Deep Directly-Trained Spiking Neural Networks for Object Detection

Qiaoyi Su[1], Yuhong Chou[2,7], Yifan Hu[3], Jianing Li[4], Shijie Mei[5,7], Ziyang Zhang[6], Guoqi Li[1,7*]

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]College of Artificial Intelligence, Xi'an Jiaotong University
[3]Department of Precision Instrument, Tsinghua University
[4]School of Computer Science, Peking University
[5]School of Vehicle and Mobility, Tsinghua University
[6]Advanced Computing and Storage Lab, Huawei Technologies Co. Ltd.
[7]Institute of Automation, Chinese Academy of Sciences

suqiaoyi2020@ia.ac.cn, huyf19@mails.tsinghua.edu.cn, lijianing@pku.edu.cn, guoqi.li@ia.ac.cn

Abstract

Spiking neural networks (SNNs) are brain-inspired energy-efficient models that encode information in spatiotemporal dynamics. Recently, deep SNNs trained directly have shown great success in achieving high performance on classification tasks with very few time steps. However, how to design a directly-trained SNN for the regression task of object detection still remains a challenging problem. To address this problem, we propose EMS-YOLO, a novel directly-trained SNN framework for object detection, which is the first trial to train a deep SNN with surrogate gradients for object detection rather than ANN-SNN conversion strategies. Specifically, we design a full-spike residual block, EMS-ResNet, which can effectively extend the depth of the directly-trained SNN with low power consumption. Furthermore, we theoretically analyze and prove the EMS-ResNet could avoid gradient vanishing or exploding. The results demonstrate that our approach outperforms the state-of-the-art ANN-SNN conversion methods (at least 500 time steps) in extremely fewer time steps (only 4 time steps). It is shown that our model could achieve comparable performance to the ANN with the same architecture while consuming 5.83× less energy on the frame-based COCO Dataset and the event-based Gen1 Dataset. Our code is available in https://github.com/BICLab/EMS-YOLO.

1. Introduction

Object detection is a key and challenging problem in computer vision. It aims to recognize multiple overlapped objects and locate them with precise bounding boxes. This task has many applications in various fields, such as autonomous driving [3], security surveillance [47], and medical imaging [29]. Most existing frameworks (*e.g.*, YOLO series [38], RCNN series [13]) for object detection use artificial neural networks (ANNs), which have high performance but also high computational complexity and energy consumption. Spiking neural networks (SNNs), known as the third generation of neural networks [32, 40], potentially serves as a more efficient and biologically inspired way to perform object detection. Specifically, SNNs utilize binary signals (spikes) instead of continuous signals for neuron communication, which reduces data transfer and storage overhead. Furthermore, the SNNs exhibit asynchronous computation and event-driven communication, which could avoid unnecessary computation and synchronization overhead. When deployed on neuromorphic hardware [33, 36], SNNs show great energy efficiency.

However, most SNNs for object detection are converted from ANNs, which have some limitations. For example, Spiking-Yolo [23, 22] needs at least 3500 time steps to match the performance of the original ANN. Spike Calibration [26] can reduce the time steps to hundreds, but it still depends on the performance of the original ANN model. Moreover, most methods for converted SNNs are only applicable for static images, and not suitable for sparse event datas, because their dynamics are designed to approximate the expected activation of the ANN and fail to capture the spatiotemporal information of DVS data [10]. A promising approach is to train SNNs directly with surrogate gradient, which can achieve high performance with few time steps and process both static images and event data efficiently.

---

*Corresponding author

Another challenge is to deal with the multi-scale object features in object detection, which demands the network has sufficient representation capability. Correspondingly, deep structure training is needed. Existing models on object detection are limited to shallow structures [7] or hybrid structures [27, 20] that may not be deployed on some neuromorphic hardware [8, 2, 30] where only spike operation is allowed. To achieve deep direct training of SNNs, Hu *et al.* [17] and Fang *et al.* [12] explored on classification tasks and proposed MS-ResNet and SEW-ResNet respectively to overcome the vanishing/exploding gradient problems and advanced the directly-trained SNNs to depths greater than 100 layers. Unfortunately, multi-scale transformation of channels and dimensions when extracting features of objects of different sizes will cause the problem of increased energy consumption due to non-spike convolutional operations in their networks to be prominent in the object detection task. Therefore, the problem of high energy consumption caused by the non-spike convolutional operations urgently needs to be addressed.

To tackle these problems, we propose a novel directly trained SNN for object detection based on the YOLO framework (EMS-YOLO). Our model is the first to use surrogate gradients to train a deep and large-scale SNN for object detection without converting from ANNs. Specially, to deal with the multi-scale object features in object detection, we design a new full-spike energy-efficient residual block, EMS-ResNet, that avoids redundant MAC operations caused by non-spike convolution. Our model can achieve high performance with few time steps and handle both static images and event data efficiently. Compared with other converted or hybrid SNNs for object detection, our model has higher performance and lower energy consumption.

Our major contributions of this paper can be summarized as follows:

- We propose EMS-YOLO, a novel directly trained spiking neural network for object detection, which could achieve better performance than the advanced ANN-SNN conversion methods while requiring only 4 time steps and inferencing in real time.

- We design an **E**nergy-efficient **M**embrane-**S**hortcut ResNet, EMS-ResNet, that enables full spiking in the network thus reducing power consumption. Moreover, we theoretically analyze that it can be trained deeply since it avoids gradient disappearance or explosion.

- The experiments on COCO and the Gen1 Datasets demonstrate that our models could achieve comparable performance to the ANN with the same architecture meanwhile reducing $5.83\times$ energy consumption.

2. Related Work

2.1. Deep Spiking Neural Networks

The training strategies for deep SNNs are mainly divided into ANN-SNN conversion and directly training SNNs. The essence of ANN-SNN conversion is to approximate the average firing rate of SNNs to the continuous activation value of ANNs that use ReLU as the nonlinearity [5, 11]. The performance of the converted SNN relies on the original ANN, and it is tough to get high-performance SNNs with low time delay and suffers performance losses during conversion. Moreover, the converted SNNs cannot work well on the sparse event data which can be effectively combined with neuromorphic hardware.

Inspired from the explosive development of ANNs, the researchers use the surrogate gradient to achieve directly training of SNNs. The mature back-propagation mechanism and diverse coding schemes [34, 21] have enabled directly-trained SNNs to work well on short time steps while requiring low power consumption. Zheng *et al.* [49] proposed the threshold-dependent batch normalization (TDBN) technique which could extend SNNs from a shallow structure (<10 layers) to a deeper structure (50 layers) based on the framework of STBP [42]. Hu *et al.* [17] and Fang *et al.* [12] advanced the achievement of high performance on classification tasks. Currently, there is some work that uses deep directly training SNN for regression tasks like object tracking [48, 44], image reconstruction [50], while the object detection task has not yet been explored.

2.2. Energy-Efficient Object Detection

Significant vision sensors in object detection include frame-based and event-based cameras [25, 37], where the latter can handle challenging scenes such as motion blur, overexposure, and low light, and have become a hot spot in recent years. The mainstream deep learning-based detectors mainly fall into two categories: two-stage frameworks (RCNN series) [13], and one-stage frameworks (YOLO series [38], SSD [31], Transformer series [51]). They are implemented based on ANNs, which achieve high performance but also bring high energy consumption. Therefore, some explorations of SNN-based object detection have attempted to provide more energy-efficient solutions.

The earliest attempts [23, 22, 26] were based on ANN-SNN conversion method which requires long inference time and cannot be applied to event camera data due to the inherent limitations of the approach. Some hybrid architectures [27, 20] tried to use directly trained SNN backbones and ANN detection heads for object detection, while these detection heads introduce an additional amount of parameters. In this work, we present the first attempt of object detection with fully and deep directly trained SNNs.

2.3. Spiking Residual Networks

The ANN-SNN conversion does not concern the design of residual blocks oriented to the trainability of deep SNNs, while the residual block is mainly utilized to achieve loss-less accuracy [18, 41]. For directly training of SNNs, the residual structure empowers it to train deeply. Zheng *et al.* [49] firstly obtain directly trained ResNet-34/50 with surrogate gradient. SEW-ResNet [12] and MS-ResNet [17] extended the depth of the SNN model to over 100 layers. However, the former is essentially an integer multiplication operation of the residual network. The latter focuses mainly on the spiking of residual paths and ignores the non-spiking structure on shortcut paths. When applied to object detection tasks with varying dimensionality and number of channels, these non-spike convolutions can result in heavy energy consumption. Therefore, we design a full spiking residual network to exploit the energy efficiency of SNN.

3. The Preliminaries of SNNs

3.1. Spiking Neuron

Neurons are the basic units of a neural network, which convert a barrage of synaptic inputs into meaningful action potential outputs. In ANNs, artificial neurons discard temporal dynamics and propagate information only in the spatial domain. In contrast, in SNNs, spiking neurons are more biologically plausible as they mimic the membrane potential dynamics and the spiking communication scheme [24]. Typically, the Leaky Integrate-and-Fire (LIF) model [1], the Hodgkin-Huxley (H-H) model [15] and the Izhikevich model [19] are the most famous ones. Among them, LIF models are widely adopted to construct SNNs due to its good trade-off between bio-plausibility and complexity. Also, they contain abundant biological features and consume less energy than the ANN neurons. In our work, we use the iterative LIF model proposed by Wu *et al.* [43] in the SNN model, which can be described as:

$$V_i^{t+1,n+1} = \tau V_i^{t,n+1}(1 - X_i^{t,n+1}) + \sum_j W_{ij}^n X_j^{t+1,n} \quad (1)$$

$$X_i^{t+1,n+1} = H(V_i^{t+1,n+1} - V_{th}) \quad (2)$$

where the $V_i^{t,n+1}$ is the membrane potential of the i-th neuron in the $n + 1$ layer at the timestep t, τ is a decay factor for leakage. The input to a synapse is the sum of j spikes $X_j^{t+1,n}$ with synaptic weights W_{ij}^n from the previous layer n. $H(\cdot)$ denotes the Heaviside step function which satisfies $H(x) = 1$ for $x \geq 0$, otherwise $H(x) = 0$. As shown in the Figure 2, a firing activity is controlled by the threshold V_{th}, and the $V^{t+1,n+1}$ will be reset to V_{rest} once the neuron emits a spike at time step $t + 1$.

3.2. Training Strategies

In order to solve the problem that the spike cannot be differentiated in back-propagation, we use the surrogate gradient [42] which can be represented as:

$$\frac{\partial X_i^{t,n}}{\partial V_i^{t,n}} = \frac{1}{a} sign(|V_i^{t,n} - V_{th}| \leq \frac{a}{2}) \quad (3)$$

where a is introduced to ensure the integral of the gradient is 1 and determines the curve steepness.

We use the TDBN [49] normalization method which considers spatial and temporal domains. The TDBN can be described as:

$$V_i^{t+1,n+1} = \tau V_i^{t,n+1}(1 - X_i^{t,n+1}) + \text{TDBN}(I_i^{t+1}) \quad (4)$$

$$\text{TDBN}(I_i^{t+1}) = \lambda_i \frac{\alpha V_{th}(I_i^{t+1} - \mu_{ci})}{\sqrt{\sigma_{ci}^2 + \epsilon}} + \beta_i \quad (5)$$

where μ_{ci}, σ_{ci}^2 are the mean and variation values for every channel using a mini-batch of sequential inputs $\{I_i^{t+1} = \Sigma_j W_{ij}^n X_j^{t+1,n} | t = 0, ..., T-1\}$, ϵ is a tiny constant to avoid dividing by zero, λ_i, β_i are two trainable parameters, and α is a threshold-dependent hyper-parameter.

3.3. Energy Consumption

The number of operations is often used to measure the computational energy consumption of neuromorphic hardware. In ANNs, each operation involves multiplication and addition (MAC) of floating point numbers, and Times of floating-point operations (FLOPs) are used to estimate computational burden. SNNs have the energy-efficient property in neuromorphic hardware since the neurons only participate in the accumulation calculation (AC) when they spike and could achieve the calculation with about the same number of synaptic operations (SyOPs). However, many current SNNs introduce additional MAC operations due to their design flaws. Thus, we quantify the energy consumption of vanilla SNNs as $E_{SNN} = \sum_n E_b$, for each block n:

$$E_b = T \times (fr \times E_{AC} \times OP_{AC} + E_{MAC} \times OP_{MAC}) \quad (6)$$

where T and fr represents the total time steps and the block firing rate. The blocks are normally convolutional or fully connected, and the energy consumption is determined by the number of AC and MAC operations (OP_{AC}, OP_{MAC}). In this work, we adopt the same structure of SNN and ANN to compare the energy consumption and assume that the data for various operations are 32-bit floating-point implementation in 45nm technology [16], where $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ$.
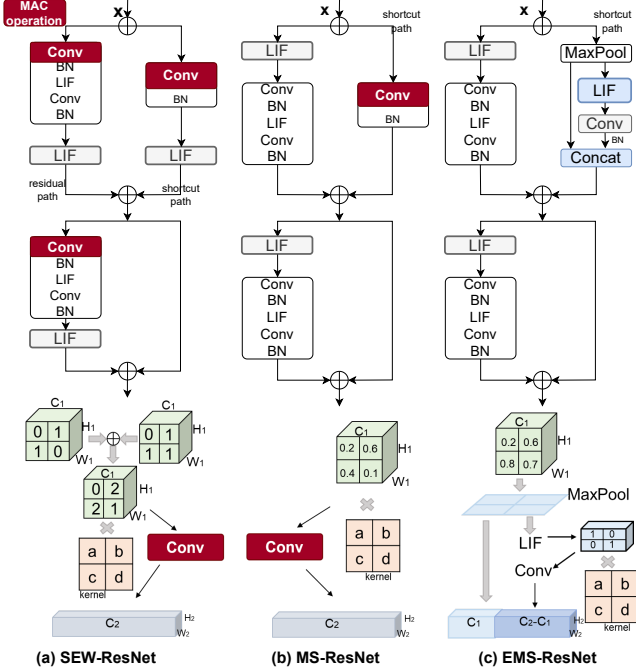
Figure 1. **The Sew-ResNet, the MS-ResNet and proposed EMS-ResNet.** (a) The sum of spikes in SEW-ResNet causes non-spike convolution operations. (b) MS-ResNet introduces non-spike convolution on shortcut paths where the dimensionality and number of channels changes. (c) The full-spike EMS-ResNet.

4. Method

4.1. Input Representation

Static Images Inputs Typically, considering the spatio-temporal feature of SNNs, the static images generated by the frame cameras are copied and used as the input frame for each time step [17, 12, 46].

Event-based Inputs Event cameras work completely differently from frame cameras where each pixel responds independently to changes in light. An event $e_n = (x_n, y_n, t_n, p_n)$ is generated for a pixel (x_n, y_n) at the timestamp t_n when the logarithmic light change $I(x, y, t)$ exceeds the threshold θ_{th}. The polarity $p_n \in \{-1, 1\}$ denotes the increase or decrease of light intensity.

Given the spatio-temporal window ζ, the asynchronous event stream $E = \{e_n \in \zeta : n = 1, ..., N\}$ represents a sparse grid of points in 3D space. In this work, we split E into temporal bins with a constant temporal window dt, which maps the events into image-like 2D representations [45]. The network processes T fixed time steps each time, and the total sequence $\Gamma = T \times dt$.

4.2. The Energy-Efficient Resisual Block

Currently, the main structures used for deep directly training in SNNs are SEW-ResNet and MS-ResNet. Here,

we define the output of the L-th residual block, denoted by $\mathbf{X}^L$, as:

$$\mathbf{X}^L = Add(\mathbf{F}^r(\mathbf{X}^{L-1}), \mathbf{F}^s(\mathbf{X}^{L-1})) \qquad (7)$$

where the residual path is $\mathbf{F}^r(\cdot)$, and the shortcut path is represented as $\mathbf{F}^s(\cdot)$. As shown in Figure 1a, SEW-ResNet implements residual learning by restricting the final LIF activation function to each of the residual and shortcut paths. When both paths transmit spikes, $Add(\mathbf{F}^r(\mathbf{X}^{L-1}), \mathbf{F}^s(\mathbf{X}^{L-1})) = 2$, which means that the non-spike convolution operation in the next block will introduce MAC operations. Although using either AND or IAND operation to avoid this problem has been tried, it comes with an unbearable performance loss. For the MS-ResNet in the Figure 1b, it is apparent that it ignores the non-spike convolution operations on shortcut paths. When the dimensionality of the network or the number of channels is constantly changing, the energy consumption caused by this component is not negligible. According to Equation 6, to achieve energy efficiency, we design a full-spike residual block (EMS-Block) from the perspective of reducing parameters, and avoiding MAC operations.

As illustrated in Figure 1c, our proposed EMS-ResNet is designed differently for channel number and dimension variation. The idea of MS-ResNet is adopted on the residual path. On the shortcut path, when the number of channels changes, an additional LIF is added before the convolution for converting the information into sparse spikes which enables the entire network can be fully spiked. We transform the EMS-Block into two formats (see Figure 2). The EMS-Block1 is used when the channel number is constant or decreasing. For increasing channel numbers, we design the EMS-Block2 that uses concatenation operations to feature reuse while reducing parameters. Considering features are usually extracted by constant downsampling in the object detection task, we use the maxpool to reduce parameters. Our design improves flow of information and the output flow on the shortcut path is conceptually approximate to the sum of synaptic inputs to neuronal membranes. The full spike feature implies energy efficient properties for the network, so we name our network as the Energy-Efficient Menbrane-Shortcut ResNet (EMS-ResNet).

4.3. The EMS-YOLO Model

Our goal is to predict the classification and position of the objects from the given static image or event stream that can be represented as $\mathbf{X} = \{X_t\}_{t=1}^T$. The dimensions of the input data X_t for each time step t are $C \times H \times W$, where C is the channel number, $H \times W$ is the resolution. To achieve this, we calculate the information of N objects $\mathbf{B} = \{B_n\}_{n=1}^N$ by:

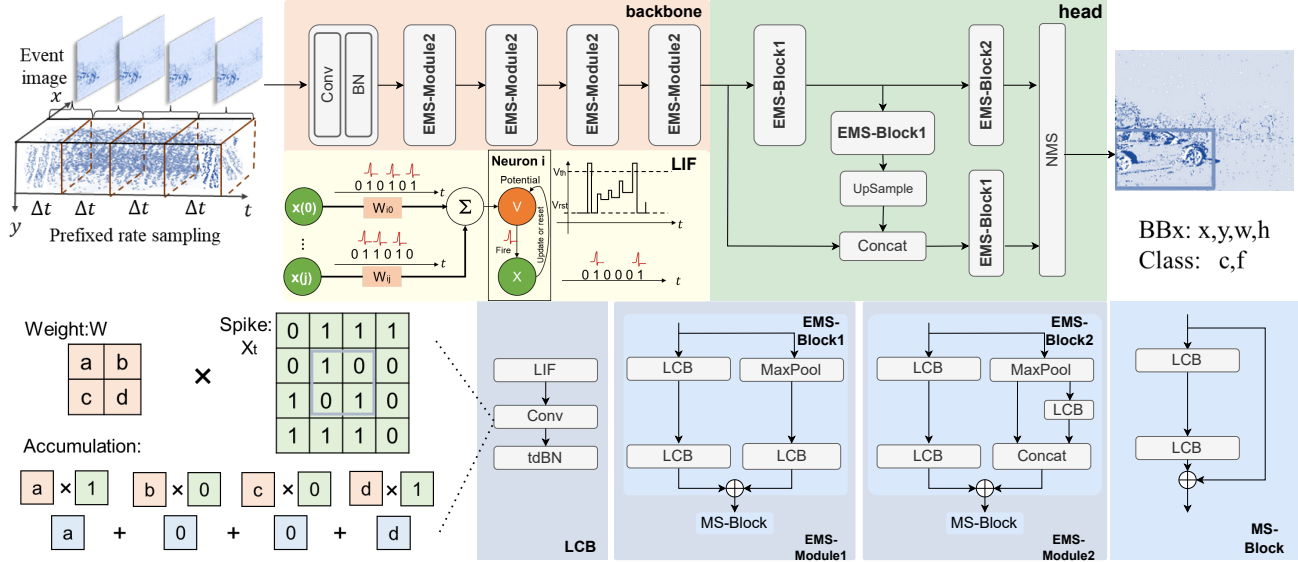$$\mathbf{B} = \mathcal{D}(\mathbf{X}) \qquad (8)$$

Figure 2. **The proposed directly-trained SNN for object detection (EMS-YOLO)**. EMS-YOLO mainly consists of backbone and head, which are mainly composed of EMS-Blocks. EMS-Module1 and EMS-Module2 are EMS-Block1 and MS-Block, EMS-Block2 and MS-Block connections respectively. EMS-Block2 is used when the number of output channels increases, otherwise, EMS-Block1 is used.

where each $B_n = \{x_n, y_n, w_n, h_n, c_n, f_n\}$ is the corresponding object's bounding box position and class prediction. $(x_n, y_n), w_n, h_n$ represent the upper-left spatial coordinates, width, and length of the bounding box, respectively. c_n, f_n are the class and confidence level of the object, respectively. $\mathcal{D}$ refers to our proposed EMS-YOLO, which could achieve performance comparable to that of a same-structure ANN, while inferring in real time.

As shown in Figure 2, our EMS-YOLO is a variation of the YOLO framework, consisting of two modules: the backbone for feature extraction and the detection heads. The analog pixel values $\mathbf{X}$ of an image are directly applied to the input layer of EMS-YOLO. For the backbone, the first convolutional layer is trained to convert inputs into spikes where LIF neurons integrate the weighted inputs and generate output spike trains when the membrane potential exceeds the trained firing threshold. Then we employ the EMS-Modules to extract object features from different dimensions and number of channels, which can enhance the robustness of the network. The number of EMS-Blocks and the channel width can be dynamically adjusted to the particular task, that Ablation experiments are shown in Sec 5.3, and EMS-ResNet18 is shown in the Figure 2 as an example.

For the detection head, to avoid the loss of SNN performance due to the multi-layer direct-connected convolution structure [49] of the conventional detection head, in this work we take the yolov3-tiny detection head [39] as an example and replace multiple directly connected convolutions with EMS-Blocks.

The main challenge of object detection, as a regression task with SNN models, is to convert the features extracted

with spike trains into accurate continuous value representations of the bounding boxes coordinates. Here, we feed last membrane potential of the neurons [50] into each detector to generate anchors of different sizes. After NMS processing, the final category and bounding box coordinates of different objects can be obtained. Our end-to-end EMS-YOLO is trained with the cross-entropy loss function[1].

4.4. Analysis of Gradient Vanishing/Explosion Problems

To explore the potential of our EMS-ResNet to enable deep training, we analyze the feasibility from a theoretical perspective. According to Gradient Norm Equality (GNE) theory [6], our EMS-ResNet could avoid exploding and vanishing gradients.

Lemma 1. *Consider a neural network that can be represented as a series of blocks as and the jth block's jacobian matrix is denoted as J_j. If $\forall j; \phi(J_j J_j^\mathsf{T}) \approx 1$ and $\varphi(J_j J_j^\mathsf{T}) \approx 0$, the network achieves "Block Dynamical Isometry" and can avoid gradient vanishing or explosion.*

Here, J_j means the Jacobian matrix of the block j, j is the index of the corresponding block. ϕ means the expectation of the normalized trace. φ means $\phi(\mathbf{A}^2) - \phi^2(\mathbf{A})$.

Briefly, the theory ensures the gradient of the network will not decrease to 0 or explode to ∞ since every block have $\phi(J_j J_j^\mathsf{T}) \approx 1$. And $\varphi(J_j J_j^\mathsf{T}) \approx 0$ makes sure that accident situation won't happen. And in most cases[49][17],

---

[1]https://github.com/ultralytics/yolov3

$\phi(J_j J_j^\mathsf{T}) \approx 1$ is enough for avoiding gradient vanish or exploding. Detailed description of the notation and the theory are in [6].

Definition 1. *(General Linear Transform) Let $\boldsymbol{f}(\boldsymbol{x})$ be a transform whose Jacobian matrix is $\boldsymbol{J}$. f is called general linear transform when it satisfies:*

$$E\left[\frac{\|\boldsymbol{f}(\boldsymbol{x})\|_2^2}{\text{len}(\boldsymbol{f}(\boldsymbol{x}))}\right] = \phi\left(\boldsymbol{J}\boldsymbol{J}^T\right) E\left[\frac{\|\boldsymbol{x}\|_2^2}{\text{len}(\boldsymbol{x})}\right]. \quad (9)$$

According to the random matrix and mean field theory [35], the data flow propagating in the network can be regarded as random variables. Thus, $\boldsymbol{x}$ is considered as a random variable. We denote the notation $E\left[\frac{\|\boldsymbol{x}\|_2^2}{\text{len}(\boldsymbol{x})}\right]$ as the 2th moment of input element.

The definition is useful for analysis the gradient. Because once the output of an EMS-Block is normalized by the BN layer, the 2th moment $E\left[\frac{\|\boldsymbol{f}(\boldsymbol{x})\|_2^2}{\text{len}(\boldsymbol{f}(\boldsymbol{x}))}\right]$ which we denote as α_2 is clear.

Lemma 2 (Multiplication). *(Theorem 4.1 in [6]) Given $\boldsymbol{J} := \prod_{j=L}^{1} \boldsymbol{J}_j$, where $\{\boldsymbol{J}_j \in \mathbb{R}^{m_j \times m_{j-1}}\}$ is a series of independent random matrices. If $(\prod_{j=L}^{1} \boldsymbol{J}_j)(\prod_{j=L}^{1} \boldsymbol{J}_j)^T$ is at least the 1st moment unitarily invariant, we have*

$$\phi\left((\prod_{j=L}^{1} \boldsymbol{J}_j)(\prod_{j=L}^{1} \boldsymbol{J}_j)^T\right) = \prod_{j=L}^{1} \phi(\boldsymbol{J}_j\boldsymbol{J}_j^T). \quad (10)$$

Lemma 3 (Addition). *(Theorem 4.2 in [6]) Given $\boldsymbol{J} := \prod_{j=L}^{1} \boldsymbol{J}_j$, where $\{\boldsymbol{J}_j \in \mathbb{R}^{m_j \times m_{j-1}}\}$ is a series of independent random matrices. If at most one matrix in $\boldsymbol{J}_j$ is not a central matrix, we have*

$$\phi(\boldsymbol{J}\boldsymbol{J}^T) = \sum_j \phi(\boldsymbol{J}_j\boldsymbol{J}_j^T). \quad (11)$$

The multiplication and addition principle provide us a technique for analysising the network with serial connections and parallel connections.

Discussion for General Linear Transform. Since the Jacobian matrix of pooling can be regarded as a matrix $\boldsymbol{J}$ that the element of matrix $[\boldsymbol{J}]_{ik} \in \{0, 1\}$, 0 for the element not chosen while 1 for element chosen. Thus the pooling layer can be regarded as a general linear transform. Similarly, the UpSampling layer in the detection head is also a generalized linear transformation. Concatenation is general linear transform too, because the function of $\boldsymbol{f}(\boldsymbol{x}) = [\boldsymbol{x}, \tilde{\boldsymbol{f}}(\boldsymbol{x})]$ can be expressed as $\boldsymbol{f}(\boldsymbol{x}) = [\boldsymbol{I} \quad \tilde{\boldsymbol{J}}]\boldsymbol{x}$ if the function $\tilde{\boldsymbol{f}}$ is general linear transform. Since the BN and

CONV layers are already discussed in [6], so we only need to assume that the LIF layer satisfy general linear transformation, which is already used in proof by [49]. Since the EMS-ResNet is a serial of basic EMS-Blocks and MS-Blocks, we can separately analysis those blocks and multiply them together.

Proposition 1. *For EMS-Block1 and EMS-Block2, the Jacobian matrix of the block can be represented as $\phi(\boldsymbol{J}_j\boldsymbol{J}_j^T) = \frac{2}{\alpha_2^{j-1}}$.*

Proof. It is detailed in **Supplementary Material Proof A.1 and Proof A.2**. □

Proposition 2. *For the EMS-ResNet, $\phi(\boldsymbol{J}\boldsymbol{J}^T) \approx 1$ can be satisfied by control the 2th moment of the input.*

Proof. It is detailed in **Supplementary Material Proof A.3**. □

Discussion about the proposition. According to [6], $\phi(\boldsymbol{J}\boldsymbol{J}^T) \approx 1$ is the formal expression for GNE which ensures gradient vanishing or explosion will not happen. However, it is enough for solving gradient problems by just avoiding the exponential increase or decrease [6]. In our backbone, the MS-Block, the only factor to evoke exponential increase of gradient, whose increasing is stopped by the interspersing of EMS-Blocks. Even if the initialize of BN do not satisfy the constant for $\phi(\boldsymbol{J}\boldsymbol{J}^T) \approx 1$, the gradient of each block will not increase as the network goes deeper. Briefly, we improve the property of network at a structural level by our EMS-Blocks.

5. Experiments

To fully validate the effectiveness of our model, we conduct experiments on the frame-based COCO2017 Dataset [28] and the event-based Gen1 Dataset [9]. The detailed introduction of the Datasets are in the **Supplementary Material B**. For the object detection task, the most widely used model evaluation metric is the mean Average Precision (mAP). We report the mAP at IOU=0.5 (mAP@0.5) and the average AP between 0.5 and 0.95 (mAP@0.5:0.95).

5.1. Experimental Setup

In all experiments, the number of detection heads is set to 2 to make a fair comparison with the previous work. The effect of different numbers of detection heads on the results is in the **Supplementary Material**. We set the reset value V_{reset} of LIF neurons to 0, the membrane time constant τ to 0.25, the threshold V_{th} to 0.5, and the coefficient α to 1. We mainly train models on 8 NVIDIA RTX3090 GPUs and adopt the SGD optimizer, the learning rate is set to $1e^{-2}$. The network is trained for 300 epochs on the COCO2017

Method	Work	Model	Time Step(T)	mAP@0.5
Backpropagation	ANN	Tiny-Yolo	/	0.262
		ResNet50	/	0.460
		ResNet34	/	0.565
ANN-SNN Conversion	Spiking-Yolo [23]	Tiny-Yolo	3500	0.257
	Bayesian Optimization [22]		500	0.211
			5000	0.258
	Spike Calibration [26]	ResNet50+Burst+ MLIPooling+SpiCalib	64	0.331
			128	0.436
			256	0.453
			512	0.454
Directly-Trained SNN	EMS-YOLO	EMS-ResNet34	**4**	**0.501**

Table 1. Results on the COCO2017 DataSet.

Dataset with a batch size of 32. On the Gen1 Dataset, we trained the model for 100 epochs, with the batch size of 128.

5.2. Effective Test

COCO2017 Dataset Our models are trained on Mosic data augmentation techniques [4] and tested to detect 80 objects of the COCO Dataset [28] on the validation set. As the first one to implement object detection task with directly trained SNNs, the main quantitative results are presented in Table 1. For comparison with the current best performance model Spike Calibration [26] based on ANN-SNN conversion method, we do experiments based on EMS-ResNet34, which has a reasonable number of parameters compared with ResNet50. We demonstrate that the directly training method can achieve higher performance at only 4 time steps, while the ANN-SNN conversion approach requires at least a several hundred time steps. Furthermore, with the same experimental settings and data augmentation methods, the performance of our model is comparable to that of the ANN with the same structure while reducing **5.83**× of energy consumption with the firing rate of 22.46%.

Method	Model	Params	T	Firing Rate	mAP @0.5:0.95	mAP @0.5
Sew-Reset	VGG -11+SSD	12.64M	5	22.22%	0.174	/
	MobileNet -64+SSD	24.26M	5	29.44%	0.147	/
	DenseNet 121-24+SSD	8.20M	5	37.20%	0.189	/
ANN	ResNet10		/	/	0.247	0.504
EMS-ResNet	EMS-Res10	**6.20M**	5	**21.15%**	**0.267**	**0.547**

Table 2. Results on Gen1 Dataset.

GEN1 Automotive Detection Dataset As the largest event camera-based dataset, it contains two categories (pedestrians and cars) [9]. Currently, only Cordone *et al.* [7] have experimented with the SNN method so far. They divided the event data into five temporal bins for each input to the network. In this paper, all the experiments on the Gen1 Dataset follow the same experimental setup with them.

The backbone of their shallow model is based on the lightweight structure of Sew-ResNet, and the detection head uses SSD. For fair comparison, we train a model with only 6.20 M parameters based on the EMS-ResNet10 structure while reducing the number of channels. The results in Table 2 show that our method can achieve the mAP@0.5:0.95 of 0.267 at the same number of time steps, which far outperforms their performance of 0.189. It is remarkable that our full-spike model is more sparse, with a spike firing rate of 21.15%, which implies lower energy consumption. We also conducted comparison experiments with an ANN-ResNet10 [14] of the same structure with reduced number of channels and discovered we even achieved better performance than the ANN network, which indicates that the SNN network may be more competitive in processing event data.
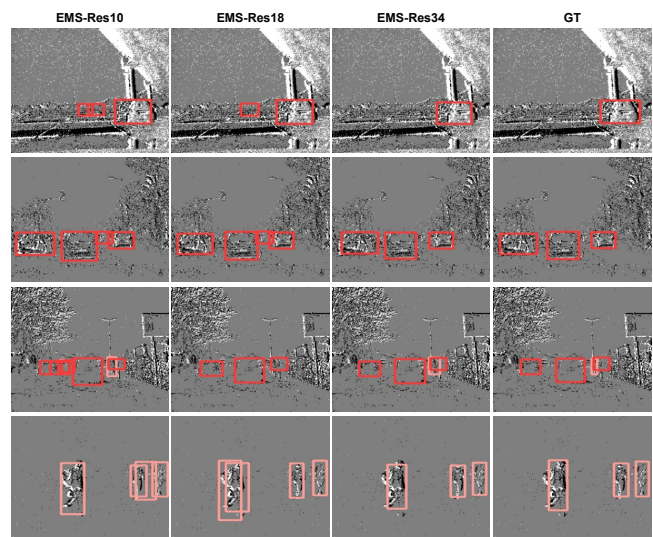

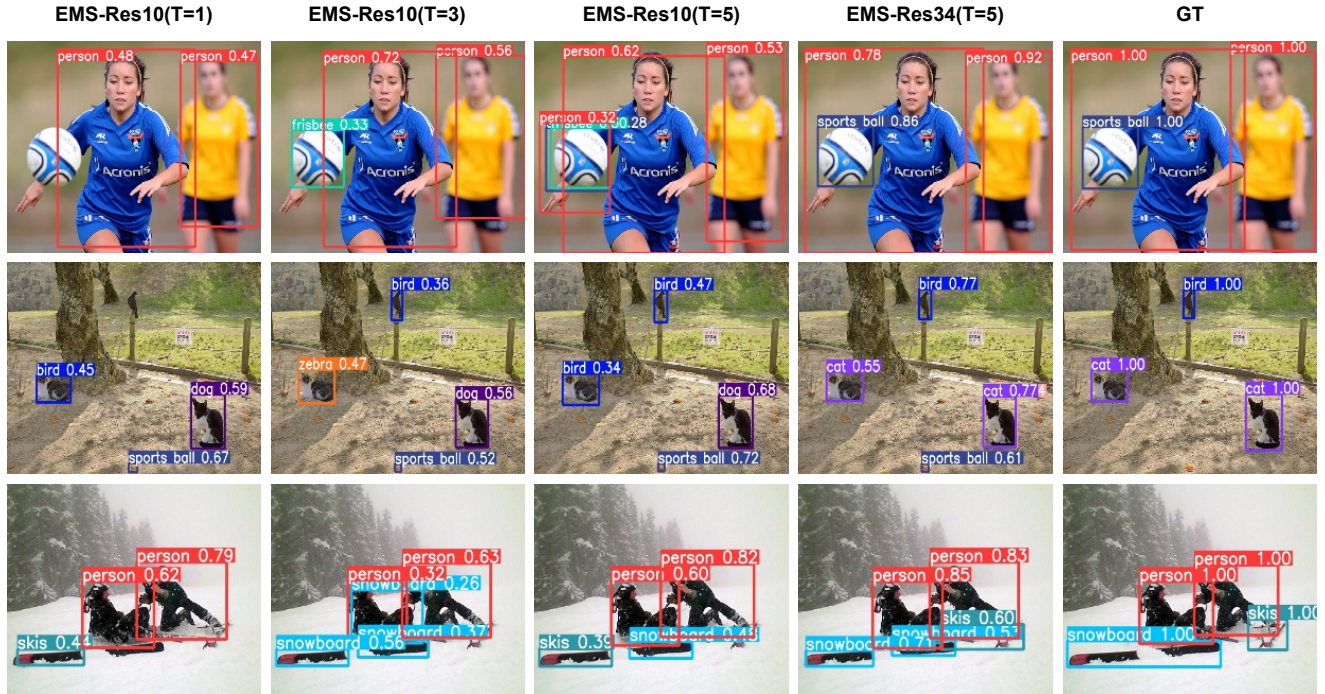Figure 3. **Object detection results on the Gen1 Dataset.**

Figure 4. **Object detection results on the COCO Dataset**. The first three columns compare the effect of time steps on performance for the same network structure. The third and fourth columns compare the effect of the depth of the network on performance.

5.3. Ablation Experiments

We conducted ablation experiments to better understand the effectiveness of different residual modules on reducing energy consumption, the effect of network depth and time step on model performance. All ablation experiments are performed by following the experimental setup in Sec 5.1, if not otherwise specified.

Model	mAP @0.5	mAP @0.5:0.95	Params	Firing Rate	Energy Efficiency
ANN-Res18	0.537	0.290	9.56M	/	$1\times$
MS-Res18	0.560	0.285	9.49M	17.08%	$2.43\times$
Sew-Res18	0.561	0.286	9.56M	18.80%	$2.00\times$*
EMS-Res18	**0.565**	0.286	**9.34M**	20.09%	**$4.91\times$**

Table 3. Impact of different residual blocks on Gen1 Dataset. *Non-spike convolution blocks are calculated as MAC operations.

Different Residual Blocks To explore the structural superiority of our EMS-Block, we made comparative experiments based on the Gen1 Dataset (shown in Table 3). All experiments are trained with only 50 epochs, and the batch-size is set to 64. From the experimental results, we found that the performance of our full-spike EMS-ResNet is comparable to those of the other two ResNets with non-spikes and the sparsity of the network is also maintained. The full-spike property of EMS-ResNet enables the network to be energy efficiency. According to Equation 6, we calcu-

lated the energy consumption of the network, excluding the convolutional energy consumption from the first coding layer. The energy consumption on ANN-ResNet18 is around $9.65mJ$, which we denote this baseline energy consumption as $1\times$. Our EMS-Res18 can reduce up to $4.91\times$ of energy consumption.

Numbers of Residual Blocks In Sec 4.4, we theoretically analyze that our EMS-ResNet can achieve deep training. Here we report results in Table 4 on the Gen1 Dataset based on EMS-Res10, EMS-Res18, and EMS-Res34, respectively. When the scale of the network is larger, the feature extraction ability becomes stronger (see Figure 3).

Model	mAP @0.5	mAP @0.5:0.95	Params	Firing Rate
EMS-Res10	0.547	0.267	6.20M	21.15%
EMS-Res18	0.565	0.286	9.34M	20.09%
EMS-Res34	0.590	0.310	14.40M	17.80%

Table 4. Ablation studies of different numbers of residual blocks on Gen1 Dataset.

Size of Time Steps Due to the sparsity of event streams, different event sampling strategies may affect the ablation experiment of time steps on Gen1 Dataset. Thus, we report the performance of EMS-ResNet10 based on the COCO2017 Dataset for T = 1, 3, 5, 7 in Table 5. The time steps can be dynamically adjusted to achieve a balance of effectiveness and efficiency according to the needs

of the actual task. We show some detection results on the test set compared in Figure 4, where it can be found that the accuracy of object detection is higher when the time step is longer. Here, we first train the model based on T=1, and then use the training model with T=1 as a pre-training model with multiple time steps, which can effectively reduce the training time.

T	1	3	5	7
mAP@0.5	0.328	0.362	0.367	0.383
mAP@0.5:0.95	0.162	0.184	0.189	0.199

Table 5. Impact of different size of time steps on COCO Dataset.

6. Conclusion

In this work, we are the first to use the deep directly trained SNN for the object detection task. Considering that object detection involves extracting multi-scale object features, we design a novel energy-efficient full-spike residual block, EMS-ResNet, that eliminates the redundant MAC operations generated by non-spike convolution on shortcut paths and residual connections. The full-spike EMS-ResNet makes it easier to deploy on neuromorphic chips. Moreover, our results show that our model (EMS-YOLO) can achieve comparable performance to that of a same-structure ANN in a very short time step and performs well on both static images and event data. We believe that our model will drive the exploration of SNN for various regression tasks while becoming possible in neuromorphic hardware deployment.

Acknowledgement

References

[1] Larry F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50(5-6):303–304, 1999.

[2] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557, 2015.

[3] Abhishek Balasubramaniam and Sudeep Pasricha. Object detection in autonomous vehicles: Status and open challenges. *arXiv preprint arXiv:2201.07706*, 2022.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015.

[6] Zhaodong Chen, Lei Deng, Bangyan Wang, Guoqi Li, and Yuan Xie. A comprehensive and modularized statistical framework for gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):13–31, 2022.

[7] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2022.

[8] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[9] Pierre de Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi. A large scale event-based detection dataset for automotive. *arXiv preprint arXiv:2001.08499*, 2020.

[10] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the performance comparison between snns and anns. *Neural networks*, 121:294–307, 2020.

[11] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks*, pages 1–8. ieee, 2015.

[12] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[14] Jiaming Gong, Wei Liu, Mengjie Pei, Chengchao Wu, and Liufei Guo. Resnet10: A lightweight residual network for remote sensing image classification. In *2022 14th International Conference on Measuring Technology and Mechatronics Automation*, pages 975–978. IEEE, 2022.

[15] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500, 1952.

[16] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 10–14. IEEE, 2014.

[17] Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks towards deep residual learning. *arXiv preprint arXiv:2112.08954*, 2021.

[18] Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[19] E.M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[20] Olof Johansson. Training of object detection spiking neural networks for event-based vision, 2021.

[21] Jacques Kaiser, Hesham Mostafa, and Emre Neftci. Synaptic plasticity dynamics for deep continuous local learning (decolle). *Frontiers in Neuroscience*, 14:424, 2020.

[22] Seijoon Kim, Seongsik Park, Byunggook Na, Jongwan Kim, and Sungroh Yoon. Towards fast and accurate object detection in bio-inspired spiking neural networks through bayesian optimization. *IEEE Access*, 9:2633–2643, 2020.

[23] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 34, pages 11270–11277, 2020.

[24] Guoqi Li, Lei Deng, Huajing Tang, Gang Pan, Yonghong Tian, Kaushik Roy, and Wolfgang Maass. Brain inspired computing: A systematic survey and future trends. 2023.

[25] Jianing Li, Xiao Wang, Lin Zhu, Jia Li, Tiejun Huang, and Yonghong Tian. Retinomorphic object detection in asynchronous visual streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1332–1340, 2022.

[26] Yang Li, Xiang He, Yiting Dong, Qingqun Kong, and Yi Zeng. Spike calibration: Fast and accurate conversion of spiking neural network for object detection and segmentation. *arXiv preprint arXiv:2207.02702*, 2022.

[27] Hong-Han Lien and Tian-Sheuan Chang. Sparse compressed spiking neural network accelerator for object detection. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(5):2060–2069, 2022.

[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 on Computer Vision*, pages 740–755. Springer, 2014.

[29] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[30] Qian Liu, Ole Richter, Carsten Nielsen, Sadique Sheik, Giacomo Indiveri, and Ning Qiao. Live demonstration: face recognition on an ultra-low power event-driven convolutional neural network asic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[31] 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 on Computer Vision*, pages 21–37. Springer, 2016.

[32] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[33] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[34] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[35] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *arXiv e-prints*, page arXiv:1606.05340, June 2016.

[36] Chi-Sang Poon and Kuan Zhou. Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities. *Frontiers in Neuroscience*, 5:108, 2011.

[37] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. 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, 2010.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

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

[40] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[41] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

[42] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

[43] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 33, pages 1311–1318, 2019.

[44] Shuiying Xiang, Tao Zhang, Shuqing Jiang, Yanan Han, Yahui Zhang, Chenyang Du, Xingxing Guo, Licun Yu, Yuechun Shi, and Yue Hao. Spiking siamfc++: Deep spiking neural network for object tracking. *arXiv preprint arXiv:2209.12010*, 2022.

[45] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230, 2021.

[46] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9393–9410, 2023.

[47] Chen Zhang and Joohee Kim. Object detection with location-aware deformable convolution and backward attention filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9452–9461, 2019.

[48] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8801–8810, 2022.

[49] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11062–11070, 2021.

[50] Lin Zhu, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian. Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3594–3604, 2022.

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.