The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Finding Visual Saliency in Continuous Spike Stream

Lin Zhu[1], **Xianzhang Chen**[1], **Xiao Wang**[3], **Hua Huang**[2,1,*]

[1] School of Computer Science and Technology, Beijing Institute of Technology, China
[2] School of Artificial Intelligence, Beijing Normal University, China
[3] School of Computer Science and Technology, Anhui University, China
{linzhu,xianzhangchen}@bit.edu.cn, wangxiaocvpr@foxmail.com, huahuang@bnu.edu.cn

Abstract

As a bio-inspired vision sensor, the spike camera emulates the operational principles of the fovea, a compact retinal region, by employing spike discharges to encode the accumulation of per-pixel luminance intensity. Leveraging its high temporal resolution and bio-inspired neuromorphic design, the spike camera holds significant promise for advancing computer vision applications. Saliency detection mimics the behavior of human beings and captures the most salient region from the scenes. In this paper, we investigate the visual saliency in the continuous spike stream for the first time. To effectively process the binary spike stream, we propose a Recurrent Spiking Transformer (RST) framework, which is based on a full spiking neural network. Our framework enables the extraction of spatio-temporal features from the continuous spatio-temporal spike stream while maintaining low power consumption. To facilitate the training and validation of our proposed model, we build a comprehensive real-world spike-based visual saliency dataset, enriched with numerous light conditions. Extensive experiments demonstrate the superior performance of our Recurrent Spiking Transformer framework in comparison to other spike neural network-based methods. Our framework exhibits a substantial margin of improvement in capturing and highlighting visual saliency in the spike stream, which not only provides a new perspective for spike-based saliency segmentation but also shows a new paradigm for full SNN-based transformer models. The code and dataset are available at https://github.com/BIT-Vision/SVS.

Introduction

The human visual system (HVS) possesses an extraordinary ability to swiftly identify and focus on visually distinct and prominent objects or regions within images or scenes (Borji, Sihite, and Itti 2012). This remarkable process has inspired advancements in computer vision, particularly in saliency detection, which aims to identify objects or areas of significance carrying valuable information in images or videos (Wu et al. 2019; Fan et al. 2019). As a burgeoning field, saliency detection has attracted the attention of researchers across various disciplines. Central to this pursuit is the detection of salient objects, a process often referred to as saliency detection or salient-object detection. This involves locating and isolating objects from their backgrounds, leading to the development of numerous models that excel in traditional image modalities (Wang et al. 2021).

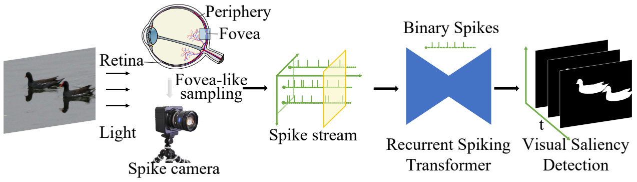

Figure 1: The motivation of detecting visual saliency in continuous spike stream. In contrast to ANNs, SNNs provide a biologically realistic model where neurons communicate through discrete spikes, making them well-suited for processing spike data with low power consumption.

However, the sensing mechanism of human vision (Sinha et al. 2017) diverges from the standard digital camera paradigm. Human vision lacks the concept of frames or discrete pictures, and its mechanism is considerably intricate. Nonetheless, cues and inspiration can be drawn from the structure and signal processing within the human retina. Researchers have designed spiking image sensors that mimic the behavior of integrate-and-fire neurons, operating asynchronously (Culurciello, Etienne-Cummings, and Boahen 2003; Shoushun and Bermak 2007; Zhu et al. 2019; Dong, Huang, and Tian 2017; Zhu et al. 2020). These sensors, in contrast to conventional cameras with fixed integration times, enable each pixel to determine its optimal integration time. Consequently, these spiking image sensors facilitate the reconstruction of visual textures without adhering to the constraints of frames. A recent advancement in this domain is the spike camera (Dong, Huang, and Tian 2017; Zhu et al. 2019), which adopts a fovea-like sampling method (FSM) and mirrors the structure and functionality of the primate fovea. Unlike dynamic vision sensors based on temporal contrast sampling (Lichtsteiner, Posch, and Delbruck 2008), the spike camera incorporates spatial (250×400) and temporal (20,000 Hz) resolution, merging visual reconstruction and motion sensitivity to effectively handle high-speed vision tasks. In this paper, we delve into the field of visual saliency within continuous spike streams. Contrary to tradi-

[*] Corresponding author: Hua Huang.

tional image modalities, visual saliency is encoded within binary spike streams in the spatio-temporal domain. Given the 20,000 Hz sampling rate of the spike camera, effectively processing the continuous spike stream presents a challenge. This leads us to a key question: "*How can visual saliency be detected from a continuous spike stream while minimizing power consumption?*" The potential lies in synergizing continuous spike streams with low-power spiking neural networks (SNNs). Compared to artificial neural networks (ANNs), SNNs offer a more biologically realistic model, with neurons communicating via discrete spikes rather than continuous activation. However, existing SNN researches have predominantly centered on tasks such as classification, optical estimation, motion segmentation, and angular velocity regression, often utilizing traditional or event cameras (Fang et al. 2021; Lee et al. 2020; Zhu et al. 2022a,b).

To the best of our knowledge, this work pioneers the exploration of visual saliency within continuous spike streams captured by the spike camera. The motivation is shown in Fig. 1. To effectively process binary spike streams, we present the Recurrent Spiking Transformer (RST) framework, a full spiking neural network architecture. Our framework comprises spike-based spatio-temporal feature extraction, recurrent feature aggregation, multi-scale refinement, and multi-step loss. To facilitate model training and validation, we have constructed an extensive real-world spike-based visual saliency dataset, enriched with diverse lighting conditions. Our contribution can be summarized as follows:

- We investigate visual saliency within continuous spike streams captured by the spike camera for the first time. To effectively process the binary spike stream, we propose a Recurrent Spiking Transformer (RST) framework, which is based on a full spiking neural network.
- We propose a recurrent feature aggregation structure to enhance the temporal property of the spiking transformer. Moreover, a multi-step loss is designed for better utilizing the temporal information of the spike stream.
- We build a novel dataset consisting of spike streams and per-object masks. Extensive experimental results on our real-world datasets demonstrate the effectiveness of our network. Our dataset will be available to the research community for further investigation and exploration.

Related Work

Visual Saliency in Traditional Image Salient object detection is an active research field in computer vision, which plays an important role in object segmentation and detection tasks. Depending on the different detection targets, this field can be divided into various sub-tasks. Traditional RGB (Wu et al. 2019; Chen et al. 2020) and RGB-D (Ji et al. 2021; Fu et al. 2021) methods aim to find salient objects from complex scenes through the color and depth information. CoSOD (Zhang et al. 2021a; Su et al. 2023) is used to detect the co-saliency objects between a group of images. VSOD (Fan et al. 2019; Yan et al. 2019; Zhang et al. 2021b; Dosovitskiy et al. 2020a) pay more attention to using the spatio-temporal feature which is helpful for detecting salient objects in continuous images.

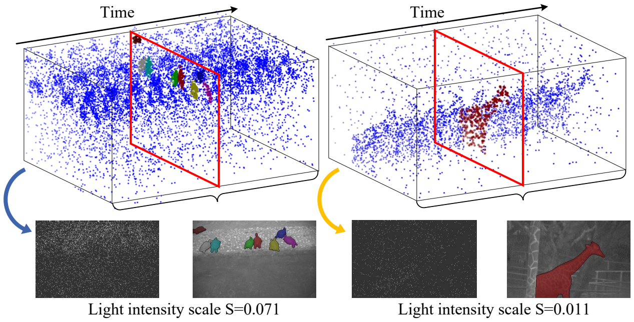

Figure 2: Visual saliency in spatio-temporal spike stream.

Neuromorphic Camera Applications Neuromorphic camera, such as Event camera (Serrano-Gotarredona and Linares-Barranco 2013; Brandli et al. 2014) and Spike camera (Dong, Huang, and Tian 2017; Zhu et al. 2019), which captures the change or accumulation of light intensity, has been widely used in computer vision applications (Xiang et al. 2021; Wang et al. 2022; Dong et al. 2019; Gu et al. 2023). For example, E2vid (Rebecq et al. 2019) applies ConvLSTM (Shi et al. 2015) to extract spatio-temporal features from event streams for video reconstruction. EV-IMO (Mitrokhin et al. 2019) use the continuous property of events to solve motion segmentation task. Spike2Flow (Zhao et al. 2022) and SCFlow (Hu et al. 2022) use spike data to generate optical flow to deal with different speed scenes. RSIR (Zhu et al. 2023) is designed to reduce the noise under general illumination for spike-based image reconstruction.

Spiking Neural Network for Vision Task Based on the capability of simulating neuron dynamics, spiking neural networks have been used for many vision tasks. Spiking-YOLO (Kim et al. 2020) trains an ANN model and uses multi-step to accumulate spikes for imitating ANN features, which is widely used for SNN training. SEW-ResNet (Fang et al. 2021) and Spikformer (Zhou et al. 2022) directly train SNN models for image classification. Spike-Flownet (Lee et al. 2020) and XLIF-FireNet(Hagenaars, Paredes-Vallés, and De Croon 2021) apply SNN to the optical flow estimation task. Spiking Deeplab (Kim, Chough, and Panda 2022) is designed to generate a dense prediction for semantic segmentation. EVSNN (Zhu et al. 2022a) uses the temporal information of neuron membrane potential to reconstruct continuous event-based video frames.

Inspired by the temporal property and the low energy consumption of the spiking neural networks, we build a recurrent spiking transformer architecture to extract spatio-temporal spiking-based features, which facilitates the detection of visual saliency in a continuous spike stream.

Visual Saliency in Continuous Spike Stream

In this section, we first analyze the sampling principle of spike cameras. Based on the characteristics of spike data, we further analyze the visual saliency in spike data and construct a spike-based visual saliency (SVS) dataset.

Spike Sampling Mechanism In a spike camera, the photoreceptor converts the intensity of light into voltage (Dong,

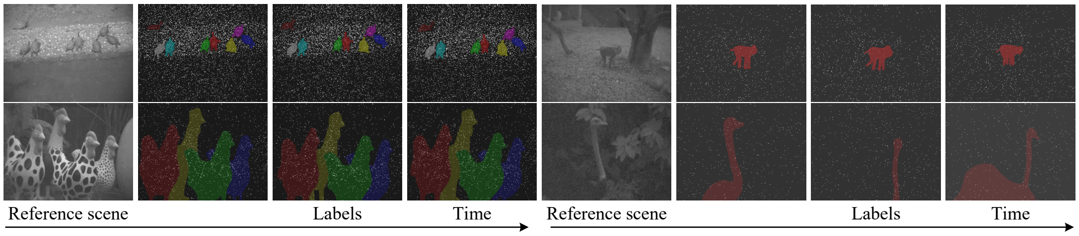

Figure 3: Samples in our spike-based visual saliency (SVS) dataset.

Huang, and Tian 2017; Zhu et al. 2019). When the voltage I surpasses a predetermined threshold ϕ, a one-bit spike is generated, simultaneously triggering a signal to reset the integrator $\int I\mathrm{d}t \geq \phi$. This process is quite similar to the integrate-and-fire neuron. Distinct luminance stimuli denoted as I result in varying spike firing rates, where the initiation of output and reset operations occurs asynchronously across multiple pixels. As a general trend, greater light intensity corresponds to higher firing speeds. The raw data captured by the spike camera takes the form of a three-dimensional spike array denoted as D. The spike camera's primary focus lies in integrating luminance intensity and emitting spikes at an exceptionally high frequency (20,000 Hz). During each sampling timestep, when a spike has just been discharged, a digital signal of "1" (indicating a spike) is produced; otherwise, a signal "0" is generated.

Spatio-temporal Analysis on Spike Visual Saliency Saliency object detection (SOD) is a task that segments the regions or objects of greatest interest in human vision from the scene. Spike cameras record the scenes through accumulating intensity and generate sparse spike data, the spike visual saliency is closer to the biological principles of human eyes. In a spike camera, when the firing threshold ϕ is reached, the integrator is reset and triggers a spike emission. The time it takes for the integrator to fill from empty to capacity is not fixed due to fluctuations in light conditions. At a microscopic level, the act of firing a spike corresponds to the recording of a consistent number of photons. Different from the conventional SOD utilizing standard cameras, the visual saliency within the continuous spike stream is hidden within the binary spikes in the spatio-temporal domain. As depicted in Fig. 2, given the binary nature of the spike stream, extracting saliency regions at specific time points necessitates simultaneous consideration of spatial and temporal factors.

Spike-based Visual Saliency Dataset The datasets play an important role for the development of new algorithms and models. In our paper, we construct the first spike-based visual saliency (SVS) dataset. We use a spike camera (Spatial resolution of 250×400 and a temporal resolution of 20,000 Hz.) to collect real-world spike data, which includes different light intensity scenes. We use the average Light Intensity Scale (LIS) to split the high and low-intensity scenes, the LIS is defined as:

$$\mathbf{LIS} = \mathbf{M}/(\mathbf{H} \times \mathbf{W}) \tag{1}$$

where $\mathbf{M}$ is the number of the spike in a frame, $\mathbf{H}$ and $\mathbf{W}$ is the camera size, the details of dataset are listed in Table 1.

	Train		Val.		Total
	high	low	high	low	
Seq. num.	24	76	8	22	130
Spikes num.	8.7B	10.3B	2.9B	3.1B	25B
Mean spikes	0.36B	0.13B	0.37B	0.14B	0.19B
Mean LIS	0.045	0.017	0.046	0.018	0.031
Mean objs.	2.6	1.8	2.9	2.2	2.1
Mean size (Pixels)	7891	7163	4667	8964	7449

Table 1: The statistics of SVS dataset.

Our dataset comprises 130 spike sequences, each of which is divided into 200 subsequences. Initially, we employ a spike-based reconstruction method (Zhu et al. 2019) to reconstruct textures, and annotate salient objects with instance labels on them. To facilitate training and evaluation, we partition the dataset into a training set and a validation set. The training set encompasses 100 sequences, encompassing 20,000 annotated frames, while the validation set consists of 30 sequences with 6,000 annotated frames. The annotated frames have a time interval of 20 ms, which corresponds to 400 spike frames. For visual reference, example annotations from our dataset can be seen in Fig. 3. Within our dataset, we offer spike frames, reference scenes, and object masks, all of which are accessible to the research community.

Learning to Detect Visual Saliency via Spiking Neural Network

Preliminary: Spiking Neural Network

1) Spiking Neuron. Different from traditional ANN models use a weighted sum of inputs to generate continuous values, SNN models transmit discrete spikes by combining the weighted sum of inputs and the membrane potential of the spiking neuron. If the membrane potential reaches a threshold $\mathbf{V}_{\mathrm{th}}$, the neuron will emit spike $\mathbf{S}_{\mathrm{t}} \in \{0, 1\}$ through a Heaviside step function $\Theta(\cdot)$ to its subsequent neuron. In this paper, we use Leaky Integrate-and-Fire (LIF) model (Gerstner et al. 2014) which is a widely used neuron model in SNN as our basic computing unit, and the dynamics equations of LIF neuron are described as:

$$\mathbf{H}_{\mathrm{t}} = \mathbf{V}_{\mathrm{t}-1} + \frac{1}{\tau} \cdot (\mathbf{X}_{\mathrm{t}} - (\mathbf{V}_{\mathrm{t}-1} - \mathbf{V}_{\mathrm{reset}})), \tag{2}$$

$$\mathbf{S}_{\mathrm{t}} = \Theta(\mathbf{H}_{\mathrm{t}} - \mathbf{V}_{\mathrm{th}}), \tag{3}$$

$$\mathbf{V}_{\mathrm{t}} = \mathbf{H}_{\mathrm{t}} \cdot (1 - \mathbf{S}_{\mathrm{t}}) + \mathbf{V}_{\mathrm{reset}} \cdot \mathbf{S}_{\mathrm{t}}, \tag{4}$$

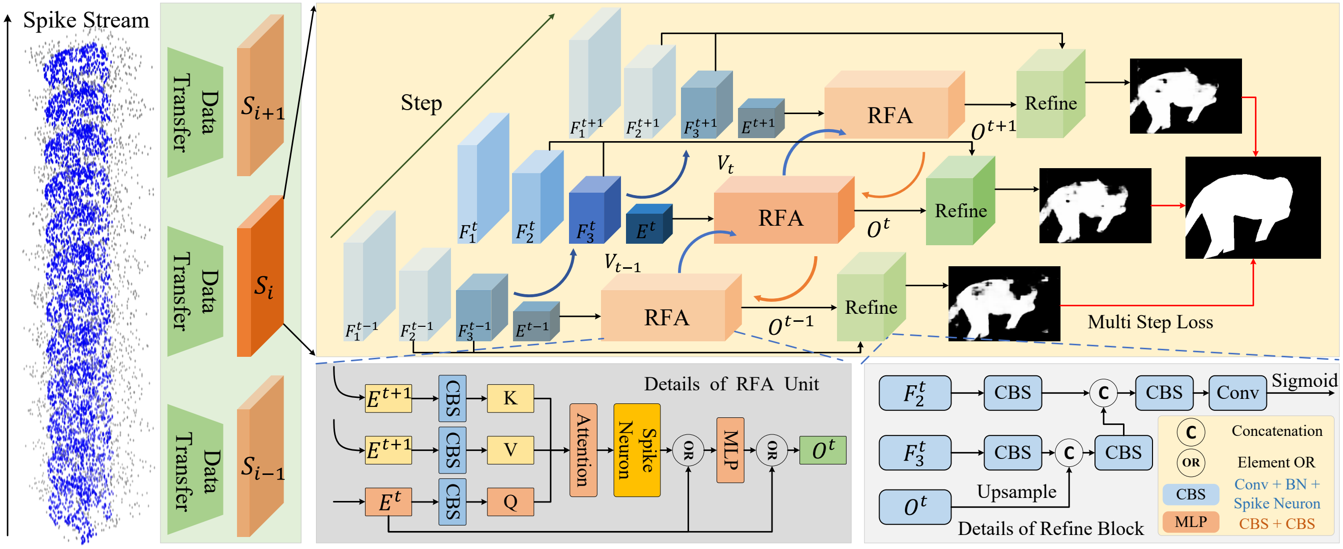

Figure 4: The framework of our Recurrent Spiking Transformer (RST). Our recurrent spiking Transformer is a full spiking neural network architecture, which comprises spike-based spatio-temporal feature extraction, recurrent feature aggregation, multi-scale refinement, and multi-step loss.

where $\mathbf{X}_t$ denotes the input to neuron at time t, τ is the membrane time constant, $\mathbf{H}_t$ is the membrane potential after neuronal dynamics at t and $\mathbf{V}_t$ represents the membrane potential after emitting spike.

2) Spiking Transformer Block. Spikformer (Zhou et al. 2022) introduces the Spiking Self Attention (SSA) in SNN and applies it to the classification task. SSA replaces the nonlinearity function by LIF neuron for each layer to emit spike sequences. Considering the property of SNN, SSA removes the softmax operation for the attention matrix and uses a scaling factor **s** to constrain the large value of the matrix. Given a feature input $\mathbf{X} \in \mathbb{R}^{B\times N\times C}$, SSA uses learnable matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C\times C}$ and spike neurons $\mathrm{SN}_Q, \mathrm{SN}_K, \mathrm{SN}_V$ to compute the query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$):

$$\begin{aligned}\mathbf{Q} &= \mathrm{SN}_Q(\mathrm{BN}(\mathbf{X}\mathbf{W}_Q)),\\ \mathbf{K} &= \mathrm{SN}_K(\mathrm{BN}(\mathbf{X}\mathbf{W}_K)),\\ \mathbf{V} &= \mathrm{SN}_V(\mathrm{BN}(\mathbf{X}\mathbf{W}_V)),\end{aligned} \tag{5}$$

where $\mathrm{BN}(\cdot)$ is Batch Normalization operation, and $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B\times N\times C}$. Then the SSA can be computed as:

$$\begin{aligned}&\mathbf{SSA}'(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{SN}(\mathbf{Q}\mathbf{K}^{\mathrm{T}}\mathbf{V} \cdot \mathbf{s}),\\ &\mathbf{SSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{SN}(\mathrm{BN}(\mathrm{Linear}(\mathbf{SSA}'(\mathbf{Q}, \mathbf{K}, \mathbf{V})))).\end{aligned} \tag{6}$$

We notice that Spikformer uses the same operations as the traditional Transformer encoder after computing self-attention. The element-add operation is used between each SSA layer and the output of SSA layer $\mathbf{O} \in \mathbb{N}$, which means that $\mathbf{O} \notin \{0, 1\}$ is no longer a binary spike sequence. In order to solve this problem and adapt SSA for our task, we propose a Recurrent Spiking Transformer (RST) to facilitate complete binary spike communication while enhancing the extraction of temporal information.

Temporal Spike Representation. To effectively leverage the temporal information of the spike stream, we employ the inter-spike interval as the spike representation. The intensity is directly correlated with either the spike count or spike frequency. Consequently, by utilizing the inter-spike intervals or straightforwardly tallying the spikes over a specific period, the temporal information in the scene can be comprehensively represented:

$$\mathbf{S} = \mathrm{C}/\Delta \mathrm{t}_{x,y}, \tag{7}$$

where C denotes the maximum grayscale, and $\Delta \mathrm{t}_{x,y}$ means the spike firing interval at pixel (x, y).

Spike-based Spatio-temporal Feature Extraction. Inspired by the temporal property of SNN, we use spike neuron to extract spike-based spatio-temporal feature. The SNN needs recurrent multi-steps to get rich features, so given a temporal spike representation $\mathbf{S} \in \mathbb{R}^{C\times H\times W}$, we first repeat $\mathbf{S}$ for $\mathbf{T}$ steps as $\mathbf{S}' \in \mathbb{R}^{T\times C\times H\times W}$ and use a CBS module (i.e., Conv + BN + Spiking Neuron) to generate multi-scale feature for each step parallelly. The CBS module is consist with a 2D convolution layer (stride 1, kernel size 3), a BatchNorm layer, a LIF neuron and a max pooling layer (stride 2):

$$\mathbf{F} = \mathrm{MP}(\mathrm{SN}(\mathrm{BN}(\mathrm{Conv2d}(\mathbf{S}')))). \tag{8}$$

Similar to traditional SOD model, we use 4-block module to extract feature $\mathbf{F}_i \in \mathbb{R}^{T\times \frac{D}{2^{4-i}}\times \frac{H}{2^i}\times \frac{W}{2^i}}$, where $i \in [1, 4]$ and the $\mathbf{D}$ is the dimension of Recurrent Spiking Transformer (RST). Vanilla Vision Transformer (Dosovitskiy et al. 2020b) usually add position embeddings for image patches, but the spike-based feature has a natural representation for the salient area, so we just use the identity feature and flatten the last feature $\mathbf{F}_4 \in \mathbb{R}^{T\times D\times \frac{H}{16}\times \frac{W}{16}}$ as the input $\mathbf{E} \in \mathbb{R}^{T\times D\times N}$ of our RST module. It not only uses the property of spike-based feature, but also keeps the spiking propagation between each module.

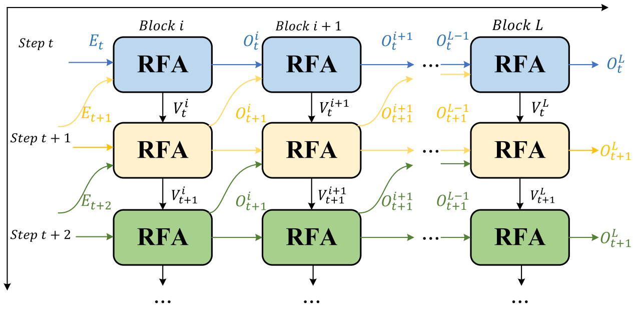

Figure 5: Recurrent mode of our RFA module. RFA uses attention mechanism to aggregate the adjacent step features $\mathrm{E_t}$ and $\mathrm{E_{t+1}}$, which will enhance the feature and generate a better saliency map at the step t.

Recurrent Feature Aggregation (RFA) via Spiking Transformer. The temporal property of SNN is dependent on the accumulation of membrane potential $\mathbf{V}$, only use $\mathbf{V}_{t-1}$ to generate $\mathbf{E}_\mathrm{t}$ may get sparse feature at the early step. Because we parallelly extract features for all steps, the feature $\mathbf{E}$ can be split as $[\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_\mathrm{T}]$ and $\mathbf{E}_\mathrm{t} \in \mathbb{R}^{1\times\mathrm{D}\times\mathrm{N}}$ is the feature at step $\mathbf{t}$, so we use $\mathbf{E}_{\mathrm{t}+1}$ to enhance current feature. At step $\mathbf{t}$, the recurrent spiking transformer block receives $\mathbf{E}_\mathrm{t}$ as Query branch and $\mathbf{E}_{\mathrm{t}+1}$ as Key and Value branch to calculate $\mathbf{Q}_\mathrm{t}, \mathbf{K}_\mathrm{t}, \mathbf{V}_\mathrm{t} \in \mathbb{R}^{1\times\mathrm{D}\times\mathrm{N}}$:

$$\begin{aligned} \mathbf{Q}_\mathrm{t} &= \mathrm{SN_Q}(\mathrm{BN}(\mathbf{E}_\mathrm{t}\mathrm{W_Q})), \\ \mathbf{K}_\mathrm{t} &= \mathrm{SN_K}(\mathrm{BN}(\mathbf{E}_{\mathrm{t}+1}\mathrm{W_K})), \\ \mathbf{V}_\mathrm{t} &= \mathrm{SN_V}(\mathrm{BN}(\mathbf{E}_{\mathrm{t}+1}\mathrm{W_V})). \end{aligned} \tag{9}$$

Then we reshape the features as $\mathbf{Q}'_\mathrm{t}, \mathbf{K}'_\mathrm{t}, \mathbf{V}'_\mathrm{t} \in \mathbb{R}^{1\times\mathrm{n}\times\mathrm{N}\times\frac{\mathrm{D}}{\mathrm{n}}}$ and calculate multi-head attention between adjacent step:

$$\mathbf{AQ}'_\mathrm{t} = \mathrm{SN}(\mathbf{Q}'_\mathrm{t}\mathbf{K}'^{\mathrm{T}}_\mathrm{t}\mathbf{V}'_\mathrm{t} \cdot \mathbf{s}), \tag{10}$$

where $\mathbf{n}$ is the number of multi head attention, $\mathbf{s} = \sqrt{\frac{\mathrm{n}}{\mathrm{D}}}$, and $\mathbf{AQ}'_\mathrm{t}$ will be reshaped as $\mathbf{AQ}_\mathrm{t} \in \mathbb{R}^{1\times\mathrm{D}\times\mathrm{N}}$. Then we use a Linear layer to project the feature and a residual element-or connection to select the salient area:

$$\mathbf{Z}_\mathrm{t} = \mathbf{E}_\mathrm{t} \vee \mathrm{SN}(\mathrm{BN}(\mathrm{Linear}(\mathbf{AQ}_\mathrm{t}))). \tag{11}$$

The feature $\mathbf{Z}_\mathrm{t}$ will be sent to an MLP module which consists of two CBS blocks to get the output $\mathbf{O}_\mathrm{t} \in \mathbb{R}^{1\times\mathrm{D}\times\mathrm{N}}$:

$$\mathbf{O}_\mathrm{t} = \mathbf{E}_\mathrm{t} \vee \mathrm{MLP}(\mathbf{Z}_\mathrm{t}). \tag{12}$$

Finally, we parallel apply this operation for each step feature. As for the final $\mathbf{T}$ step, use its identity feature for each branch. After that, we concat $[\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_\mathrm{T}]$ as $\mathbf{O} \in \mathbb{R}^{\mathrm{T}\times\mathrm{D}\times\mathrm{N}}$, reshape it to $\mathbf{F}' \in \mathbb{R}^{\mathrm{T}\times\mathrm{D}\times\frac{\mathrm{H}}{16}\times\frac{\mathrm{W}}{16}}$ and feed it to Multi-scale Refinement Block.

Multi-scale Refinement. Different from the SNN-based classification task, the saliency map is closely related to the feature spatial size and the semantic information, so it is necessary to have a multi-scale refinement module. In this section, we design a Spiking Multi-scale Refinement block to aggregate the semantic information from different scale features. The refinement block uses CBS block as the basic unit and nearest interpolation to upsample feature. In our model, $\mathbf{F}_2, \mathbf{F}_3, \mathbf{F}'$ are used for feature refinement and upsample, and the output $\mathbf{S} \in \mathbb{R}^{\mathrm{T}\times\mathrm{D}\times\frac{\mathrm{H}}{4}\times\frac{\mathrm{W}}{4}}$ will forward a 1×1 Conv2d layer and a Sigmoid function to generate the saliency map.

Efficient Multi-step Loss. Traditional SNN-based methods for segmentation and classification tasks usually calculate average value for multi-steps as the final result, it may lose information along the time dimension in our task. To better use the relationship among multi-steps, we respectively calculate the loss for every step result, it can also establish constraints for early step's feature. We use binary cross entropy loss $\mathcal{L}_{bce}$ (De Boer et al. 2005), IoU Loss $\mathcal{L}_{iou}$ (Rahman and Wang 2016) and SSIM Loss $\mathcal{L}_{ssim}$ (Wang et al. 2004) to train our model. And for a SNN model with step $\mathbf{T}$, the final loss $\mathcal{L}$ can be expressed as:

$$\mathcal{L} = \sum_{i=1}^{\mathbf{T}} \alpha_i(\mathcal{L}_{bce} + \mathcal{L}_{iou} + \mathcal{L}_{ssim}), \tag{13}$$

where $\alpha_i = (\mathbf{T} - i + 1)/\sum_{i=1}^{\mathbf{T}}$ is the weight of each step, we set $\mathbf{T} = 5$ in our experiment.

Experiment

Experiment Setup

Dataset. We use our spike-based visual saliency (SVS) dataset to test and verify our proposed method. The details are shown in Table 1.

Comparative SNN Models. Since there is no SNN-based salient object detection method, we compare our methods with four mainstream SNN-based architectures. Spiking Deeplab and Spiking FCN (Kim, Chough, and Panda 2022) are methods for semantic segmentation, Spikformer (Zhou et al. 2022) is designed for classification tasks, and the EVSNN (Zhu et al. 2022a) uses SNN for event-based video reconstruction. We modify these methods to adapt spiking-based SOD and train them on SVS dataset.

Training Details. For a fair comparison, we use the same setting for all methods. AdamW is used to train 20 epochs for all models and the initial learning rate is set to 2×10^{-5}, which linearly decays with the epoch until 2×10^{-6}. We use 256×256 as the input size and the time interval of spike data is set to 0.02s, which means the methods will get 400 frames spike at each iteration. We respectively train the model on two settings: single-step and multi-step. When using multi-step mode for training, the same spike data is input at each step and the model iterates five steps, which will result in better performance on a single frame. When using single-step mode, we input continuous spike data in the temporal domain, and the model only iterates once for each input.

Evaluation Metrics. Inspired by traditional SOD tasks, we use Mean absolute error (MAE) (Borji et al. 2015), maximum F-measure F_β^{max}, mean F-measure mF_β (Achanta et al. 2009), and Structure-measure S_m (Fan et al. 2017) as our evaluation metrics, to evaluate the quality between predict saliency map $\mathbf{S}$ and ground-truth label $\mathbf{G}$.

Quantitative Experiment

Table 2 shows the quantitative results for all methods using different steps on our SVS dataset, and our method has the

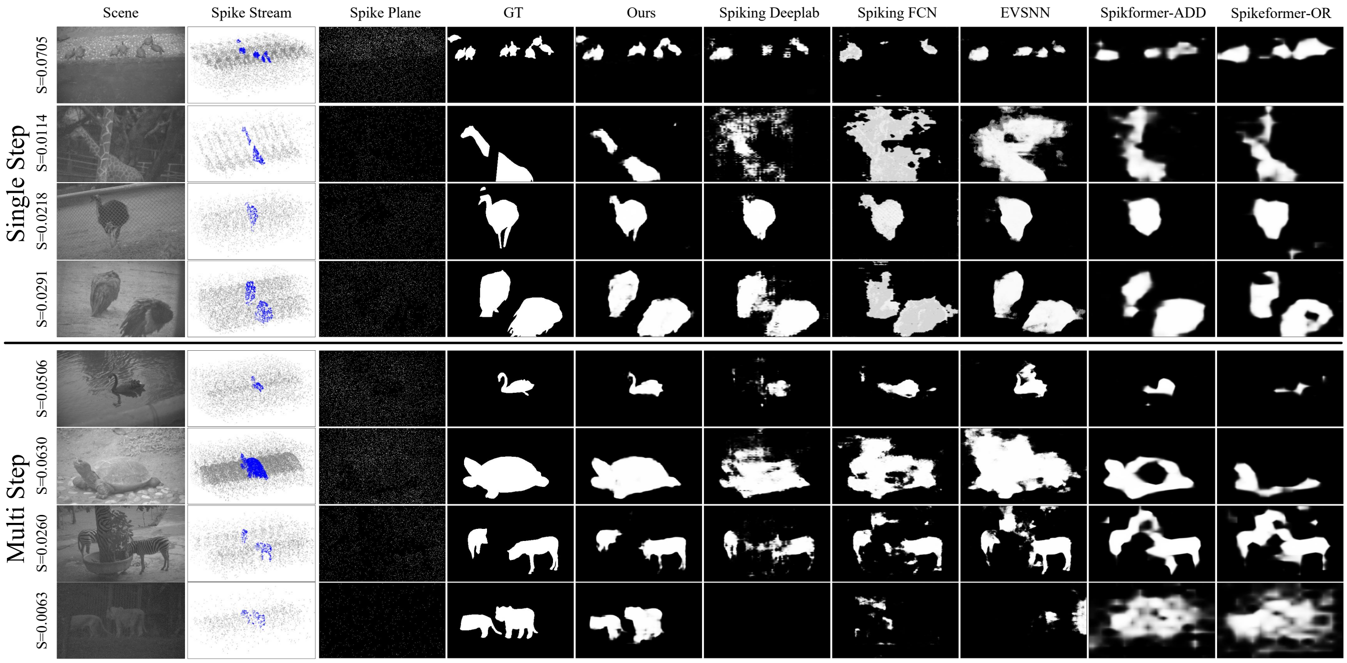

Figure 6: Qualitative results on our SVS dataset. S denotes the light intensity scale of the scene. Spikformer-ADD employs non-spikes in its residual connection, while the remaining methods utilize a full spiking neural network architecture. Our model excels in capturing finer details compared to other SNN-based methods in both single-step and multi-step settings.

Method	Single Step				Multi Step			
	MAE↓	$F_\beta^{max}\uparrow$	$mF_\beta\uparrow$	$S_m\uparrow$	MAE↓	$F_\beta^{max}\uparrow$	$mF_\beta\uparrow$	$S_m\uparrow$
Spiking Deeplab	0.1026	0.5310	0.5151	0.6599	0.0726	0.6175	0.6051	0.7125
Spiking FCN	0.1210	0.4779	0.4370	0.6070	0.0860	0.5970	0.5799	0.6911
EVSNN	0.1059	0.5221	0.4988	0.6583	0.0945	0.6267	0.5850	0.7023
Spikformer-ADD	0.1185	0.4638	0.4415	0.6119	0.0717	0.6890	0.6731	0.7563
Spikformer-OR	0.1389	0.4527	0.4408	0.6068	0.0738	0.6526	0.6323	0.7161
Ours	**0.0784**	**0.6313**	**0.6171**	**0.6970**	**0.0554**	**0.6981**	**0.6882**	**0.7591**

Table 2: Quantitative comparison on our SVS dataset. Spikformer-ADD employs non-spikes in its residual connection, while the remaining methods utilize a full spiking neural network architecture.

best performance on both two settings. Notice that the step setting has a significant influence on all methods, the reason is that spiking neurons need some steps to accumulate the membrane potential. The Spikformer-ADD model has a better result on multi-steps than other comparison methods, this is because Spikformer-ADD uses the element-add operation for residual connection in its SSA module to enhance the feature that transfers floating numbers between each block. If we replace the element-add as the element-or operation, the Spikformer-OR has lower performance. Although our model transfers the whole spike-based features among all modules, our method can predict better than Spikformer-ADD.

Qualitative Experiment

Fig. 6 illustrates the results of various methods in both single and multi-step modes. Notably, when confronted with intricate scenes featuring comparable objects and backgrounds, our method excels in delineating object contours and edges, surpassing other approaches. Furthermore, our method exhibits remarkable robustness across diverse illumination conditions, generating distinct saliency maps for target objects even in challenging low-light scenes, unlike other comparison methods that experience diminished effectiveness in such scenarios.

Performance Analysis in the Temporal Domain

Benefiting our recurrent spiking Transformer module, our model can be easily extended to continuous salient object detection. Unlike other SNN-based methods that necessitate multiple steps for sufficient information extraction, our model achieves high-quality prediction results with just a single inference step. The continuous detection results are depicted in Fig. 7. Remarkably, our model accurately pre-

Recurrent Mode	MAE↓	F_β^{max} ↑	mF_β ↑	S_m ↑
Vanilla	0.0581	0.6811	0.6716	0.7522
Forward	0.0611	0.6696	0.6599	0.7432
Ours (Reverse)	**0.0554**	**0.6981**	**0.6882**	**0.7591**

Table 3: Effect of different recurrent modes.

Method	RFAs	MAE↓	F_β^{max} ↑	mF_β ↑	S_m ↑
w/o Refine	6	0.0701	0.6298	0.6205	0.7190
Refine	0	0.0642	0.6789	0.6622	0.7385
Refine	2	0.0571	0.6942	0.6829	0.7548
Refine	4	0.0559	0.6965	0.6848	0.7563
Refine	8	0.0580	0.6818	0.6716	0.7466
Ours	6	**0.0554**	**0.6981**	**0.6882**	**0.7591**

Table 4: Effect of RST module and refine module.

dicts results even with a 20,000 Hz spike input, where each input corresponds to a single step. This remarkable efficiency leads to minimal energy consumption during continuous spike stream processing. In direct comparison with its ANN-based counterpart, which consumes 167 mJ per inference, our method operates at a mere 5.8 mJ, signifying a substantial reduction in power usage by a factor of 28.7. Further details are available in our supplementary materials.

Ablation Study

Effect of Recurrent Spiking Transformer. In spiking neurons, the membrane potential is useful for extracting spatio-temporal features. Maximizing the utility of these features across all steps promises enhanced model performance compared to the vanilla SNN propagation mode. As shown in Table 3, we test the effect of different recurrent modes. The "Vanilla" means the SNN architecture without additional recurrent structure, "Forward" means using the output of step $t-1$ as the key and value batch of RST module. "Reverse" is our recurrent mode shown in Fig. 5. "Forward" get a worse result than "Vanilla", this can be attributed to sparse information in the early steps, potentially leading to unfavorable effects when directly fusing features. "Reverse" mode operates in parallel, efficiently enhancing features by fusing those from step $t+1$, thus showcasing its effectiveness in bolstering current features.

Effect of RFA and Refine Module. As shown in Table 4, we test the effect of the number of RFA modules and the refine block. Removing the refine block results in a significant performance drop, emphasizing its necessity for robust dense pixel prediction tasks. The influence of RST modules on the final results is evident, yet a noteworthy observation is that performance improvement does not exhibit a linear trend with an increasing number of RST modules. This could be attributed to challenges in effectively training larger models within the limitations of the dataset.

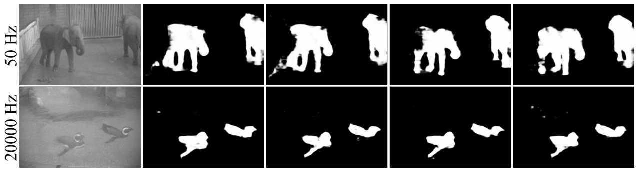

Figure 7: Results from our model during single-step inference using continuous spike data input. The top row illustrates results from 50 Hz spike data input, while the bottom row showcases results from 20,000 Hz spike data input.

Recurrent Mode	Loss	MAE↓	mF_β ↑	S_m ↑
Vanilla	Vanilla	0.0635	0.6690	0.7521
Vanilla	Multi step	0.0581	0.6716	0.7522
Reverse	Vanilla	0.0613	0.6872	**0.7666**
Reverse	Multi step	**0.0554**	**0.6882**	0.7591

Table 5: Effect of the multi-step loss.

Operation	MAE↓	F_β^{max} ↑	mF_β ↑	S_m ↑
ADD	0.0576	0.6803	0.6710	0.7540
Concat	0.0567	0.6933	0.6834	0.7563
Ours (OR)	**0.0554**	**0.6981**	**0.6882**	**0.7591**

Table 6: Effect of the element-wise operation in RST.

Effect of Multi-step Loss. We compare the effect of the vanilla loss and our multi-step loss, the results are shown in Table 5. Our multi-step loss assigns greater weight to early step results, aiding the model in concentrating on sparse features at the beginning. This strategic approach mitigates SNN's reliance on step size to a certain degree, ultimately reducing prediction error rates.

Element-Wise Operation in RST. We test the effect of element-wise operation in our RST module. As shown in Table 6, the "element-and" operation can cause all spikes to 0 on the training stage, so it is difficult to train this model. The "Concat" operation can get similar results to our method, but the computation complexity will increase rapidly as the dimension rises. The "ADD" operation performs worse than others, the reason is that usage of the refine block will convert the features back to spike-based binary features. Considering both the energy consumption and the performance, we use "OR" operation in our model.

Conclusion

In this paper, we explore visual saliency in continuous spike streams using the spike camera. We introduce the Recurrent Spiking Transformer framework, efficiently extracting spatio-temporal features for visual saliency detection while minimizing power consumption. Our constructed spike-based dataset validates the superiority of our RST framework over other SNN-based models, advancing spike-based saliency detection and offering a fresh perspective for SNN-based transformers. This study also innovates transformer models in continuous spike stream analysis.

Acknowledgments

This work is partially supported by grants from the National Natural Science Foundation of China under contract No. 62302041 and China National Postdoctoral Program under contract No. BX20230469.

References

Achanta, R.; Hemami, S.; Estrada, F.; and Susstrunk, S. 2009. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, 1597–1604. IEEE.

Borji, A.; Cheng, M.-M.; Jiang, H.; and Li, J. 2015. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12): 5706–5722.

Borji, A.; Sihite, D. N.; and Itti, L. 2012. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1): 55–69.

Brandli, C.; Berner, R.; Yang, M.; Liu, S.-C.; and Delbruck, T. 2014. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10): 2333–2341.

Chen, Z.; Xu, Q.; Cong, R.; and Huang, Q. 2020. Global context-aware progressive aggregation network for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 10599–10606.

Culurciello, E.; Etienne-Cummings, R.; and Boahen, K. A. 2003. A biomorphic digital image sensor. *IEEE journal of solid-state circuits*, 38(2): 281–294.

De Boer, P.-T.; Kroese, D. P.; Mannor, S.; and Rubinstein, R. Y. 2005. A tutorial on the cross-entropy method. *Annals of operations research*, 134: 19–67.

Dong, S.; Huang, T.; and Tian, Y. 2017. Spike Camera and Its Coding Methods. In *2017 Data Compression Conference (DCC)*.

Dong, S.; Zhu, L.; Xu, D.; Tian, Y.; and Huang, T. 2019. An Efficient Coding Method for Spike Camera Using Inter-Spike Intervals. In *2019 Data Compression Conference (DCC)*, 568–568. IEEE.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020a. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020b. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, D.-P.; Cheng, M.-M.; Liu, Y.; Li, T.; and Borji, A. 2017. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, 4548–4557.

Fan, D.-P.; Wang, W.; Cheng, M.-M.; and Shen, J. 2019. Shifting more attention to video salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8554–8564.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 21056–21069.

Fu, K.; Fan, D.-P.; Ji, G.-P.; Zhao, Q.; Shen, J.; and Zhu, C. 2021. Siamese network for RGB-D salient object detection and beyond. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5541–5559.

Gerstner, W.; Kistler, W. M.; Naud, R.; and Paninski, L. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Gu, D.; Li, J.; Zhu, L.; Zhang, Y.; and Ren, J. S. 2023. Reliable Event Generation with Invertible Conditional Normalizing Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Hagenaars, J.; Paredes-Vallés, F.; and De Croon, G. 2021. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 7167–7179.

Hu, L.; Zhao, R.; Ding, Z.; Ma, L.; Shi, B.; Xiong, R.; and Huang, T. 2022. Optical flow estimation for spiking camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17844–17853.

Ji, W.; Li, J.; Yu, S.; Zhang, M.; Piao, Y.; Yao, S.; Bi, Q.; Ma, K.; Zheng, Y.; Lu, H.; et al. 2021. Calibrated RGB-D salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9471–9481.

Kim, S.; Park, S.; Na, B.; and Yoon, S. 2020. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11270–11277.

Kim, Y.; Chough, J.; and Panda, P. 2022. Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering*, 2(4): 044015.

Lee, C.; Kosta, A. K.; Zhu, A. Z.; Chaney, K.; Daniilidis, K.; and Roy, K. 2020. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *European Conference on Computer Vision*, 366–382. Springer.

Lichtsteiner, P.; Posch, C.; and Delbruck, T. 2008. A 128 $\times$ 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2): 566–576.

Mitrokhin, A.; Ye, C.; Fermüller, C.; Aloimonos, Y.; and Delbruck, T. 2019. EV-IMO: Motion segmentation dataset and learning pipeline for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6105–6112. IEEE.

Rahman, M. A.; and Wang, Y. 2016. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, 234–244. Springer.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6): 1964–1980.

Serrano-Gotarredona, T.; and Linares-Barranco, B. 2013. A 128×128 1.5% Contrast Sensitivity 0.9% FPN 3 μs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers. *IEEE Journal of Solid-State Circuits*, 48(3): 827–838.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Shoushun, C.; and Bermak, A. 2007. Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(3): 346–357.

Sinha, R.; Hoon, M.; Baudin, J.; Okawa, H.; Wong, R. O.; and Rieke, F. 2017. Cellular and circuit mechanisms shaping the perceptual properties of the primate fovea. *Cell*, 168(3): 413–426.

Su, Y.; Deng, J.; Sun, R.; Lin, G.; Su, H.; and Wu, Q. 2023. A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection. *IEEE Transactions on Multimedia*.

Wang, W.; Lai, Q.; Fu, H.; Shen, J.; Ling, H.; and Yang, R. 2021. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 3239–3259.

Wang, X.; Wu, Z.; Jiang, B.; Bao, Z.; Zhu, L.; Li, G.; Wang, Y.; and Tian, Y. 2022. Hardvs: Revisiting human activity recognition with dynamic vision sensors. *arXiv preprint arXiv:2211.09648*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wu, R.; Feng, M.; Guan, W.; Wang, D.; Lu, H.; and Ding, E. 2019. A mutual learning method for salient object detection with intertwined multi-supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8150–8159.

Xiang, X.; Zhu, L.; Li, J.; Wang, Y.; Huang, T.; and Tian, Y. 2021. Learning super-resolution reconstruction for high temporal resolution spike stream. *IEEE Transactions on Circuits and Systems for Video Technology*.

Yan, P.; Li, G.; Xie, Y.; Li, Z.; Wang, C.; Chen, T.; and Lin, L. 2019. Semi-supervised video salient object detection using pseudo-labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7284–7293.

Zhang, K.; Dong, M.; Liu, B.; Yuan, X.-T.; and Liu, Q. 2021a. Deepacg: Co-saliency detection via semantic-aware contrast gromov-wasserstein distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13703–13712.

Zhang, M.; Liu, J.; Wang, Y.; Piao, Y.; Yao, S.; Ji, W.; Li, J.; Lu, H.; and Luo, Z. 2021b. Dynamic context-sensitive filtering network for video salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1553–1563.

Zhao, R.; Xiong, R.; Zhao, J.; Yu, Z.; Fan, X.; and Huang, T. 2022. Learning optical flow from continuous spike streams. *Advances in Neural Information Processing Systems*, 35: 7905–7920.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; and Yuan, L. 2022. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*.

Zhu, L.; Dong, S.; Huang, T.; and Tian, Y. 2019. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 1432–1437. IEEE.

Zhu, L.; Dong, S.; Huang, T.; and Tian, Y. 2020. Hybrid coding of spatiotemporal spike data for a bio-inspired camera. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7): 2837–2851.

Zhu, L.; Dong, S.; Li, J.; Huang, T.; and Tian, Y. 2022a. Ultra-high temporal resolution visual reconstruction from a fovea-like spike camera via spiking neuron model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 1233–1249.

Zhu, L.; Wang, X.; Chang, Y.; Li, J.; Huang, T.; and Tian, Y. 2022b. Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3594–3604.

Zhu, L.; Zheng, Y.; Geng, M.; Wang, L.; and Huang, H. 2023. Recurrent Spike-based Image Restoration under General Illumination. In *Proceedings of the 31st ACM International Conference on Multimedia*, 8251–8260.