity on a per-pixel basis using bilinear voting [53].

Mixed-Density Event Stack (MDES): For each channel c, this representation selects the most recent $\frac{N_e}{2^c}$ events, where N_e are the events in the time window. For each window, it aggregates the polarity at that pixel [34].

Event Histograms: Events in the time window are split by polarity and then summed into two channels [31].

Time Surface: We convert events into time surfaces from [22] with a decay constant of $\tau = 5$ms. We then sample it at $\frac{N_c}{2} = 6$ equally spaced timestamps, once for

positive and negative events resulting in $N_c = 12$ channels. *TORE*: The Time-Ordered Recent Event Volumes store event timestamps in per-pixel queues. We use queues with capacity $\frac{N_c}{2} = 6$, one for positive and one for negative events, and concatenate them to $N_c = 12$ channels.
Learned representation: Event Spike Tensor (EST) [13] is a learnable event representation that employs a Quantization Layer featuring a trainable kernel. This approach enables the model to effectively transform raw events, optimizing their performance for a given task.
Training Details We adopt the training procedure from YOLOv6 [24]. For each backbone, representation, and dataset, we train for 100 epochs, using Stochastic Gradient Descent with Nesterov momentum increasing from 0.5 to 0.83 over the first 2 epochs. We use a batch size of 32, and a Cosine learning rate schedule, starting at 0.00323 and ending at 0.000387 after 100 epochs. We adopt the classification and box regression losses in [24].

Results: Figs. 5 summarizes the results of the above experiments. For both datasets, Gen1 and 1 Mpx, and all backbones, there is a clear correlation between the GWD and the task performance, i.e. task performance increases as GWD decreases. This conclusion holds for all three network backbones. In particular, MDES with a Swin-V2 detection backbone consistently achieves the highest mAP with 0.43 on Gen1 and 0.39 on 1 Mpx. It also consistently achieves the lowest GWD with 0.38 on Gen1 and 0.40 on 1 Mpx. Utilizing the learnable EST in YOLOv6 with SwinV2 backbone, we achieved a 45.31 mAP score on the Gen1 validation set and a GWD score of 0.3552, positioning EST between MDES and ERGO-12 in Fig. 1b, confirming the expected ranking. We also see that the Swin V2 backbone outperforms other backbones on both datasets for all tested representations. We conclude that while the representation affects task performance, the neural network also has an influence. However, we see that the overall ranking of the representations is preserved.
Using Fewer Samples While in the previous section, we reported the GWD over the validation set of the Gen1 and the 1 Mpx datasets, averaging this metric over such a large dataset still incurs a high computational cost and would make such a metric infeasible for optimization. Therefore, here we investigate if we can use smaller sample sizes to speed up the computation of GWD. In Fig. 6, we show the GWD for the representations in Sec. 4.2 while varying the number of samples. We see that as the sample number decreases, the mean values of the metric change, but the ranking between representations is still preserved reliably after around 100 samples, and thus we use GWD_{100} to optimize over representations. Below this number, the ranking of representations can fluctuate. However, we found that this happens due to a bad convergence of Eq. (5).

Timing Results: We time the computation of the GWD over 100 samples of the Gen1 validation set, where each sample comprises 50,000 events. We run our experiment on an AMD EPYC 7702 32-core server-grade CPU with 32 GB RAM and achieve a runtime of 15 minutes. Note that computing the GWD does not require a GPU. By contrast, training the models in Fig. 5 requires 2 days on a single Tesla V100 GPU with 32 GB of memory, making GWD computation 192 times faster.

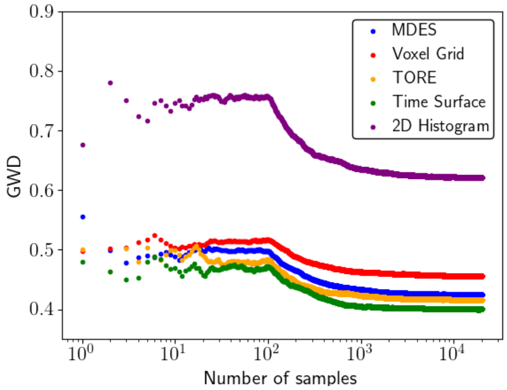


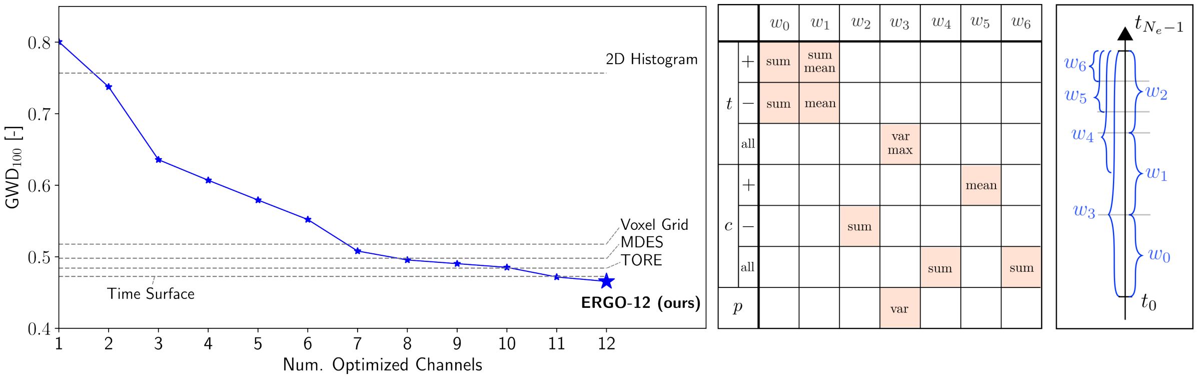
Figure 6: Gromov-Wasserstein Discrepancy for a varying number of samples from the validation set.

4.3. Optimization of Event Representations

Here we report the results of optimizing the event representation according to the procedure in Sec. 3.3. We optimize $N_c = 12$ channels, and at each optimization cycle for channel c, the Bayesian Optimizer selects 100 configurations and then keeps the best-performing configuration.

We show the result of this optimization procedure in Fig. 7. The left shows how the GWD decreases as new channels are added to the representation. We see that after 2 channels, our method outperforms 2D Histograms, after 6 channels we outperform Time Surfaces, after 7 channels we outperform Voxel Grids / TORE, after 9 channels we outperform Mixed-Density Event Stacks and finally, after 12 channels, we achieve a GWD_{100} of 0.47. On the right, we show the different windows (columns) and measurement functions (rows) that are selected. We do not show the order since our representation is unique up to a random channel permutation. However, in the appendix, we show which features are selected at each stage. We see that all windows and all measurement functions are selected at least once, showing how our representation tries to diversify as much as possible. Moreover, timestamp-based measurements often show multiple aggregations, which we argue are necessary to replicate their complex continuous signal.

Comparison with State-of-the-Art Here we compare our method against state-of-the-art recurrent and feed-forward methods on the Gen1 and 1 Mpx test sets. We summarize the results in Tab. 1. Recurrent methods in-



		w_0	w_1	w_2	w_3	w_4	w_5	w_6
t	$+$	sum	sum mean					
	$-$	sum	mean					
	all				var max			
c	$+$						mean	
	$-$			sum				
	all					sum		sum
p					var			

Figure 7: Gromov-Wasserstein Discrepancy for 100 samples (left). At each channel, a Bayesian optimizer selects the next best hyperparameter triple. The chosen hyperparameters are broken down by window and measurement function (right).

Method	Event Repr.	Recurrent	mAP↑	
			Gen1 [9]	1 Mpx [38]
MatrixLSTM+YOLOv3 [5]	MatrixLSTM	✓	0.310	-
E2VID+RetinaNet [41]	Reconstructions	✓	0.270	0.250
RED [38]	Voxel Grid	✓	0.400	0.430
RVT-B [16]	2D Histogram	✓	0.472	0.474
ASTMNet [25]	Asynchronous attention embedding	✓	0.467	**0.483**
Events+RetinaNet [38]	Voxel Grid	✗	0.340	0.180
Events+SSD [19]	2D Histogram	✗	0.301	0.340
Events+RRC [6]	2D Histogram	✗	0.307	0.343
Events+YOLOv3 [20]	2D Histogram	✗	0.312	0.346
AEGNN [46]	Graph	✗	0.163	-
EAGR [15]	Graph	✗	0.321	-
Spiking DenseNet [8]	Spike Train	✗	0.189	-
AsyNet [32]	2D Histogram	✗	0.145	-
SwinV2 + YOLOv6 (ours)	2D Histogram	✗	0.339	0.327
	Time Surface	✗	0.490	0.383
	TORE	✗	0.436	0.381
	Voxel Grid	✗	0.395	0.375
	MDES	✗	0.427	0.378
	ERGO-12	✗	0.493	0.400
	ERGO-12	✗	**0.504***	0.406*

* denotes ours with augmentation

Table 1: Comparison of state-of-the-art event-based object detectors on the test sets of Gen1 [9] and 1 Mpx [38].

clude MatrixLSTM+YOLOv3 [5], which uses a recurrent, learned input representation with a YOLOv3 [42] detector, E2VID+RetinaNet [41] which uses recurrent E2VID reconstructions with a RetinaNet [27] detector, RVT-B which uses a recurrent transformer [16], and RED [38], a recurrent CNN. ASTMNet [25] utilizes asynchronous attention embedding and incorporates recurrent layers in deeper layers, claiming comparable results on both datasets. However, it is not open source. Feedforward methods are grouped into graph-based methods NVS-S [26], AEGNN [46] and EAGR [15], sparse method AsyNet [32], spiking method Spiking DenseNet [8], and finally feedforward dense methods Events+RetinaNet [38], Events+SSD [19] Events+RRC [6], and Events+YOLOv3 [20] which use events as input, and use the RetinaNet [27], SSD [29], RRC [43] or YOLOv3 [42] detector repectively.

We compare these methods to our best-performing YOLOv6 detector from Sec. 4.2 with a SwinV2 transformer backbone trained on various input representations. These include the ones analyzed in Sec. 4.2, *i.e.* the 2D Histogram, Time Surface, TORE, Voxel Grid, and MDES, as well as the optimized representation ERGO-12. Note that these methods do not include data augmentation. We also trained our model with Mixup and Mosaic augmentation from [24], and is indicated by an asterisk * in Tab. 1.

Results On the Gen1 dataset, we see that YOLOv6 with the SWINv2 backbone and ERGO-12 input and data augmentation outperforms all state-of-the-art methods by up to 2.9% mAP by achieving an mAP of 0.504. The runner-up is RVT-B, which uses a recurrent vision transformer. Even without data augmentation, our network with ERGO-12 achieves 0.493, which improves the mAP by 2.1% compared to RVT-B. Compared to the other feed-forward methods based on YOLOv6, ERGO-12 achieves an 0.3 mAP higher mAP, the next best being YOLOv6 with Time Surface [22] with 0.490. Interestingly, as indicated in Fig. 1 (b), the difference in mAP performance between ERGO-12 and Time Surface is 1.8 mAP on the validation set of the Gen1 dataset, which is substantially larger than on the test set.

On 1 Mpx, the best-performing method is ASTMNet with an mAP of 0.483, followed by RVT-B with 0.474. From the feed-forward methods YOLOv6 with ERGO-12 and data augmentation achieves the highest score with 0.406, outperforming runner-up method YOLOv6 with Time Surfaces by 2.3% mAP. Even without augmentation, ERGO-12 achieves a 1.7% higher score than YOLOv6 with Time Surfaces. Compared to state-of-the-art feed-forward methods, ERGO-12 with data augmentation achieves a 6.0% higher score, the runner-up being Events+YOLOv3 with 0.346. On this dataset, recurrent methods are known to perform better since many sequences include stops or slow-motion scenarios. This is challenging for feed-forward methods since they are not able to maintain long-term memory. In general, the improvement of ERGO-12 on 1 Mpx is higher compared to Gen1.

4.4. Object Classification

As an additional task, we investigate the relationship between GWD and the object classification task performance using the *ResNet-34* [17] backbone classifier, pre-trained on ImageNet [44], while changing the input convolution to be compatible with $N_c = 12$ channel

Representation	Description	# of Channels	Accuracy(%)
2D Histogram	Positive and negative event counts	2	46.10
Time Surface	A hierarchy of event-based time-surfaces	12	57.58
TORE	Time-ordered recent event volumes	12	54.64
Voxel Grid	Voxelized grid with bilinear voting aggregation	12	52.40
MDES	Mixed density event stack	12	53.30
ERGO-12	Event representation from *GWD* optimization	12	**61.40**

Table 2: Mini N-ImageNet validation accuracy evaluated on various event representations.

representations. We use the large-scale neuromorphic variant of the ImageNet dataset [44], captured from an event camera that observes monitor-displayed images from ImageNet. Event sequences were recorded using the *480* × *640* resolution Samsung DVS Gen3 event camera [49]. We report the GWD computed over the validation set of the Gen1 dataset. Subsequently, we train the model on Mini N-ImageNet. We opt for GWD on the Gen1 dataset to show the generalization capability of GWD across diverse datasets.

Representations: Excluding the learned representation, we evaluate using identical representations as in object detection. Each representation is computed over 12 channels, except for the 2D Histogram, which uses two channels.

Training Details: Adopting the methodology from the N-ImageNet study [21], all inputs are resized to a *224* × *224* dimension, optimizing GPU memory and inference duration. Training is initialized from scratch with a learning rate set at $3 \cdot 10^{-}4$ and spans 100 epochs. The Adam optimizer with a Nesterov momentum of 0.9 and a weight decay of 0.0001 is used alongside a batch size of 64.

Results: Table 2 summarizes the results of the experiments. There is a clear correlation between the GWD (even on different datasets, in this case, Gen1) and the task performance, i.e. task performance increases as GWD decreases. This conclusion holds for all tested representations on the Mini N-ImageNet dataset. We obtained the following validation set accuracies: 2D Histogram (46.10%), Time Surface (57.58%), TORE (54.64%), Voxel Grid (52.40%), MDES (53.30%), ERGO-12 (61.4%). Their ranking is consistent with the GWD ranking in Fig. 5. Therefore, the GWD can be used also for classification.

5. Conclusion

State-of-the-art event-based deep learning methods typically need to convert raw events into dense input representations before they can be processed by standard networks. However, selecting this representation is very expensive since it requires training a separate neural network for each representation and comparing the validation scores. In this work, we circumvent this bottleneck by measuring the quality of event representations with the Gromov Wasserstein Discrepancy (GWD), which is 200 times faster to compute. We validated extensively on multiple tasks, datasets and neural network backbones that the performance of neural networks trained with a representation correlates with its GWD. We then used this metric to, for the first time, optimize over a large family of representations, revealing a new, powerful representation, ERGO-12. With it, we outperform state-of-the-art representations by 1.9 mAP on the 1 Mpx dataset and 0.3 mAP on the Gen1 dataset, two object detection benchmarks. We also exceed existing representation by 3.8% on the tasks of classification. Moreover, we even outperform the state-of-the-art by 2.1 mAP on Gen1 and state-of-the-art feed-forward methods by 6.0 mAP on the 1 Mpx dataset. This work thus opens a new unexplored field of explicit representation optimization that will push the limits of event-based learning methods.

6. Acknowledgment

This work was supported by the Swiss National Science Foundation through the National Centre of Competence in Research (NCCR) Robotics (grant number 51NF40_185543), and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

7. Appendix

Here we add additional qualitative results and proofs to support the work in the main manuscript. We will refer to sections, equations, figures, and tables in the main manuscript with the prefix "M-", while referring to those in the appendix with "A-". We start by providing additional details about ERGO-12 and GWD in Sec. A-7.1 and include two proofs regarding the robustness of Gromov-Wasserstein Discrepancy (GWD) in Sec. A-7.2. Afterward, we provide more results with fewer optimized channels in Sec. A-7.3. Finally, we show the qualitative results of our method on the Gen1 and 1 Mpx datasets in Sec. A-7.4.

7.1. Additional Details on ERGO-12 and GWD

ERGO-12 details: We provide more details of our optimized representation in Fig. A-9. As can be seen from the top sub-figure, we show the optimized channels in more detail than in Figure M-7. At each new step, there is a decrease in GWD, which demonstrates that additional channels reduce the distance. We calculated GWD on the Gen1 [9] validation dataset, which contained 100 samples, and plotted the results as dashed horizontal lines for chosen representations. The blue line shows the performance of the optimization process after each channel addition. We can observe that, for example, our optimized representation outperforms the Voxel Grid after seven channels and MDES after nine channels. Furthermore, we found that the optimization process initially selected the time function, which capitalizes on the high temporal resolution of event cameras to minimize GWD. Subsequently, counts and polarity were used.

In the bottom sub-figure of Fig. A-9, we visualize the channels of ERGO-12 (our optimized representation after 12 channels). For visualization, we min-max normalized the channels within the range of 0-255. Each channel emphasizes different parts of the image. For instance, the last channel highlights the left edges of the pedestrian, while the seventh channel emphasizes the right part. Our optimization process enables us to capture as much information as possible at different scales and resolutions (spatial and temporal), which is highly advantageous when training with common object detectors. The optimized representation achieves an mAP of over 50% on the Gen1 dataset, and it represents the first non-recurrent neural network architecture that scores over 40% mAP on the 1 Mpx [38] dataset.

Mathematical properties of the GWD: The GWD introduced in [39] and used in this work does not satisfy all axioms of a distance measure and is thus not a metric. It is a generalization of the GW *Distance* that is specifically designed for spaces where an L2 metric comparison is not suitable, as in this work where we compare raw events and representations. [39] showed that using KL-divergence (Eq. 9) with the kernel in Eq. 7 can effectively discard outliers, which we leverage in our work. Due to this more general formalism, the GW Discrepancy does not satisfy symmetry, or the triangle inequality (due to the KL-Divergence in Eq. 9), but ensures non-negativity, and is 0 only for equal sets. Absolute scalability is also not satisfied (see Eq. 7), but is not a common property of distance measures.

7.2. Invariances of the GWD for Events

In this section, we will go over some basic properties of the GWD for events. In particular, we will show that it is invariant to affine feature transformations, concatenation with a constant, and duplication of the features. For clarity, we repeat here the definition of the GWD for events, following Eq. M-5:

$$L(\mathcal{E}, \mathcal{F}) = \min_{T} \sum_{i,j,k,l} T_{ij} T_{kl} \mathcal{L}(C^e_{ik}, C^f_{jl}) \tag{14}$$

with similarity matrices for Eqs. M-7 and M-8.

$$C^e_{ik} = e^{\frac{-\|e_i - e_k\|^2}{2h^2\sigma_e^2}}, \quad C^f_{jl} = e^{\frac{-\|f_{\mathbf{x}_j} - f_{\mathbf{x}_l}\|^2}{2h^2\sigma_f^2}} \tag{15}$$

$$\sigma_e^2 = \underset{i<j}{\text{mean}} \|e_i - e_j\|^2, \quad \sigma_f^2 \quad = \underset{i<j}{\text{mean}} \|f_{\mathbf{x}_i} - f_{\mathbf{x}_j}\|^2. \tag{16}$$

Affine transformation: We expect that if we apply an affine transformation to the event representation, the score should not change since information in the representation remains distinctive. Moreover, we do not want the GWD to be sensitive to the scale of the feature. We see that replacing representation features with

$$f^*_{\mathbf{x}} = a f_{\mathbf{x}} + b \tag{17}$$

changes only the similarity matrix C^f_{jl} to

$$C^{f,*}_{jl} = \exp\left(\frac{-\|f^*_{\mathbf{x}_j} - f^*_{\mathbf{x}_l}\|^2}{2h^2\sigma_f^{2,*}}\right). \tag{18}$$

We see that the norms and data-dependent variances then transform as

$$\|f^*_{\mathbf{x}_j} - f^*_{\mathbf{x}_l}\|^2 = a^2 \|f_{\mathbf{x}_j} - f_{\mathbf{x}_l}\|^2 \tag{19}$$

$$\sigma_f^{2,*} = a^2 \sigma_f^2 \tag{20}$$

We thus see that

$$C_{jl}^{f,*} = \exp\left(\frac{-\|f^*_{\mathbf{x}_j} - f^*_{\mathbf{x}_l}\|^2}{2h^2\sigma_f^{2,*}}\right) \tag{21}$$

$$= \exp\left(\frac{-a^2\|f_{\mathbf{x}_j} - f_{\mathbf{x}_l}\|^2}{2h^2a^2\sigma_f^2}\right) \tag{22}$$

$$= \exp\left(\frac{-\|f_{\mathbf{x}_j} - f_{\mathbf{x}_l}\|^2}{2h^2\sigma_f^2}\right) \tag{23}$$

$$= C_{jl}^f \tag{24}$$

which shows that the similarity matrix does not change. The minimizer of Eq. M-5 thus also does not change, which means the GWD is invariant to this affine transformation. This invariance is only possible through the use of a data-dependent variance, and thus highlights its advantage.

Invariances to Concatenation In the case of concatenation, we consider the following transformation:

$$f^*_{\mathbf{x}} = [f_{\mathbf{x}} \| c_{\mathbf{x}}] \tag{25}$$

where $[.\|.]$ denotes concatenation, and $c_{\mathbf{x}} \in \mathbb{R}^C$ denotes a pixel dependent additional feature. Again, we find that only the similarity matrix C^f_{jl} is affected, and in particular, only the norm and variance, which become:

$$\|f^*_{\mathbf{x}_j} - f^*_{\mathbf{x}_l}\|^2 = \|f_{\mathbf{x}_j} - f_{\mathbf{x}_l}\|^2 + \|c_{\mathbf{x}_j} - c_{\mathbf{x}_l}\|^2 \tag{26}$$

$$\sigma_f^{2,*} = \sigma_f^2 + \underset{i<j}{\text{mean}}\|c_{\mathbf{x}_i} - c_{\mathbf{x}_j}\|^2 \tag{27}$$

We will consider two special cases: $c_{\mathbf{x}} = c$, a constant vector, and $c_{\mathbf{x}} = f_{\mathbf{x}}$ the same feature. In the first case, the additional terms above become 0, meaning that the norm does not change, and thus the metric stays the same. In the second case, the norm transforms as in the affine case, multiplying the squared norm and variance by 2. For the same reasons as before, the metric also stays the same. Generalizing this result to more general $c_{\mathbf{x}}$ remains future work.

7.3. Fewer optimized channels

Figure 8 depicts a correlation between the GWD (given on the x-axis, computed on the Gen1 validation dataset with 100 samples) and the task performance (mAP on object detection task). Since the Swin V2 backbone outperforms all other backbones, it is the only backbone shown in the plot, and the 2D Histogram, which is the poorest-performing method, is omitted. The results demonstrate that our optimized representation with nine and seven channels performs better than MDES and Voxel Grid, respectively, which is consistent with the findings in Figure 9. Furthermore, we observe that the results on 1 Mpx correlate with GWD computed on the Gen1 validation dataset with 100 samples, which highlights the generalization capabilities of our approach.

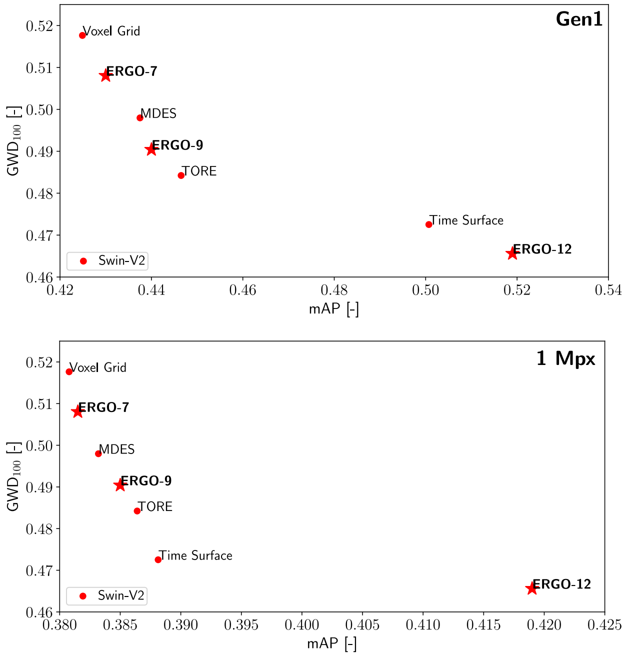


Figure 8: Correlation of the Gromov-Wasserstein Discrepancy with the mAP (higher is better) for object detection on Gen1 [9] (top) and 1 Mpx [38] (bottom) datasets. ERGO-12, ERGO-9, and ERGO-7 represent our optimized representations with twelve, nine, and seven channels. The mAP is reported on the validation set, while the Gromov-Wasserstein Discrepancy is reported on the Gen1 validation dataset with 100 chosen samples.

7.4. Qualitative results

We present qualitative object detection results on the 1 Mpx and Gen1 datasets in Figs. 10 and 11, respectively. Our approach exhibits the ability to detect objects that are not present in the ground truth.

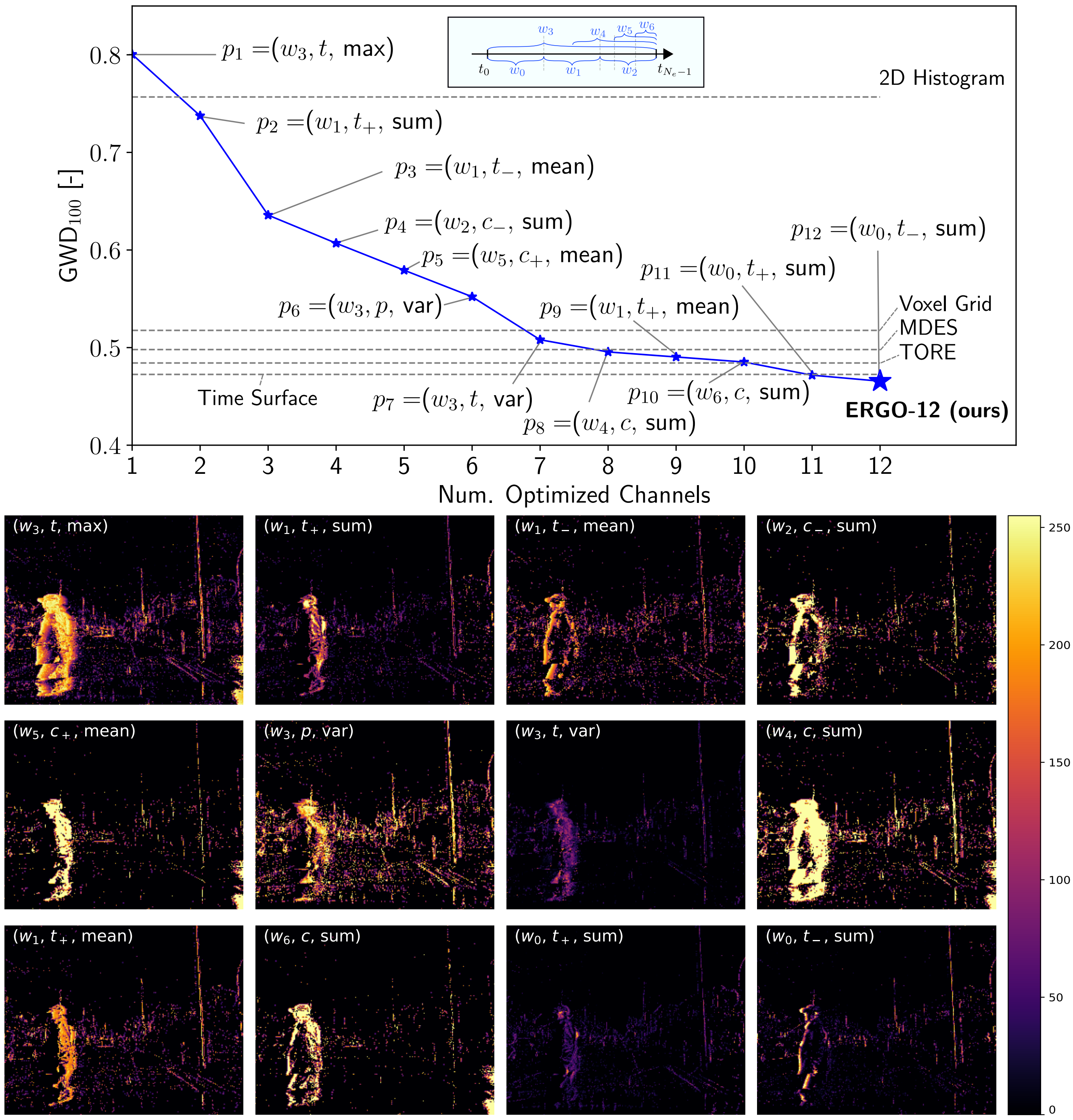


Figure 9: Visualization of the channels of ERGO-12, min-max normalized in the range 0-255. The channels are ordered in row-major order, and the hyperparameters selected are shown in the top left of each subfigure.

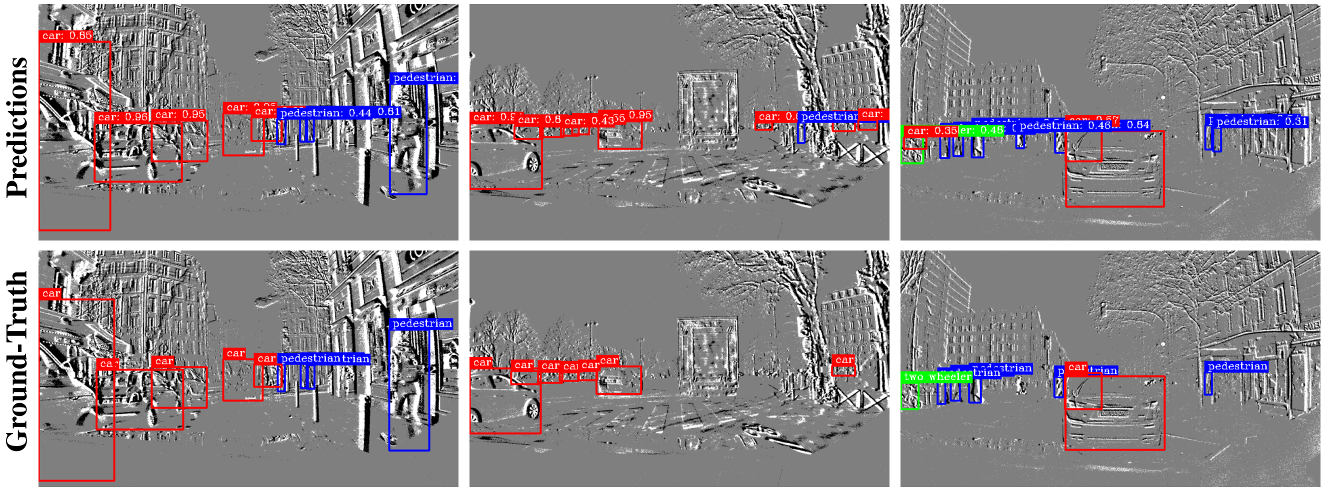


Figure 10: Qualitative results of our method with ERGO-12 input on the 1 Mpx [38] dataset. (top row) predictions, and (bottom row) ground truth. Note that sometimes our method detects objects that do not appear in the ground truth.

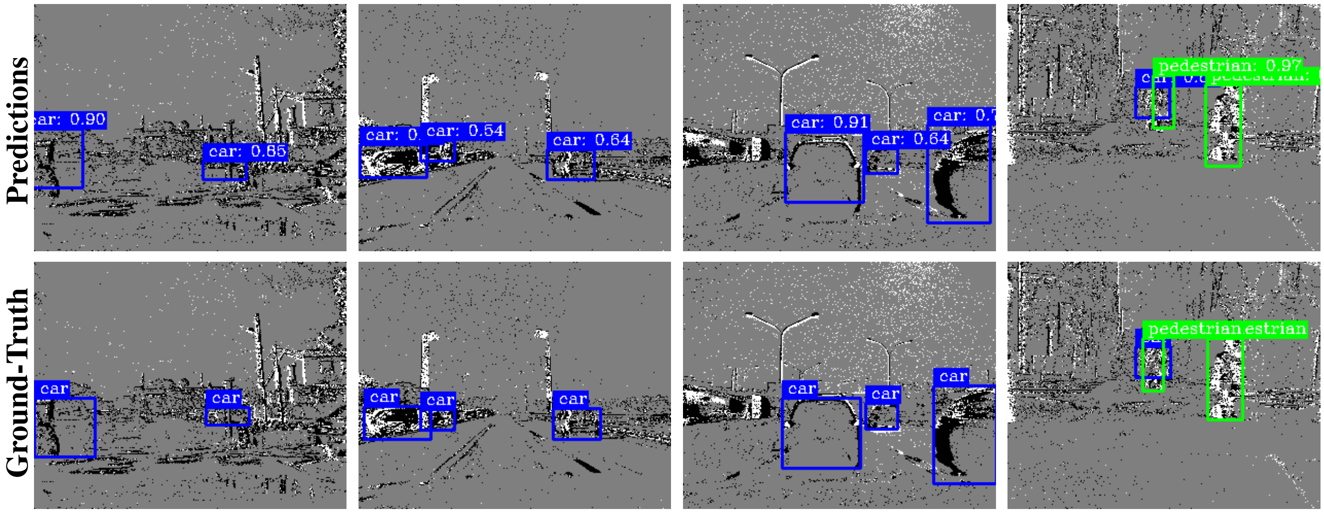


Figure 11: Qualitative results of our method with ERGO-12 input on the Gen1 [9] dataset. (top row) predictions, and (bottom row) ground truth. Note that sometimes our method detects objects that do not appear in the ground truth.

References

[1] Iñigo Alonso and Ana C Murillo. EV-SegNet: Semantic segmentation for event-based cameras. In *CVPRW*, 2019. 2, 3

[2] Robert W. Baldwin, Ruixu Liu, Mohammed Bakheet Almatrafi, Vijayan K. Asari, and Keigo Hirakawa. Time-ordered recent event (tore) volumes for event cameras. *IEEE T-PAMI.*, 45:2519–2532, 2021. 2

[3] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019. 2

[4] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Trans. Image Process.*, 29:9084–9098, 2020. 2

[5] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2020. 8

[6] Nicholas F. Y. Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *CVPRW*, 2018. 8

[7] Emmanuel Chevallier, Didong Li, Yulong Lu, and David Dunson. Exponential-wrapped distributions on symmetric spaces. *SIAM Journal on Mathematics of Data Science*, 4:1347–1368, 12 2022. 4

[8] Loic Cordone, Benoit Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. *Int. Joint Conf. Neural Netw. (IJCNN)*, 2022. 8

[9] Pierre de Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi. A large scale event-based detection dataset for automotive. *arXiv e-prints*, abs/2001.08499, 2020. 1, 6, 8, 10, 11, 13

[10] Yongjian Deng, Hao Chen, Hai Liu, and Youfu Li. A voxel graph cnn for object classification with event cameras. *Conference of Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[11] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *(ICLR)*, 2021. 2

[12] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE T-PAMI.*, 2020. 1

[13] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019. 3, 5, 6, 7

[14] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *European Conference of Computer Vision (ECCV)*, pages 766–781, 2018. 3

[15] Daniel Gehrig and Davide Scaramuzza. Pushing the limits of asynchronous graph-based object detection with event cameras. *arXiv*, 2022. 8

[16] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. *arXiv*, 2022. 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 8

[18] Florian Häse, Matteo Aldeghi, Riley J. Hickman, Loïc M. Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, (8):031406, 2021. 5

[19] Massimiliano Iacono, Stefan Weber, Arren Glover, and Chiara Bartolozzi. Towards event-driven object detection with off-the-shelf deep learning. In *IROS*, 2018. 8

[20] Zhuangyi Jian, Pengfei Xia, Kai Huang, Walter Stechele, Guang Chen, Zhenshan Bing, and Alois Knoll. Mixed frame-/event-driven fast pedestrian detection. *ICRA*, 2019. 8

[21] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2146–2156, October 2021. 3, 9

[22] Xavier Lagorce, Garrick Orchard, Francesco Gallupi, Bertram E. Shi, and Ryad Benosman. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE T-PAMI.*, 39(7):1346–1359, July 2017. 6, 8

[23] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Front. Neurosci.*, 10:508, 2016. 2

[24] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, and Xiaolin Wei. Yolov6: A single-stage object detection framework for industrial applications. *arXiv*, 2022. 6, 7, 8

[25] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Trans. Image Process.*, 31:2975–2987, 2022. 8

[26] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021. 2, 8

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2999–3007, 2017. 8

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference of Computer Vision (ECCV)*, pages 740–755. 2014. 6

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference of Computer Vision (ECCV)*, 2016. 8

[30] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu

Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, 2022. 6

[31] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 5419–5427, 2018. 2, 6

[32] Nico A. Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *European Conference of Computer Vision (ECCV)*, 2020. 8

[33] Anindya Mondal, R Shashant, Jhony H Giraldo, Thierry Bouwmans, and Ananda S Chowdhury. Moving object detection for event-based vision using graph spectral clustering. In *ICCVW*, pages 876–884. IEEE, 2021. 2

[34] Yeongwoo Nam, Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. Stereo depth from events cameras: Concentrate and focus on the future. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 6114–6123, June 2022. 3, 5, 6

[35] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. HFirst: A temporal approach to object recognition. *IEEE T-PAMI.*, 37(10):2028–2040, 2015. 2

[36] Xavier Pennec. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *J. Math. Imaging Vis.*, 25(1):127–154, 2006. 4

[37] José A. Perez-Carrasco, Bo Zhao, Carmen Serrano, Begoña Acha, Teresa Serrano-Gotarredona, Shouchun Chen, and Bernabé Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets. *IEEE T-PAMI.*, 35(11):2706–2719, Nov. 2013. 2

[38] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020. 3, 6, 8, 10, 11, 13

[39] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In *ICML 2016*, June 2016. 4, 10

[40] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, pages 5099–5108, 2017. 2

[41] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE T-PAMI.*, 2019. 8

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[43] Jimmy Ren, Xiaohao Chen, Jianbo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. Accurate single stage detector using recurrent rolling convolution. *Conference of Computer Vision and Pattern Recognition (CVPR)*, 2017. 8

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *Int. J. Com. Vis.*, 115(3):211–252, Apr. 2015. 8, 9

[45] Salem Said, Lionel Bombrun, Yannick Berthoumieu, and Jonathan H. Manton. Riemannian gaussian distributions on the space of symmetric positive definite matrices. *IEEE Trans. Inf. Theory*, 63(4):2153–2170, 2017. 4

[46] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. AEGNN: Asynchronous event-based graph neural networks. 2022. 2, 8

[47] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. EventNet: Asynchronous recursive event processing. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[48] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 1731–1740, 2018. 2

[49] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, Yohan Roh, Hyunku Lee, Yibing Wang, Ilia Ovsiannikov, and Hyunsurk Ryu. A 640x480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation. In *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2017. 9

[50] Lin Wang, S. Mohammad Mostafavi I. , Yo-Sung Ho, and Kuk-Jin Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. *Conference of Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 5

[51] Bo Zhao, Ruoxi Ding, Shoushun Chen, Bernabe Linares-Barranco, and Huajin Tang. Feedforward categorization on AER motion events using cortex-like features in a spiking neural network. *IEEE Trans. Neural Netw. Learn. Syst.*, 26(9):1963–1978, Sept. 2015. 2

[52] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems (RSS)*, 2018. 2

[53] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Conference of Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 5, 6