Scalable Event-by-event Processing of Neuromorphic Sensory Signals With Deep State-Space Models

Mark Schöne
TU Dresden
Dresden, Germany
mark.schoene@tu-dresden.de

Neeraj Mohan Sushma
Ruhr-University Bochum
Bochum, Germany

Jingyue Zhuge
TU Dresden
Dresden, Germany

Christian Mayr
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Centre for Tactile Internet with Human-in-the-Loop (CeTI)
TU Dresden
Dresden, Germany

Anand Subramoney
Royal Holloway, University of London
Egham, United Kingdom

David Kappel
Ruhr-University Bochum
Bochum, Germany

ABSTRACT

Event-based sensors are well suited for real-time processing due to their fast response times and encoding of the sensory data as successive temporal differences. These and other valuable properties, such as a high dynamic range, are suppressed when the data is converted to a frame-based format. However, most current methods either collapse events into frames or cannot scale up when processing the event data directly event-by-event. In this work, we address the key challenges of scaling up event-by-event modeling of the long event streams emitted by such sensors, which is a particularly relevant problem for neuromorphic computing. While prior methods can process up to a few thousand time steps, our model, based on modern recurrent deep state-space models, scales to event streams of millions of events for both training and inference. We leverage their stable parameterization for learning long-range dependencies, parallelizability along the sequence dimension, and their ability to integrate asynchronous events effectively to scale them up to long event streams. We further augment these with novel event-centric techniques enabling our model to match or beat the state-of-the-art performance on several event stream benchmarks. In the Spiking Speech Commands task, we improve state-of-the-art by a large margin of 7.7 % to 88.4 %. On the DVS128-Gestures dataset, we achieve competitive results without using frames or convolutional neural networks. Our work demonstrates, for the first time, that it is possible to use fully event-based processing with purely recurrent networks to achieve state-of-the-art task performance in several event-based benchmarks.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**.

KEYWORDS

Event-stream modeling, state-space models, event-based vision, deep learning, neuromorphic sensors

1 INTRODUCTION

Inspired by the sensory systems in biology, neuromorphic sensors implement an asynchronous and event-based encoding of local environmental changes [Caviglia et al. 2017; Chan et al. 2007; Lichtsteiner et al. 2008; Perot et al. 2020a; Posch et al. 2011]. This sensing paradigm promises several advantages over classical sensors, including energy efficiency, low latency, increased temporal resolution, and dynamic range. For example, systems subject to rapidly changing environments or lighting conditions, such as autonomous robots, benefit from the high dynamic range and low latency of event-based vision sensors. Subsequent processing stages such as machine learning systems, must be compatible with the sensors' asynchronous and temporally sparse event-streams to fully leverage the neuromorphic sensing paradigm.

However, machine learning methods struggle to effectively handle event-streams asynchronously in an event-by-event processing setting. This is due to three key challenges of working with neuromorphic event-streams: (1) Integrating neuromorphic signals event-by-event requires learning interactions between events far apart in time and/or spatial dimensions. This is the well-known problem of learning *long-range dependencies* that has been extensively studied in the recurrent neural networks literature [Bengio et al. 1994; Hochreiter 1991]. (2) Neuromorphic sensors can emit large numbers of events per second, generated in parallel from up to a million asynchronous input channels [Perot et al. 2020a]. Effectively processing very long sequences from neuromorphic sensors requires *parallelization* to use sparsity and asynchrony effectively. The vast advances of highly parallel hardware accelerators favor sequence modeling methods that allow parallelization along the sequence length. (3) *Asynchronous processing*. Neuromorphic sensors produce events in irregular time intervals from many asynchronous sensor channels. Most modern machine learning algorithms require a fixed step size to process sequences effectively. Continuous-time methods have been developed to handle irregular sequences [Schirmer et al. 2022], but struggle with very long sequences due to a limited ability to learn long-range dependencies. Ultimately, machine learning systems that use event-based sensors today often collapse events into frames and thus lose many of the advantages of direct event-based processing.

This work demonstrates the first scalable machine learning method to effectively learn event-based representations directly from high-dimensional asynchronous event-streams. Our method

uses linear state-space models (SSMs), a class of machine learning models that have successfully modeled complex sequential data [Gu, Goel, and Ré 2022]. They are a type of recurrent neural network that can be efficiently parallelized along the sequence dimension (challenge 2). Together with their ability to model long-range dependencies (challenge 1), this property allows significant improvements on tasks like sequential image processing [Gu, Goel, and Ré 2022; J. T. Smith et al. 2023] and raw-audio processing [Goel et al. 2022]. However, asynchronous integration of inputs (challenge 3) has not been addressed by the literature. Furthermore, the machine learning tasks covered by the literature require modeling sequences of a few thousand up to a hundred thousand steps, while neuromorphic sensory signals pose examples of even longer sequences ranging up to millions of events. This work addresses modeling of long asynchronous time-series (challenge 3), while maintaining long-range dependency learning and parallelization as introduced by the SSM literature. We propose several novel techniques on top of SSMs and apply them directly to process neuromorphic sensor signals event by event. We demonstrate that this effectively mitigates all three challenges of (1) long-range dependencies, (2) parallelization, and (3) asynchronous processing. Remarkably, the state-space model extracts spatio-temporal features from event-based vision streams without any convolutional layers.

To our knowledge, this is the first scalable event-by-event processing method for neuromorphic event-streams that achieves compelling results compared to previous frame-based approaches.

2 RELATED WORK

Learning representations from neuromorphic event-streams requires methods that handle very long sequences and sequences with events that are irregularly sampled in time from a set of asynchronous sources. Existing methods represent event-streams as time-frames and learn these representations either end-to-end [Gehrig et al. 2019], or construct them manually [Barchid et al. 2022; Innocenti et al. 2021; Lagorce et al. 2017; Liu et al. 2022]. The time-frame representation allows the processing of the data with convolutional neural networks (CNNs) as well as recurrent neural networks (RNNs) and their spiking variants (SNNs). Given a particular time-frame representation, events can be integrated into the frame representation asynchronously [Cordone et al. 2021; Messikommer et al. 2020]. Zubić et al. [2024] apply state-space models to frames extracted from an event-based vision sensor to speed up training. An exception from the frame-based paradigm is Martin-Turrero et al. [2024], who collect a set of events into a learned tensor representation that can be updated asynchronously to allow asynchronous inference on event-streams. However, we significantly outperform their asynchronous method on DVS128-Gestures and even outperform their synchronous method, while using much fewer parameters. To the best of our knowledge, ours is the first work that operates fully asynchronously on the event-stream and at the same time scales to state-of-the-art performance on standard neuromorphic benchmarks.

Spiking neural networks. Spiking neural networks are often formulated as continuous-time models and discretized for simulation, much like state-space models. The continuous-time formulation theoretically allows asynchronous event-based simulations. In practice, however, most researchers discretize on relatively coarse-grained equidistant time grids. For example, the spiking audio models of Bittar and Garner [2022] and Hammouamri et al. [2024] use simulation steps of $\Delta t = 25\,\text{ms}$ and $\Delta t = 10\,\text{ms}$ respectively. Vision models even use simulation steps of up to $\Delta t = 100\,\text{ms}$ [Liu et al. 2022] or just simulate a total of 4 steps on a single sample [Fang et al. 2023]. In contrast, our method integrates every single event asynchronously with arbitrary resolution without sacrificing simulation scalability.

Long sequences. The problem of modeling long sequences has been addressed by recent developments in linear state-space models (SSMs). SSM-like models were first proposed and shown to have long-range memory in Voelker et al. [2019] and have since been developed to be highly effective scalable models [Gu, Goel, and Ré 2022] via structured linear transformations that efficiently parallelize on modern accelerators. SSMs have successfully been scaled to very long sequences such as raw-audio processing of up to 128 000 steps [Goel et al. 2022], and for autoregressive DNA modeling consisting of over a over a million steps using a time-variant SSM [Gu and Dao 2023]. Time-variant diagonal SSMs have recently demonstrated exceptional performance on large-scale language modeling [De et al. 2024; Gu and Dao 2023]. The parallelization property of linear SSMs has since been applied to parallelize training of neuromorphic systems that operate on time-frames [Fang et al. 2023; Yarga and Wood 2023].

Irregular sequences. Conventional deep learning models are unsuitable for long, irregular time series. They either lack effective time-coding paradigms, such as CNNs or RNNs, or suffer from unfavorable time complexity, such as self-attention-based models. An early attempt to learn long-range dependencies in irregular time series, such as neuromorphic event-streams, was presented by Neil et al. [2016]. They added a new gating mechanism to recurrent architectures that allowed the model to attend to specific frequencies in the data. Ansari et al. [2023] and Schirmer et al. [2022] leveraged continuous-time state-space models driven by stochastic differential equations that integrate discrete observations in a probabilistic manner. J. T. Smith et al. [2023] show that deterministic state-space models can solve the single source pendulum toy task, which was already used by Schirmer et al. [2022] while maintaining the favorable properties of SSMs discussed above. Neuromorphic event-streams are at an entirely different scale than the benchmarks used in deep learning papers. They feature up to a million asynchronous source channels and can sample millions of events per second [Perot et al. 2020b]. We refine the work of J. T. Smith et al. [2023] to handle asynchronous irregular time series of this scale and show promising results on neuromorphic benchmark datasets.

3 SCALABLE EVENT-STREAM MODELING WITH DEEP STATE-SPACE MODELS

We focus on the *simplified state-space layer* (S5) [J. T. Smith et al. 2023] due to its favorable trade-off between simplicity and efficiency for the set of tasks of interest to us. In sec. 3.1, we review the S5 model, and in sec. 3.2, we show how it can be used for efficient, scalable event-stream modeling. In sec. 2 we briefly review other related state-space model architectures.

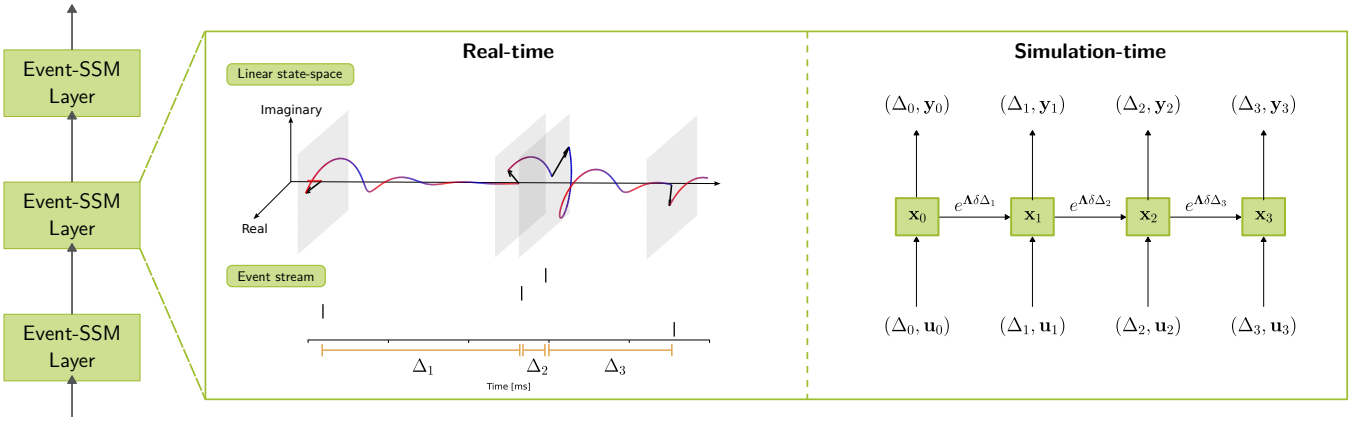

Figure 1: In real-time, our model evolves a linear time-invariant state-space in continuous time and integrates the delta-coded event-stream along the way. The strength of our model stems from its duality with a linear time-variant recurrence relation discretized over the event times. This allows the simulation to leverage the associative scan primitive to parallelize the dynamical system over time.

3.1 Deep State-Space Models

A linear time-invariant state-space model (SSM) in continuous time is given by the linear system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{1}$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\,, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^H$ is the state-space vector and $\dot{\mathbf{x}}$ is its time derivative, $\mathbf{y} \in \mathbb{R}^N$ is the output vector, $\mathbf{u} \in \mathbb{R}^N$ is the input signal, and $\mathbf{A} \in \mathbb{R}^{H\times H}$, $\mathbf{B} \in \mathbb{R}^{H\times N}$, $\mathbf{C} \in \mathbb{R}^{N\times H}$, $\mathbf{D} \in \mathbb{R}^{N\times N}$ are the learnable parameters of the system. Effectively modeling event-streams requires learning dependencies between distant events and irregularly sampled events. Both requirements can be addressed with the above continuous-time model. The extensive theory on linear time-invariant systems allows us to reason efficiently about the system's long-term stability. The continuous-time dynamics can also be discretized on any set of timestamps for irregular sequence modeling. We will elaborate on both features below.

Long-range dependencies. Recurrent sequence models, such as recurrent neural networks, fundamentally suffer from a trade-off between long-term memory and vanishing gradients [Bengio et al. 1994; Hochreiter 1991]. SSMs mitigate this problem by avoiding non-linear recurrence. In contrast to most non-linear dynamics, the linear system in eq. (1) can be carefully tuned for effective long-range dependency modeling.

The set of real matrices that are *diagonalizable* over the complex numbers are dense in the space of real matrices, i.e. a randomly initialized state-space model is diagonalizable almost certainly over the complex numbers. In this case, there exists a diagonal matrix $\mathbf{\Lambda} \in \mathbb{C}^{H\times H}$ and an invertible projection $\mathbf{P} \in \mathbb{C}^{H\times H}$, such that $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$. Hence, there is an equivalent diagonalized SSM with state space variable $\tilde{\mathbf{x}}$ such that $\mathbf{x}(t) = \mathbf{P}^{-1}\tilde{\mathbf{x}}(t)$. We summarize that the class of state-space models defined by

$$\dot{\mathbf{x}}(t) = \mathbf{\Lambda}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{3}$$
$$\mathbf{y}(t) = \mathfrak{Re}\left(\mathbf{C}\mathbf{x}\left(t\right)\right) + \mathbf{D}\mathbf{u}(t) \tag{4}$$

covers the same dynamics as eqs. (1) and (2), where $\mathfrak{Re}(\mathbf{z})$ denotes the real part of a complex variable $\mathbf{z}$, $\mathbf{\Lambda} \in \mathbb{C}^{H\times H}$ is diagonal and $\mathbf{B} \in \mathbb{C}^{H\times N}$, $\mathbf{C} \in \mathbb{C}^{N\times H}$, $\mathbf{D} \in \mathbb{R}^{N\times N}$.

The diagonalized system comes with computational and conceptual advantages. While applying the recurrent operator $\mathbf{A}$ requires $O\left(H^2\right)$ operations per step, the diagonal operator $\mathbf{\Lambda}$ requires only $O\left(H\right)$ operations per step. Furthermore, the H learnable parameters of the state-to-state operator $\mathbf{\Lambda}$ are precisely its spectrum. Since the spectrum defines the system's long-term behavior, we can effectively control its long-term behavior by carefully parameterizing $\mathbf{\Lambda}$.

Long-range dependencies in $\mathbf{x}(t)$ can be modeled effectively if the entries of $\mathbf{\Lambda}$ have negative real part, i.e. they reside on the left half-plane of $\mathbb{C}$. Gu, Goel, Gupta, et al. [2022], Orvieto, S. L. Smith, et al. [2023], and J. T. Smith et al. [2023] enforce the left-half plane condition on the spectrum by parameterizing $\mathbf{\Lambda} = -\exp(\mathbf{\Phi}) + i\mathbf{\Theta}$, with potentially diagonal matrices $\mathbf{\Phi}, \mathbf{\Theta} \in \mathbb{R}^{H\times H}$. While any positive activation function could force the real part of the eigenvalues to be negative, the exponential function is used throughout the literature.

Discretization. Most prior works discretize the continuous system in eqs. (3) and (4) on a regular grid $t_0, \ldots, t_M$ with $t_m - t_{m-1} \equiv \Delta > 0$. J. T. Smith et al. [2023] discretize S5 with the zero-order hold method, which yields a discrete-time system

$$\mathbf{x}_k = \overline{\mathbf{\Lambda}}\mathbf{x}_{k-1} + \overline{\mathbf{B}}\mathbf{u}_k \tag{5}$$
$$\mathbf{y}_k = \mathfrak{Re}\left(\overline{\mathbf{C}}\mathbf{x}_k\right) + \overline{\mathbf{D}}\mathbf{u}_k\,, \tag{6}$$

where $\mathbf{x}_k = \mathbf{x}(t_k), \mathbf{u}_k = \mathbf{u}(t_k)$, Δ is the step size and

$$\overline{\mathbf{\Lambda}} = e^{\mathbf{\Lambda}\Delta}\,, \quad \overline{\mathbf{B}} = \mathbf{\Lambda}^{-1}\left(\overline{\mathbf{\Lambda}} - 1\right)\mathbf{B}\,, \quad \overline{\mathbf{C}} = \mathbf{C}\,, \quad \overline{\mathbf{D}} = \mathbf{D}\,. \tag{7}$$

This discretization method yields two properties that were later found essential for long-range modeling by Orvieto, S. L. Smith, et al. [2023]. Firstly, the state-to-state operator $\overline{\mathbf{\Lambda}}$ is parameterized by an exponential, which enhances stability in conjunction with the half-plane parameterization described above. Secondly, the input

to the state-space $\overline{\mathbf{B}}\mathbf{u}_k$ is modulated by a coefficient that depends on the spectrum of the $\mathbf{\Lambda}$. This coefficient balances the magnitudes of the different components of $\mathbf{x}_k$ taking their respective effective time scales into account, which leads to more stable learning dynamics on very long sequences. We elaborate an extension for asynchronous sensors in sec. 3.2.

Simplified state-space layers. The S5 model [J. T. Smith et al. 2023] is a stack of simplified state-space layers, depicted in fig. 2. The S5 layer consists of the linear state-space model as described in eqs. (5) and (6) and a non-linear multiplicative transformation. According to eq. (7), $\overline{\mathbf{\Lambda}} = e^{(-e^{\mathbf{\Phi}}+i\mathbf{\Theta})\Delta}$ and $\overline{\mathbf{B}} = \mathbf{\Lambda}^{-1}\left(\overline{\mathbf{\Lambda}} - 1\right)\mathbf{B}$. The learnable parameters are $\mathbf{\Phi}, \mathbf{\Theta} \in \mathbb{R}^{H\times H}$, $\mathbf{B} \in \mathbb{C}^{H\times N}$, $\mathbf{C} \in \mathbb{C}^{N\times H}$, $\mathbf{D} \in \mathbb{R}^{N\times N}$, where $\mathbf{\Phi}, \mathbf{\Theta}, D$ are diagonal matrices. A non-linear multiplicative interaction similar to the GLU activation [Dauphin et al. 2017] is applied to the SSM output

$$\mathbf{v}_k = \mathrm{GeLU}\left(\mathbf{y}_k\right) \tag{8}$$

$$\mathbf{z}_k = \mathbf{y}_k \odot \mathrm{sigmoid}\left(\mathbf{W}\mathbf{v}_k\right) . \tag{9}$$

In addition, skip connections and normalization layers are used in S5, which is in line with most modern deep learning models.

Parallelization. Parallelization is a critical component of modern deep learning systems. Non-linear recurrent neural networks such as LSTMs or biologically plausible spiking neural networks lack parallelization along the sequence lengths. Such models have significantly restricted throughput on highly parallel processors such as GPUs. Therefore, training them has been limited to small models or datasets, posing a major drawback in the modern era of scalable deep learning.

Whereas the linear recurrence of eq. (5) allows for efficient parallelization along the sequence in addition to training stability. Therefore, variants of linear state-space models such as S5 learn long-range dependencies in sequences, and scale computationally to very long sequences up to a hundred thousand time steps [Goel et al. 2022]. A comprehensive treatment of the parallelization of linear recurrence equations based on prefix sums of associative operators can be found in Blelloch [1990]. Consider the first-order recurrence relation similar to eq. (5)

$$\mathbf{x}_k = \begin{cases} \overline{\mathbf{B}}\mathbf{u}_0 & i = 0 \\ \overline{\mathbf{\Lambda}} \cdot \mathbf{x}_{k-1} + \overline{\mathbf{B}}\mathbf{u}_k & 0 < i \leq T \end{cases} . \tag{10}$$

Let $\mathbf{c}_k = (\mathbf{a}_k, \mathbf{b}_k)$. As shown in Blelloch [1990], the operation in eq. (10) can be reduced to an associative operator

$$\mathbf{c}_k \otimes \mathbf{c}_l = (\mathbf{a}_k\mathbf{a}_l,\ \mathbf{a}_k\mathbf{b}_l + \mathbf{b}_k) . \tag{11}$$

Hence, the associative scan primitive resolves a recurrence of length M in $O\left(\log M\right)$ time given sufficiently many parallel processors.

3.2 Scalable Event-stream Modeling

In this subsection, we show that these advantageous properties of S5 are also useful in the case of asynchronous event-streams. An event-stream is an ordered set $E = \{(t_m, j_m) \mid m = 0, \ldots, M\}$ of event times $t_0 < \cdots < t_M \in \mathbb{R}$ and corresponding event source channels $j_m \in \{1, \ldots, J\}$. As we will see in the following, our event-based SSM operates efficiently on the differences $\Delta_m = t_m - t_{m-1}$ instead of the timestamps themselves. A linear projection translates the integer representation of event sources j_m to the model's vector representation via $\mathbf{u}_m = \mathbf{E} \cdot \mathrm{onehot}(j_m)$ for $m \in \{0, \ldots, M\}$. This operation can be efficiently implemented as a look-up table that queries the j_m-th column from the projection matrix $\mathbf{E} \in \mathbb{R}^{J\times N}$, a common practice in language modeling.

Discretization on irregular event-streams. Consider a set of J asynchronous channels. In continuous time, an event-stream can then be represented as a sum of dirac deltas

$$\mathbf{u}(t) = \sum_{m=0}^{M} \delta\left(t - t_m\right)\mathbf{u}_m . \tag{12}$$

The general analytical solution of the ODE in eq. (3) with initial conditions $\mathbf{x}\left(t_0\right) = 0$ for the delta coded input as in eq. (12) is

$$\mathbf{x}(t_k) = \int_{t_0}^{t_k} e^{\mathbf{\Lambda}(t_k - s)}\mathbf{B}\mathbf{u}(s)\mathrm{d}s = \sum_{m=0}^{M} e^{\mathbf{\Lambda}(t_k - t_m)}\mathbf{B}\mathbf{u}_m . \tag{13}$$

This solution admits a recursive formulation

$$\begin{aligned} \mathbf{x}_k = \mathbf{x}(t_k) &= \sum_{m=0}^{M} e^{\mathbf{\Lambda}(t_k - t_m)}\mathbf{B}\mathbf{u}_m \\ &= e^{\mathbf{\Lambda}(t_k - t_{k-1})}\left(\sum_{m=0}^{M-1} e^{\mathbf{\Lambda}(t_{k-1} - t_m)}\mathbf{B}\mathbf{u}_m\right) + \mathbf{B}\mathbf{u}_k \\ &= e^{\mathbf{\Lambda}\Delta_k}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k , \end{aligned} \tag{14}$$

where $\Delta_k = t_k - t_{k-1}$. We obtain a formalism to process the irregularly sampled event-stream with a discrete (linear) recurrent neural network, whose state update depends on both the event times via Δ_k and the input values $\mathbf{u}_k$. Notably, this RNN can be simulated fully event-based in the discrete time domain for both inference and learning. Since the RNN state is just a linear combination of the input events, there is no need for advanced event-timing dependent gradient computation methods such as EventProp [Wunderlich and Pehle 2021].

Input normalization of asynchronous events. In sec. 3.1, we discuss the benefits of the input normalization factor $\mathbf{\Lambda}^{-1}\left(\overline{\mathbf{\Lambda}} - 1\right)$ that emerges from the zero-order hold discretization for stable learning. Since the normalization factor depends on the time step Δ through $\overline{\mathbf{\Lambda}}$, the inputs are effectively weighted w.r.t. their timing relative to other inputs. In contrast, we argue that asynchronous events should be independently integrated into the state-space. The normalization factor of an event should not depend on the relative timings of other asynchronous events. Therefore, we disentangle the Δ in (7) as $\Delta = \delta\Delta_k$, where δ is a new learnable parameter and $\Delta_k = t_k - t_{k-1}$ are the actual differences of time steps. Compared to eq. (7), we obtain our asynchronous discretization method with an event time independent normalization factor

$$\mathbf{x}_k = \overline{\mathbf{\Lambda}}_k\mathbf{x}_{k-1} + \overline{\mathbf{B}}\mathbf{u}_k \tag{15}$$

$$\mathbf{y}_k = \mathfrak{Re}\left(\overline{\mathbf{C}}\mathbf{x}_k\right) + \overline{\mathbf{D}}\mathbf{u}_k , \tag{16}$$

with

$$\overline{\mathbf{\Lambda}}_k = e^{\mathbf{\Lambda}\delta\Delta_k} , \quad \overline{\mathbf{B}} = \mathbf{\Lambda}^{-1}\left(e^{\mathbf{\Lambda}\delta} - 1\right)\mathbf{B} , \quad \overline{\mathbf{C}} = \mathbf{C} , \quad \overline{\mathbf{D}} = \mathbf{D} . \tag{17}$$

The inputs are therefore normalized by the units' individual time scales $\tau = \frac{1}{\delta}$, but not by the event-timing dependent Δ_k. We provide evidence supporting this strategy in tab. 5

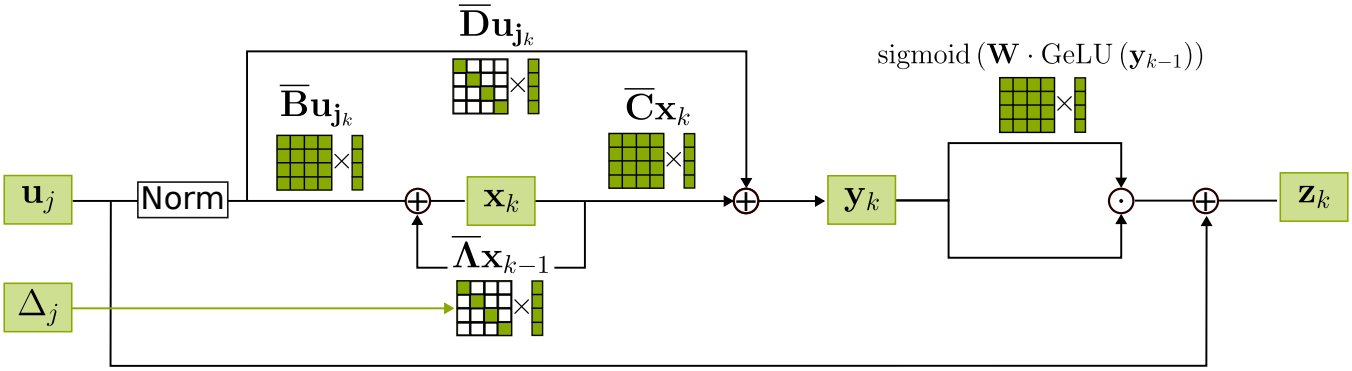

Figure 2: Our modified simplified state-space layer consists of an SSM followed by a non-linear multiplicative transformation. A skip connection and a normalization layer complete the block. The information about event timings is passed to the model via the differences $\Delta_i = t_i - t_{i-1}$.

Note that the trainable parameters are $\delta \in \mathbb{R}_+^{H\times H}, \mathbf{\Phi}, \mathbf{\Theta} \in \mathbb{R}^{H\times H}, \mathbf{B} \in \mathbb{C}^{H\times N}, \mathbf{C} \in \mathbb{C}^{N\times H}, \mathbf{D} \in \mathbb{R}^{N\times N}$, where again $\mathbf{\Phi}, \mathbf{\Theta}, \mathbf{D}$ and δ are diagonal matrices.

Parallelization. With the parameterization in eq. (17), the system described by eq. (5) and eq. (6) becomes a linear time-variant system. The associative scan is still a valid parallelization primitive, since

$$\begin{aligned}\overline{\mathbf{\Lambda}}(\Delta_k) \cdot \overline{\mathbf{\Lambda}}(\Delta_l) &= e^{\mathbf{\Lambda}\delta\Delta_k} e^{\mathbf{\Lambda}\delta\Delta_l} \\ &= e^{\mathbf{\Lambda}\delta(\Delta_k+\Delta_l)} = \overline{\mathbf{\Lambda}}(\Delta_k + \Delta_l)\ . \end{aligned} \tag{18}$$

Therefore, the operator

$$\begin{aligned}\mathbf{c}_k \otimes \mathbf{c}_l &= \left(\overline{\mathbf{\Lambda}}(\Delta_k), \mathbf{b}_k\right) \otimes \left(\overline{\mathbf{\Lambda}}(\Delta_l), \mathbf{b}_l\right) \\ &= \left(\overline{\mathbf{\Lambda}}(\Delta_k + \Delta_l),\ \overline{\mathbf{\Lambda}}(\Delta_k)\mathbf{b}_l + \mathbf{b}_k\right) \end{aligned} \tag{19}$$

acting on $\mathbf{c}_k = \left(\overline{\mathbf{\Lambda}}(\Delta_k), \mathbf{b}_k\right)$ is associative as well, and can be used to parallelize the recurrent system described by eqs. (15) and (16) with parameterization as in eq. (17).

Event-pooling architecture. Event-by-event processing becomes expensive when the model size and the sequence length are scaled up. Although the recurrent operations are cheap in the case of diagonal SSMs, every processed event propagates through the network's numerous dense feed-forward transformations. Furthermore, saving the activations of every event for backpropagation through time causes accelerators to run out of memory when training on long sequences. We mitigate this issue by introducing an event-pooling mechanism that, by subsampling the sequence, can drastically reduce the required compute and memory. Subsampling architectures are widely used in vision models and were proposed for recurrent architectures in Graves and Schmidhuber [2008]. They have also been used in state-space models (e.g. [Goel et al. 2022]). A sequence of lengths M with vectors of dimension H is compressed to a sequence of length M/p. Oftentimes, the vector dimension is increased upon sequence subsampling to Hq. Since linear recurrences effectively compress information [Orvieto, De, et al. 2023], we decided to apply event-pooling after each state-space layer. Hence, M inputs $\mathbf{u}_m$ are integrated into the state-space $\mathbf{x}$, but only a subsampled sequence of length M/p is forwarded to the linear transformation $\overline{\mathbf{C}}\mathbf{x}$.

Similar to frame-based methods, our subsampling architecture reduces the computational overhead by pooling a set of events. In the context of continous-time state-space models, subsampling is equivalent to averaging over the spatio-temporal representation computed by the state-space eq. (15) for p consecutive events. While converting events into frames is a preprocessing step, subsampling can be applied in multiple layers of the model to form hierarchical representations as common practice in audio and vision models.

4 EXPERIMENTS

We evaluate our method, **Event-SSM**, on three event-based datasets that are popularly used in the neuromorphic community. The datasets are provided as raw event-streams, which we process directly without preprocessing into frames. The Spiking Heidelberg Digits (SHD) and Spiking Speech Commands (SSC) datasets were proposed to standardize the evaluation of neuromorphic models [Cramer et al. 2022], both consisting of spike trains that were converted from microphone recordings. DVS128 Gestures (DVS) is a small-scale action recognition dataset [Amir et al. 2017] consisting of a set of 11 gestures recorded with a dynamic vision sensor in 128×128 pixels resolution. While the number of samples in SSC exceeds the other two datasets by an order of magnitude, the number of events per sample in the DVS dataset exceeds the two audio datasets by more than an order of magnitude. An overview of the statistics of the three datasets is presented in tab. 1.

All models presented in this work feature six simplified state-space layers as depicted in fig. 2, with state sizes of either $H = 64$ or $H = 128$. To improve generalization, we implemented an event-based variant of CutMix data augmentation [Yun et al. 2019]. Additional samples were generated by randomly mixing existing ones, i.e. a contiguous stream of events was randomly mixed into another sample of the same batch. Labels were mixed according to the relative number of events from both samples. We implement our model in JAX [Bradbury et al. 2018]. The efficient parallelism of our method allows us to train on the larger SSC (600 M events per epoch) and DVS (390 M events per epoch) datasets in 2 – 10 h on a single

Dataset	Classes	Training samples	Median number of events
Spiking Heidelberg Digits	20	8 200	8 000
Spiking Speech Commands	35	75 500	8 100
DVS128 Gestures	11	1 100	300 000

Table 1: The datasets used to evaluate our event-stream modeling method differ in the number of samples present in the dataset as well as the number of events per sample per dataset. All values are given to two significant digits.

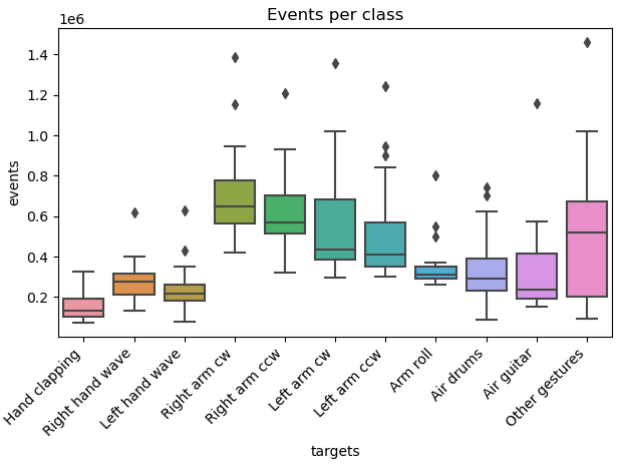

Figure 3: Distribution of the number of events per class in the DVS128-Gesture dataset. The median number of events per sample is about 300,000, and the maximum number of events per sample is about 1.5 million.

A100 GPU. For implementation details and precise hyperparameters, we refer the reader to our published code repository [1].

By reviewing the code of published papers included in our baselines in tab. 2 and tab. 4, we found that it is common practice to pick the best model based on the test set instead of a separate validation set. We did the same for a fair comparison with the baseline methods, even though we believe that this procedure does not align with best practice.

4.1 Spiking audio processing

We present a new state-of-the-art classification result on both spiking audio datasets. These results show that the exact integration of spike timings can, in fact, improve the performance of spiking audio models. A combination of time-jitter, channel-jitter, random noise, drop-event, and cut-mix data augmentations was applied to improve generalization. Tab. 2 shows our results on the SHD dataset. We note that model performance is almost saturated on this dataset. Furthermore, we observed a larger variance with different random seeds compared to the larger SSC dataset.

[1] https://github.com/Efficient-Scalable-Machine-Learning/event-ssm

Spiking Heidelberg Digits	Async. events	Test accuracy	Num params
Bittar and Garner [2022]	✗	93.1 %	0.1 M
Bittar and Garner [2022]	✗	94.6 %	3.9 M
Hammouamri et al. [2024]	✗	95.1 %	0.2 M
Event-SSM	✓	**95.9** %	0.4 M

Table 2: Comparison of our Event-SSM to the state-of-the-art on the Spiking Heidelberg Digits dataset [Cramer et al. 2022].

Spiking Speech Commands	Async. events	Test accuracy	Num params
Bittar and Garner [2022]	✗	71.7 %	0.1 M
Bittar and Garner [2022]	✗	77.4 %	3.9 M
Hammouamri et al. [2024]	✗	79.8 %	0.7 M
Hammouamri et al. [2024]	✗	80.7 %	2.5 M
Event-SSM	✓	85.3 %	0.1 M
Event-SSM	✓	**88.4** %	0.6 M

Table 3: Comparison of our Event-SSM to the state-of-the-art on the Spiking Speech Commands dataset [Cramer et al. 2022].

This leads us to the conclusion that the much less saturated and larger scale SSC dataset is more appropriate for evaluating state-of-the-art methods.

Results for SSC are shown in tab. 3. Our method significantly outperforms the state-of-the-art set recently by Hammouamri et al. [2024] by a margin of almost 6.6 %, while using much fewer parameters.

4.2 Event-based vision processing

The DVS128 Gestures dataset was recorded with a dynamic vision sensor of 128×128 resolution [Lichtsteiner et al. 2008]. Each pixel is represented by two channels, encoding the two event polarities, resulting in $C = 128 \times 128 \times 2 = 32768$ asynchronous channels for the DVS128 Gestures dataset. The large number of asynchronous channels results in a very large number of events per second. An overview of the distribution of the number of events per sample across the classes of the DVS dataset can be obtained from fig. 3. Consequently, learning representations for event-based vision is one of the largest scale benchmarks for event-based processing systems. The previous best-performing baseline models collected events into a 4-d tensor representation which was then processed by convolutional neural networks composed of artificial or spiking neurons. Doing this mitigates the computational overhead of processing every event individually, and circumvents the need to process very long sequences of irregularly sampled events. In contrast, our event-based state-space model directly processes the event-stream recorded from the dynamic vision sensor with a recurrent neural network, without using spatial convolutions. Spatio-temporal representations are solely learned from the linear

DVS128 Gesture	Async. events	Test accuracy	Num params
Yousefzadeh et al. [2019]	✗	95.2 %	1.2 M
Xiao et al. [2022]	✗	96.9 %	-
Subramoney et al. [2023]	✗	97.8 %	4.8 M
She et al. [2022]	✗	98.0 %	1.1 M
Liu et al. [2022]	✗	**98.8** %	-
Martin-Turrero et al. [2024]	✗	96.2 %	14 M
Martin-Turrero et al. [2024]	✓	94.1 %	14 M
CNN + S5 (time-frames)	✗	97.8 %	6.8 M
CNN + S5 (event-frames)	✗	97.3 %	6.8 M
Event-SSM	✓	97.7 %	0.8 M + 4.2 M

Table 4: Comparison of our Event-SSM to the state-of-the-art on the DVS128-Gesture dataset [Amir et al. 2017]. We report our model's number of parameters as the parameters of the SSM + embedding look-up. Due to the sensor resolution, most parameters are learned embedding vectors rather than SSM parameters.

Model	Accuracy
Event-SSM	(86.9 ± 0.4) %
S5 with Dirac discretization	(84.6 ± 0.4) %
S5 with ZOH discretization	(74.4 ± 0.3) %
S5 with ZOH and $\Delta_k \equiv 1$	(80.8 ± 0.1) %

Table 5: A comparison of our proposed method (Event-SSM) with the Dirac discretized S5 model (14) (Dirac), S5 with zero-order hold discretization (ZOH), and S5 with all $\Delta_k \equiv 1$, i.e. without parsing information about event timings. We report means and standard deviations from 5 runs with random seeds on the SSC dataset.

state-space model and non-linear feedforward transformations. Despite breaking with the pervasive convention of binning events into time-frames and processing with CNNs, our event-based state-space model achieved competitive results as reported in tab. 4. To improve generalization, a combination of data augmentation methods such as spatial-jitter, time-jitter, random noise, drop-event, geometric augmentations [Li et al. 2022], and cut-mix were applied.

Training recurrent networks with backpropagation through time (BPTT) requires storing (or recomputing) the activations for every step in the sequence. The large number of events per sample, therefore, quickly saturated GPU memory. To fit reasonable batch sizes into the 40 GB HBM memory of our A100s, we sliced the training data into shorter sequences. Yet, evaluation was conducted on full samples of up to 1.5 M events. Surprisingly, we found that training on slices of 32 768 events suffices to reach the baseline performance.

4.3 Ablation study

In sec. 3.2, we argued that naively applying the discretization methods of most SSM works to asynchronous event-streams is not ideal. Tab. 5 compares our method as presented in eq. (17) with the popular zero-order hold (ZOH) method employed by J. T. Smith et al. [2023], the naive integration of Dirac delta pulses in eq. (12), and vanilla S5 without passing in the timestamps at all. Integrating events according to eq. (17) clearly improves the performance over the other methods. These results provide further evidence that event timings can improve representations of event-based systems.

5 DISCUSSION

This work presents a scalable method for the modeling of irregular event-stream data. Our method addresses the major challenges of event-based processing — long-range dependencies, asynchronous processing, and parallelization. The model operates directly on the address event representation of the event-stream and, in contrast to most related works, never uses 2D or 3D convolutions. The stable state-space model parameterization allows asynchronous recurrent training and inference on very long event-streams of more than a million events, such as those given by event-based vision sensors. Our ablation study shows that asynchronous event channels require discretization methods that have not yet been used in the machine learning literature. Furthermore, we observe a clear advantage of integrating exact temporal information compared to binning events into frames for the larger audio processing task SSC. This result also highlights the need to establish high-quality, large-scale datasets of event-streams for challenging machine learning tasks, that will allow us to carefully scrutinize the advantages and disadvantages of event-based machine learning.

Although our model and all its parts were carefully designed to most effectively operate directly with events, there is no explicit event-generating mechanism in the neural network itself. The event-based processing is solely driven by external events. On the one hand, this result breaks with the commonly held belief that event data is most effectively processed with event-based neural networks. On the other hand, this observation provides an interesting direction for future work to explore how the properties that allow our model to scale to long event-streams can be joined with the efficient processing paradigms of event-based networks.

ACKNOWLEDGMENTS

The authors thank Matthias Jobst for early discussions on the project as well as Jamie Lohoff, Erika Covi and Matthias Jobst for their valuable feedback on the manuscript. Mark Schöne is supported with funds from Bosch-Forschungsstiftung im Stifterverband. David Kappel is funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) project ESCADE (01MN23004A). Christian Mayr is affiliated to German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. Neeraj Mohan Sushma is funded by the German Federal Ministry of Education and Research (BMBF) project EVENTS (16ME0733). This work was partially funded by the German Federal Ministry of Education and Research (BMBF) and the free state of Saxony within the ScaDS.AI center of excellence for AI research.

REFERENCES

Arnon Amir et al.. 2017. "A Low Power, Fully Event-Based Gesture Recognition System." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7388–7397. DOI: 10.1109/CVPR.2017.781.

Abdul Fatir Ansari, Alvin Heng, Andre Lim, and Harold Soh. 23–29 Jul 2023. "Neural Continuous-Discrete State Space Models for Irregularly-Sampled Time Series." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. PMLR, (23–29 Jul 2023), 926–951.

Sami Barchid, José Mennesson, and Chaabane Djéraba. 2022. "Bina-Rep Event Frames: A Simple and Effective Representation for Event-Based Cameras." In: *2022 IEEE International Conference on Image Processing (ICIP)*, 3998–4002. DOI: 10.1109/ICIP46576.2022.9898061.

Y. Bengio, P. Simard, and P. Frasconi. 1994. "Learning long-term dependencies with gradient descent is difficult." *IEEE Transactions on Neural Networks*, 5, 2, 157–166. DOI: 10.1109/72.279181.

Alexandre Bittar and Philip N. Garner. 2022. "A surrogate gradient spiking baseline for speech command recognition." *Frontiers in Neuroscience*, 16.

Guy E. Blelloch. Nov. 1990. *Prefix Sums and Their Applications*. Tech. rep. CMU-CS-90-190. School of Computer Science, Carnegie Mellon University, (Nov. 1990).

James Bradbury et al.. 2018. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. (2018). http://github.com/google/jax.

Stefano Caviglia, Luigi Pinna, Maurizio Valle, and Chiara Bartolozzi. 2017. "Spike-Based Readout of POSFET Tactile Sensors." *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64, 6, 1421–1431. DOI: 10.1109/TCSI.2016.2561818.

Vincent Chan, Shih-Chii Liu, and Andr van Schaik. 2007. "AER EAR: A Matched Silicon Cochlea Pair With Address Event Representation Interface." *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54, 1, 48–59. DOI: 10.1109/TCSI.2006.887979.

Loic Cordone, Benoît Miramond, and Sonia Ferrante. July 2021. "Learning from Event Cameras with Sparse Spiking Convolutional Neural Networks." In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*. (July 2021).

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. 2022. "The Heidelberg Spiking Data Sets for the Systematic Evaluation of Spiking Neural Networks." *IEEE Transactions on Neural Networks and Learning Systems*, 33, 7, 2744–2757. DOI: 10.1109/TNNLS.2020.3044364.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. June 2017. "Language Modeling with Gated Convolutional Networks." In: *Proceedings of the 34th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by Doina Precup and Yee Whye Teh. Vol. 70. PMLR, (June 2017), 933–941.

Soham De et al.. Feb. 2024. "Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models." arXiv:2402.19427 [cs] type: article. (Feb. 2024). DOI: 10.48550/arXiv.2402.19427.

Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Dec. 2023. "Parallel Spiking Neurons with High Efficiency and Ability to Learn Long-term Dependencies." *Advances in Neural Information Processing Systems*, 36, (Dec. 2023), 53674–53687.

Daniel Gehrig, Antonio Loquercio, Konstantinos Derpanis, and Davide Scaramuzza. 2019. "End-to-End Learning of Representations for Asynchronous Event-Based Data." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5632–5642. DOI: 10.1109/ICCV.2019.00573.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Re. 17–23 Jul 2022. "It's Raw! Audio Generation with State-Space Models." In: *Proceedings of the 39th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. PMLR, (17–23 Jul 2022), 7616–7633. https://proceedings.mlr.press/v162/goel22a.html.

Alex Graves and Jürgen Schmidhuber. 2008. "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 21. Curran Associates, Inc.

Albert Gu and Tri Dao. Dec. 2023. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces." arXiv:2312.00752 [cs] type: article. (Dec. 2023). DOI: 10.48550/arXiv.2312.00752.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. 2022. "On the Parameterization and Initialization of Diagonal State Space Models." In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho.

Albert Gu, Karan Goel, and Christopher Ré. 2022. "Efficiently Modeling Long Sequences with Structured State Spaces." In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ilyass Hammouamri, Ismail Khalfaoui-Hassani, and Timothée Masquelier. 2024. "Learning Delays in Spiking Neural Networks using Dilated Convolutions with Learnable Spacings." In: *The Twelfth International Conference on Learning Representations*.

Sepp Hochreiter. 1991. "Untersuchungen zu dynamischen neuronalen Netzen." *Diploma, Technische Universität München*, 91, 1, 31.

S. Innocenti, F. Becattini, F. Pernici, and A. Del Bimbo. Jan. 2021. "Temporal Binary Representation for Event-Based Action Recognition." In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society, Los Alamitos, CA, USA, (Jan. 2021), 10426–10432. DOI: 10.1109/ICPR48806.2021.9412991.

Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman. 2017. "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 7, 1346–1359. DOI: 10.1109/TPAMI.2016.2574707.

Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. 2022. "Neuromorphic Data Augmentation for Training Spiking Neural Networks." In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*. Springer-Verlag, Tel Aviv, Israel, 631–649. ISBN: 978-3-031-20070-0. DOI: 10.1007/978-3-031-20071-7_37.

Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. 2008. "A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor." *IEEE Journal of Solid-State Circuits*, 43, 2, 566–576. DOI: 10.1109/JSSC.2007.914337.

Chang Liu, Xiaojuan Qi, Edmund Y. Lam, and Ngai Wong. 2022. "Fast Classification and Action Recognition With Event-Based Imaging." *IEEE Access*, 10, 55638–55649. DOI: 10.1109/ACCESS.2022.3177744.

Carmen Martin-Turrero, Maxence Bouvier, Manuel Breitenstein, Pietro Zanuttigh, and Vincent Parret. 2024. *ALERT-Transformer: Bridging Asynchronous and Synchronous Machine Learning for Real-Time Event-based Spatio-Temporal Data*. (2024). arXiv: 2402.01393 [cs.CV].

Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. 2020. "Event-Based Asynchronous Sparse Convolutional Networks." In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Springer International Publishing, Cham, 415–431. ISBN: 978-3-030-58598-3.

Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. "Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences." In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc.

Antonio Orvieto, Soham De, Caglar Gulcehre, Razvan Pascanu, and Samuel L. Smith. July 2023. "On the Universality of Linear Recurrences Followed by Nonlinear Projections." arXiv:2307.11888 [cs] type: article. (July 2023). DOI: 10.48550/arXiv.2307.11888.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. July 2023. "Resurrecting recurrent neural networks for long sequences." In: *Proceedings of the 40th International Conference on Machine Learning* (ICML'23). Vol. 202. JMLR.org, (July 2023), 26670–26698.

Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. 2020a. "Learning to Detect Objects with a 1 Megapixel Event Camera." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 16639–16652.

Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. 2020b. "Learning to Detect Objects with a 1 Megapixel Event Camera." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 16639–16652.

Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. 2011. "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS." *IEEE Journal of Solid-State Circuits*, 46, 1, 259–275. DOI: 10.1109/JSSC.2010.2085952.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. 2022. "Modeling Irregular Time Series with Continuous Recurrent Units." In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA* (Proceedings of Machine Learning Research). Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato. Vol. 162. PMLR, 19388–19405.

Xueyuan She, Saurabh Dash, and Saibal Mukhopadhyay. 2022. "Sequence Approximation using Feedforward Spiking Neural Network for Spatiotemporal Learning: Theory and Optimization Methods." In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. 2023. "Simplified State Space Layers for Sequence Modeling." In: *The Eleventh International Conference on Learning Representations*.

Anand Subramoney, Khaleelulla Khan Nazeer, Mark Schöne, Christian Mayr, and David Kappel. 2023. "Efficient recurrent architectures through activity sparsity and sparse back-propagation through time." In: *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=lJdOlWg8td.

Aaron Voelker, Ivana Kajić, and Chris Eliasmith. 2019. "Legendre Memory Units: Continuous-Time Representation in Recurrent Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc.

Timo C Wunderlich and Christian Pehle. 2021. "Event-based backpropagation can compute exact gradients for spiking neural networks." *Scientific Reports*, 11, 1, 12829.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. 2022. "Online Training Through Time for Spiking Neural Networks." In: *Advances in*

Neural Information Processing Systems. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 20717–20730.

Sidi Yaya Arnaud Yarga and Sean U. N. Wood. June 2023. "Accelerating SNN Training with Stochastic Parallelizable Spiking Neurons." In: *2023 International Joint Conference on Neural Networks (IJCNN)*. arXiv:2306.12666 [cs]. (June 2023), 1–8. DOI: 10.1109/IJCNN54540.2023.10191884.

Amirreza Yousefzadeh et al.. 2019. "Asynchronous Spiking Neurons, the Natural Key to Exploit Temporal Sparsity." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9, 4, 668–678. DOI: 10.1109/JETCAS.2019.2951121.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. 2019. "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features." In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 6022–6031. DOI: 10.1109/ICCV.2019.00612.

Nikola Zubić, Mathias Gehrig, and Davide Scaramuzza. 2024. *State Space Models for Event Cameras*. (2024). arXiv: 2402.15584 [cs.CV].