Unveiling the Spatial-temporal Effective Receptive Fields of Spiking Neural Networks

Jieyuan Zhang[1], **Xiaolong Zhou**[1], **Shuai Wang**[1] **Wenjie Wei**[1], **Hanwen Liu**[1], **Qian Sun**[1], **Malu Zhang**[1,3]*, **Yang Yang**[1], **Haizhou Li**[2,3]

[1]University of Electronic Science and Technology of China,

[2]The Chinese University of Hong Kong, Shenzhen ,

[3]Shenzhen Loop Area Institute

Abstract

Spiking Neural Networks (SNNs) demonstrate significant potential for energy-efficient neuromorphic computing through an event-driven paradigm. While training methods and computational models have greatly advanced, SNNs struggle to achieve competitive performance in visual long-sequence modeling tasks. In artificial neural networks, the effective receptive field (ERF) serves as a valuable tool for analyzing feature extraction capabilities in visual long-sequence modeling. Inspired by this, we introduce the Spatio-Temporal Effective Receptive Field (ST-ERF) to analyze the ERF distributions across various Transformer-based SNNs. Based on the proposed ST-ERF, we reveal that these models suffer from establishing a robust global ST-ERF, thereby limiting their visual feature modeling capabilities. To overcome this issue, we propose two novel channel-mixer architectures: multi-layer-perceptron-based mixer (MLPixer) and splash-and-reconstruct block (SRB). These architectures enhance global spatial ERF through all timesteps in early network stages of Transformer-based SNNs, improving performance on challenging visual long-sequence modeling tasks. Extensive experiments conducted on the Meta-SDT variants and across object detection and semantic segmentation tasks further validate the effectiveness of our proposed method. Beyond these specific applications, we believe the proposed ST-ERF framework can provide valuable insights for designing and optimizing SNN architectures across a broader range of tasks. The code is available at EricZhang1412/Spatial-temporal-ERF.

1 Introduction

Spiking Neural Networks (SNNs) [1, 2] have emerged as a prominent research focus, characterized by binary spike activation that offers high sparsity, event-driven processing [3, 4], and biological plausibility [5]. Recent advances in encoding schemes [6, 7, 8], training methodologies [9, 10], and neuromorphic hardware [11, 12, 13] have enabled SNNs to achieve remarkable success in diverse tasks, including image processing [14, 15, 16], point/event analysis [17, 18], language understanding [19, 20, 21], and speech processing [22, 23, 24]. Nonetheless, SNNs still struggle to achieve performance comparable to their Artificial Neural Networks (ANNs) counterparts in visual long-sequence modeling tasks.

Compared to conventional image classification, visual long-sequence modeling tasks [25, 26] demand spatially dense outputs with prediction scales several orders of magnitude higher. This paradigm requires architectures capable of modeling long-range spatial dependencies, which are essential for

*Corresponding author: maluzhang@uestc.edu.cn

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

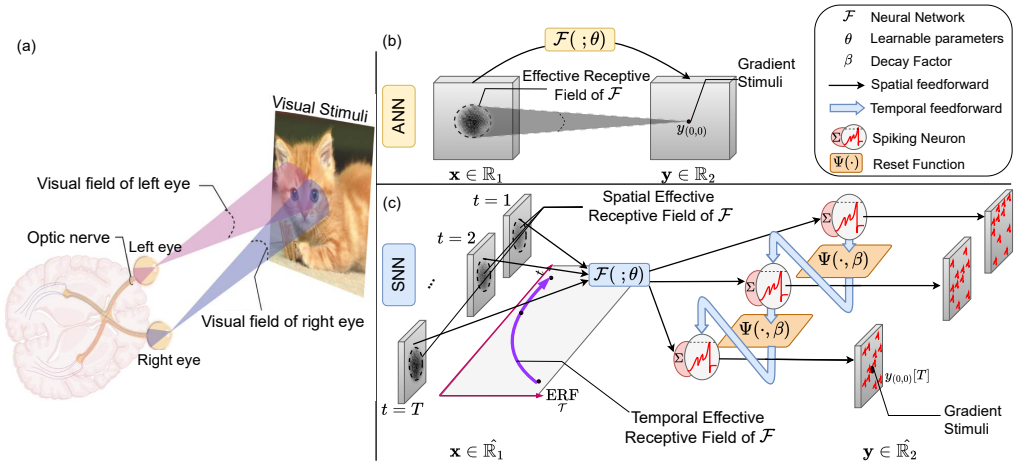

Figure 1: (a) The human visual field. (b) ERF in ANNs. (c) ST-ERF in SNNs. It extends the ERF to the temporal dimension, thus facilitating a comprehensive analysis of feature extraction in SNNs.

achieving competitive performance [25]. The Transformer [27] architecture introduces a self-attention mechanism that enables effective modeling of long-range spatial dependencies [28, 29, 30]. Motivated by this, recent studies have proposed various Transformer-based SNNs [31, 32, 33, 34], achieving notable progress in visual long-sequence tasks [15]. However, simply combining Transformers with SNNs may lead to suboptimal designs without fully considering the intrinsic spatio-temporal dynamics of spiking neurons. To bridge this gap, a more structured and interpretable framework is required to examine how SNNs model spatio-temporal dependencies. In this context, receptive field (RF) analysis offers a concrete lens through which their feature extraction capacity and attention allocation can be theoretically characterized.

In neuroscience, the RF represents the region of sensory input that can modulate a neuron's activity [35]. Borrowing this concept, the deep learning community defines a neuron's RF as the region of the input that can influence its output [36], with its size determined by network topology. However, this topology-based definition treats all positions within the RF equally, ignoring the learnable weights that determine the actual contribution of each input location. To refine this problem, researchers proposed the effective receptive field (ERF) [37] to quantify input features' contributions to output features via gradient analysis. Unlike topology-based RF, gradient-based ERF provides a more faithful characterization of the network's feature extraction patterns. However, such framework cannot be directly applied to SNNs due to the intrinsic spatio-temporal dynamics of spiking neurons. Therefore, we introduce the Spatial-Temporal Effective Receptive Field (ST-ERF) framework, which quantifies input feature contributions across spatio-temporal locations to characterize SNNs' feature extraction patterns. By jointly modeling temporal dependencies and spatial relationships, ST-ERF facilitates a comprehensive analysis of information processing in SNNs.

Based on the proposed ST-ERF, we analyze various Transformer-based SNNs and identify that existing models fail to establish effective global receptive fields across all timesteps. This limitation stems from the prevalent use of convolutional channel-mixers, which inherently introduce locality bias [38]. Despite facilitating efficient local feature extraction and long-range sparse modeling, this architectural design fundamentally constrains the long-range dense spatial interactions necessary for effective visual long-sequence modeling in SNNs. Building on these insights, we propose two novel channel-mixer architectures: multi-layer-perceptron-based mixer (MLPixer) and splash-and-reconstruct block (SRB). These designs use pixel-wise MLPs to keep spatial features separate when mixing channels, which reduces locality bias and improves the global receptive field in early stages of Transformer-based SNNs. Extensive experiments demonstrate the effectiveness of our methods on visual long-sequence tasks. Specially, on COCO 2017 object detection and ADE20K semantic segmentation, our Meta-SDT-Base [33] with SRB achieves 48.9% $\mathrm{AP}^{\mathrm{b}}_{50}$ and 43.7% mIoU, respectively, while maintaining a smaller model size. These results surpass state-of-the-art Transformer-based SNNs, thereby validating our ST-ERF analysis and further advancing SNNs in visual long-sequence modeling. Our main contributions are listed as follows:

- We propose the ST-ERF framework, extending the traditional ERF concept to the temporal dimension with rigorous mathematical formalization. ST-ERF systematically quantifies how input features at different spatial and temporal locations contribute to output features. This provides a theoretical tool for understanding and optimizing feature extraction in SNNs.

- We analyze various Transformer-based SNNs using the ST-ERF framework, revealing a critical limitation that existing models fail to establish a global ERF across all timesteps. To overcome this issue, we introduce two novel channel mixer designs: MLPixer and SRB, enabling Transformer-based SNNs to fully exploit their global modeling potential of long-range dependencies.

- We conduct extensive experiments on visual long-sequence modeling tasks and demonstrate that our method achieves superior performance. For instance, our Meta-SDT-Base with SRB achieves 48.9% $\mathrm{AP}^{\mathrm{b}}_{50}$ on COCO 2017 detection and 43.7% mIoU on ADE20K segmentation. It significantly outperforms existing state-of-the-art Transformer-based SNNs while using a smaller model size. These results strongly support the validity of our ST-ERF theoretical analysis and demonstrate the effectiveness of the proposed architectural designs.

2 Related Works

2.1 Receptive Field in Neural Networks

The human visual system perceives the external world through the visual fields of both eyes. As illustrated in Figure 1(a), each eye covers a specific region of the visual space, and these regions partially overlap in the center to enable binocular vision. Neurons along the visual pathway respond selectively to stimuli within their RFs. Over the past decades, the RF theory has profoundly influenced our understanding of how the brain filters and integrates visual information across spatial locations [39]. Inspired by the RF theory, deep neural networks adopt a similar principle by characterizing hierarchical ERFs that capture progressively abstract representations of input data. As shown in Figure 1(b), the ERF [37] formalizes this process by analyzing how spatial stimuli contribute to network activations. ERF has motivated extensive research at different architectural levels, from understanding basic operators [40] to designing higher-level modules and network structures such as Adaptive Receptive Fields [41], RF-Next [42], and AutoRF [43]. RF-based analysis has also driven advances in computational efficiency for lightweight architectures, influencing the development of CNN-based MobileNet variants [44, 45, 46] and MLP-based networks such as MLP-Mixer [47] and TSMixer [48]. Building on these insights, this work extends the ERF concept to SNNs, offering a theoretical framework for analyzing and optimizing their spatio-temporal feature extraction processes.

2.2 Visual Long-sequence Modeling in SNNs

Visual long-sequence modeling refers to tasks that require multiple predictions per image, rather than a single-label classification [49]. These tasks mainly include detection, segmentation, video understanding, and so on [25]. As these tasks involve modeling complex spatial and temporal dependencies, they demand architectures capable of capturing long-range contextual information. Transformer has become the dominant paradigm for visual long-sequence modeling owing to its global self-attention mechanism and flexible scalability. However, such models still suffer from high computational costs, primarily due to the quadratic complexity of self-attention, dense prediction requirements, and high-resolution inputs [50]. Recently, leveraging the sparse spike-driven nature of SNNs has emerged as a promising direction to mitigate these computational costs. Spike-driven Transformer series [51, 33, 52] adapt the standard Metaformer into an SNN framework for object detection and semantic segmentation, demonstrating the feasibility of SNNs in dense prediction tasks. Spike2Former [15] integrates normalized integer leaky-and-integrated firing (NI-LIF) neurons and spike-driven deformable attention to achieve competitive performance on segmentation benchmarks while maintaining low energy consumption. Despite these advancements, SNNs still lag behind ANNs in visual long-sequence modeling. This underscores the need for deeper investigation into SNNs' spatio-temporal bottlenecks and architectural optimization.

3 Theoretical Analysis of Spatio-temporal Effective Receptive Field

In this section, we first introduce the concept of ERF in conventional ANNs. Subsequently, we extend this conventional ERF into the temporal dimension to characterize the ST-ERF in SNNs. Finally, we introduce a loss-derived method to efficiently compute ST-ERF in SNNs.

3.1 ERF in ANNs

The concept of the ERF has been widely adopted to analyze how input features contribute to network activations and how such influences are distributed within the RF [40, 41]. Under the assumption of a single channel per layer, Luo et al. [37] mathematically characterized how each input feature contributes to the output of a neural network layer. It can be defined as follows:

$$\mathrm{ERF}_{(i,j)}[y_{(m,n)};\mathbf{x}] = \frac{\partial y_{(m,n)}}{\partial x_{(i,j)}}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}_1$ is the input feature and $\mathbf{y} \in \mathbb{R}_2$ is the output feature. In this manner, the ERF measures the partial derivative of an output feature $y_{(m,n)} \in \mathbf{y}$ with respect to each input feature $x_{(i,j)} \in \mathbf{x}$ within a given layer. As illustrated in Figure 1(b), the ERF of a given network $\mathcal{F}(;\theta)$ describes the input regions that contribute to a particular output activation.

As shown in Eq. (1), the ERF can be computed at any output location. However, most studies evaluate the ERF at the central output feature $y_{(0,0)}$ by assigning a unit gradient to this location [53, 54]. This practice establishes a centered and symmetric reference, ensuring stable and comparable visualization results. In this work, we also follow the setting of [37] and adopt the ERF at the central output feature $y_{(0,0)}$ as the evaluation metric.

3.2 ST-ERF in SNNs

Due to the inherent temporal dynamics, SNNs require additional consideration of the input at each timestep. To address this, we formally define the ST-ERF (i.e., $\mathrm{ERF}^{(\mathcal{S},\mathcal{T})}$). Firstly, we redefine the mapping relationship of SNNs. Consider a SNN layer with learnable parameters θ that maps input spike features $\mathbf{x}[1:T] \in \hat{\mathbb{R}_1}$ to output spike features $\mathbf{y}[1:T] \in \hat{\mathbb{R}_2}$:

$$\mathbf{y}[1:T] = \mathcal{F}(\mathbf{x}[1:T];\theta), \mathcal{F}: \hat{\mathbb{R}_1} \to \hat{\mathbb{R}_2}. \tag{2}$$

Its ERF needs to account not only for the accumulation across spatial dimensions but also for that across temporal dimensions. Specifically, $\mathrm{ERF}^{(\mathcal{S},\mathcal{T})} \in \hat{\mathbb{R}_1}$ can be expressed as:

$$\mathrm{ERF}^{(\mathcal{S},\mathcal{T})}_{(i,j)}[\,y_{(m,n)}[t],\tau;\mathbf{x}\,] = \frac{\partial y_{(m,n)}[t]}{\partial x_{(i,j)}[t-\tau]}, 1 \le t \le T, 0 \le \tau \le t-1. \tag{3}$$

Accordingly, $\mathrm{ERF}^{(\mathcal{S},\mathcal{T})}$ quantifies how much each input feature $x_{(i,j)}[t-\tau] \in \mathbf{x}$ at a previous timestep $t-\tau$ contributes to a specific output feature $y_{(m,n)}[t] \in \mathbf{y}$. Based on this definition, the spatial ERF (i.e., $\mathrm{ERF}^{(\mathcal{S})}$) can be seen as the weighted average of the ST-ERFs over all timesteps:

$$\mathrm{ERF}^{(\mathcal{S})}_{(i,j)}[\,y_{(m,n)};\mathbf{x}] = \frac{1}{T}\sum_{t=1}^{T}\sum_{\tau=0}^{t-1} w(t,\tau)\cdot \mathrm{ERF}^{(\mathcal{S},\mathcal{T})}_{(i,j)}[\,y_{(m,n)}[t];\mathbf{x},\tau\,], \tag{4}$$

where $w(t,\tau)$ represents the relative contribution of the input with delay τ at time t to the output. The specific form of $w(t,\tau)$ depends on the neuronal dynamics and network architecture. For example, in Leaky Integrate-and-Fire (LIF) neurons, inputs closer to the current time step may have a higher influence due to the decay of membrane potential over time.

The temporal ERF (i.e., $\mathrm{ERF}^{(\mathcal{T})}$) can be seen as the integration over the spatial dimensions of ST-ERF to indicate the contribution of inputs at different timesteps to the final output:

$$\mathrm{ERF}^{(\mathcal{T})}[\tau;\mathbf{x}] = \sum_{i,j}\sum_{m,n}\mathrm{ERF}^{(\mathcal{S},\mathcal{T})}_{(i,j)}[\,y_{(m,n)}[T];\mathbf{x},\tau\,]. \tag{5}$$

As shown in Figure 1(c), we visualize an example of the ST-ERF. Similar to conventional ERF analysis, we focus on the center of the feature map at a specific timestep (e.g., the final timestep in Fig. 1(c)) to analyze the spatio-temporal feature representations in an SNN. Depending on the purpose of analysis, one may investigate the spatial distribution of the ST-ERF at a given timestep (spatial ERF) or its temporal distribution across one or more layers (temporal ERF).

3.3 Loss-Derived Calculation for ST-ERFs

Based on Eq. (3), computing the ST-ERF in SNNs requires evaluating first-order derivatives of outputs with respect to all input features. To obtain the ST-ERF conveniently, we introduce the loss-derived calculation method to efficiently compute using PyTorch's Automatic Differentiation functionality. Consider a SNN with input spike features $s^{\ell-1}$, output spike features s^{ℓ} at the ℓ-th layer, and an arbitrary loss function $\mathcal{L}$. The spatial ERF of SNNs can be easily obtained by calculating the average of the gradient of the loss with respect to input features at position (i, j) across all timesteps T. Specifically, it can be computed as follows:

$$\mathrm{ERF}^{(\mathcal{S})}_{(i,j)}[\, s^{\ell}_{(0,0)}] = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial s^{\ell}_{(0,0)}[t]}{\partial s^{\ell-1}_{(i,j)}[t]} = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial \mathcal{L}}{\partial s^{\ell-1}_{(i,j)}[t]}, \text{when} \forall t, \frac{\partial \mathcal{L}}{\partial s^{\ell}_{(\hat{i},\hat{j})}[t]} = \begin{cases} 1, & \hat{i}=0, \hat{j}=0, \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

The temporal ERF of SNNs can be obtained by calculating the sum of the gradient of the loss function with respect to input features at timestep $T-\tau$ across all spatial positions. It can be computed as:

$$\mathrm{ERF}^{(\mathcal{T})}[\tau] = \sum_{i,j}\sum_{\hat{i},\hat{j}}\frac{\partial s^{\ell}_{(\hat{i},\hat{j})}[T]}{\partial s^{\ell-1}_{(i,j)}[T-\tau]} = \sum_{i,j}\frac{\partial \mathcal{L}}{\partial s^{\ell-1}_{(i,j)}[T-\tau]}, \quad \text{when } \forall \hat{i},\hat{j}, \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[T]} = 1. \tag{7}$$

Proof can be found in Appendix A. We refer to the conditions in Eq. (6) and (7) as gradient stimuli. Based on this proposition, we could easily obtain the spatial and temporal ERF with automatic back-propagation without an explicit loss function.

4 Problem Analysis on Transformer-based SNNs using ST-ERF

In this section, we use the ST-ERF framework to analyze existing Transformer-based SNNs and identify their limitations in visual long-sequence modeling tasks.

4.1 Different ST-ERF Behaviors in Transformer-based SNNs

We apply the ST-ERF framework to analyze Transformer-based SNNs' spatial ERF behaviors across all timesteps. Specifically, we compared two groups of architectures(a: ViT-like architecture group and b: Meta-architecture group) with their ANN counterparts to investigate the differences in the formation of their spatial ERFs. For the loss-derived calculation, we set the central patch across all channels and timesteps in the output tensor as the gradient stimuli (uniform values of 1), then perform automatic back-propagation. Each experiment comprised 60 iterations using randomly sampled input tensors under standard normal distribution ($\mu = 0, \sigma^2 = 1$). Note that we average the ST-ERF over all timesteps to obtain a clear visualization.

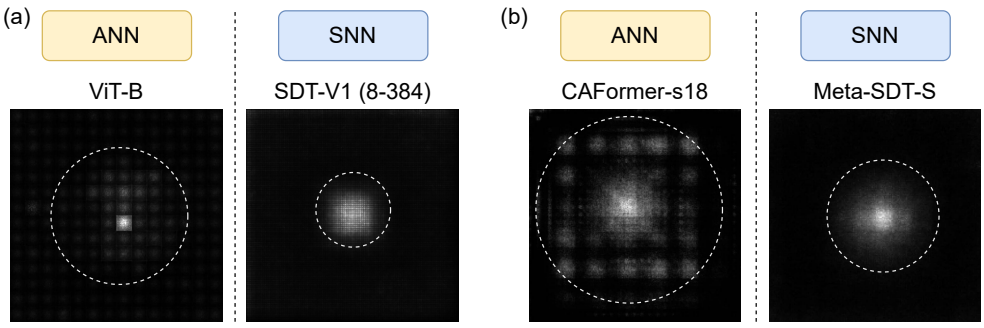

Figure 2: Comparison of spatial ERF with ANN Vision Transformers and ST-ERF with different Transformer-based SNNs. (a) ViT-like architecture comparison group: ViT-B and SDT-V1. (b) Meta-architecture comparison group: CAFormer-s18 and its counterpart Meta-SDT-S.

The comparison of ViT-like architectures is illustrated in Figure 2(a). Compared with the classic ViT-B, SDT-V1 exhibits a more centrally concentrated yet markedly narrower spatial ERF. This

observation suggests that the architectural modifications in SDT-V1 may restrict the receptive field's spatial extent, thereby enhancing its attention on local spatial dependencies. In fact, SDT-V1 adopts a fundamentally different strategy from the vanilla ViT in the patch splitting stage. Specifically, SDT-V1 employs the Spike Patch Splitting (SPS) module, consisting a Patch Splitting Module (PSM) to linearly project the input image and a Relative Position Embedding (RPE) [55] block to generate the latent position information. The SPS module incorporates multiple convolutional layers at the early stage of the network, facilitating low-level spatial features extraction from input images.

The comparison of meta-architectures is illustrated in Figure 2(b). Although Meta-SDT exhibits ERF behaviors similar to those of its ANN counterparts, it also struggles to maintain long-range feature attention. This limitation can be attributed to the additional employment of convolutional layers, which tend to emphasize localized features rather than global spatial contexts. Compared with CAFormer, Meta-SDT introduces the Re-parameterization Convolution (RepConv)[56] to perform the linear projection of queries, keys, and values[51]. This design enhances local feature extraction, yet it inherently constrains the model's capacity to aggregate information across distant spatial regions. Together, these findings suggest that the convolutional operations enhances local feature sensitivity but poses challenges for maintaining long-range spatial coherence in Transformer-based SNNs.

4.2 Visual Long-sequence Modeling Needs Global ST-ERF

Visual long-sequence modeling tasks often involve dense predictions across an entire image, requiring the processing of thousands of input tokens [25]. Therefore, capturing long-range dependencies and global context is crucial for achieving accurate and robust representations [57]. Prior studies have found that vision models with global receptive fields often excel at segmentation and detection, for instance when using self-attention mechanisms as in Transformer architectures [57, 58]. In contrast, architectures lacking global context integration tend to struggle. While early convolutional layers excel at extracting low-level structural patterns [59], their locality inherently limits the capacity to capture long-range dependencies, making them suboptimal for visual long-sequence tasks.

However, despite the need for global spatial awareness in visual long-sequence modeling, Transformers-based SNNs still fail to achieve a truly global ST-ERF. As discussed above, they tend to focus heavily on the center and expand to limited size. This contrasts sharply with the expected behavior required for visual long-sequence modeling tasks, where the weak global ST-ERF limits information aggregation and consequently degrades performance on such scenarios [57].

5 Methods

In this section, we propose two novel channel-mixing designs, MLPixer and SRB, which enable Transformer-based SNNs to more effectively capture long-range dependencies. Furthermore, we integrate these modules into the Meta-SDT architecture to enhance performance on visual long-sequence modeling tasks.

5.1 Design of Channel Mixer Block

To enhance the global modeling capability of SNN in visual long-sequence tasks, we propose two novel channel mixer designs. The first is the multi-layer perceptron-based mixer (MLPixer), which employs a two-layer MLP structure to more effectively extract global features. It is defined as follows:

$$\mathrm{MLPixer}(\mathbf{X}) = \mathrm{BN}\Big(\mathrm{MLP}\big(\mathbb{SN}\{\mathrm{BN}(\mathrm{MLP}\{\mathbb{SN}(\mathbf{X})\})\}\big)\Big), \tag{8}$$

where $\mathbf{X} \in \mathbb{R}^{T\times B\times N\times D}$ denotes the input of channel mixers in the Transformer block. $\mathbb{SN}(\cdot)$ denotes a spiking neuron layer that transforms the input sequence into the spike trains. $\mathrm{MLP}(\cdot)$ denotes a single-layer fully connected (FC) operation, and $\mathrm{BN}(\cdot)$ denotes batch normalization.

Compared with vanilla channel mixers [60, 51] that rely on convolution operations, the MLPixer employs a two-layer MLP operation to mix features across channels. This design reduces reliance on convolutional operations, mitigating the ERF's bias toward a Gaussian-like central concentration and enabling SNNs to capture long-range dependencies more effectively.

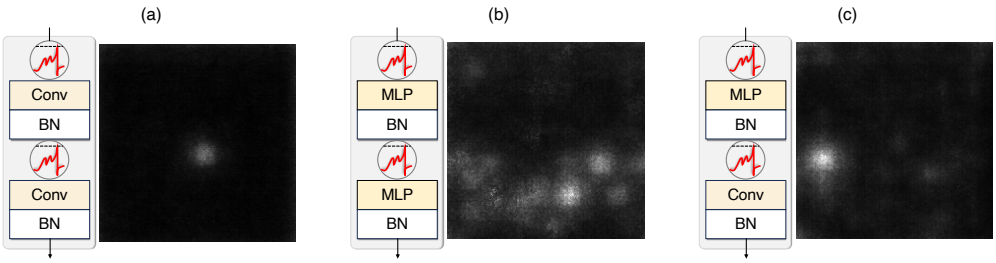

Figure 3: Comparison between the original channel mixer design and our proposed methods, along with their ST-ERF. For clearer visualization, the ST-ERF is averaged over timesteps. (a) Vanilla convolution-based channel mixer. (b) Proposed MLPixer architecture. (c) Proposed SRB architecture. Obviously, the vanilla convolution-based channel mixer exhibits a limited ST-ERF, whereas our MLPixer and SRB modules achieve a more global ST-ERF. Moreover, due to the reduced use of convolutions, MLPixer exhibits an even broader effective receptive field.

Building on this, we further propose the SRB module. It replaces only the second convolution in the channel mixer with a single-layer MLP operation. Specifically, the SRB is defined as follows:

$$\mathrm{SRB}(\mathbf{X}) = \mathrm{BN}\Big(\mathrm{MLP}\big(\mathbb{SN}\{\mathrm{BN}(\mathrm{Conv}\{\mathbb{SN}(\mathbf{X})\})\}\big)\Big). \tag{9}$$

Here, $\mathrm{Conv}(\cdot)$ denotes a 1×1 convolution operation. In this manner, SRB module reduces additional parameters while maintaining performance. To validate the effectiveness of our approach, we visualize the ERFs of the Conv-based mixer, the MLPixer, and the SRB modules.

As shown in Figure 3(a), the vanilla convolution-based channel mixer exhibits a limited ST-ERF. In contrast, the proposed MLPixer and SRB modules demonstrate a more global ST-ERF. Furthermore, the comparison between Figure 3(b) and Figure 3(c) further demonstrates that MLPixer exhibits a more global ERF. This stems from reduced use of convolutions and further suggests that MLPs provide stronger global modeling capacity than convolutional operators. We will validate the proposed module on visual long-sequence modeling tasks in the experiment section.

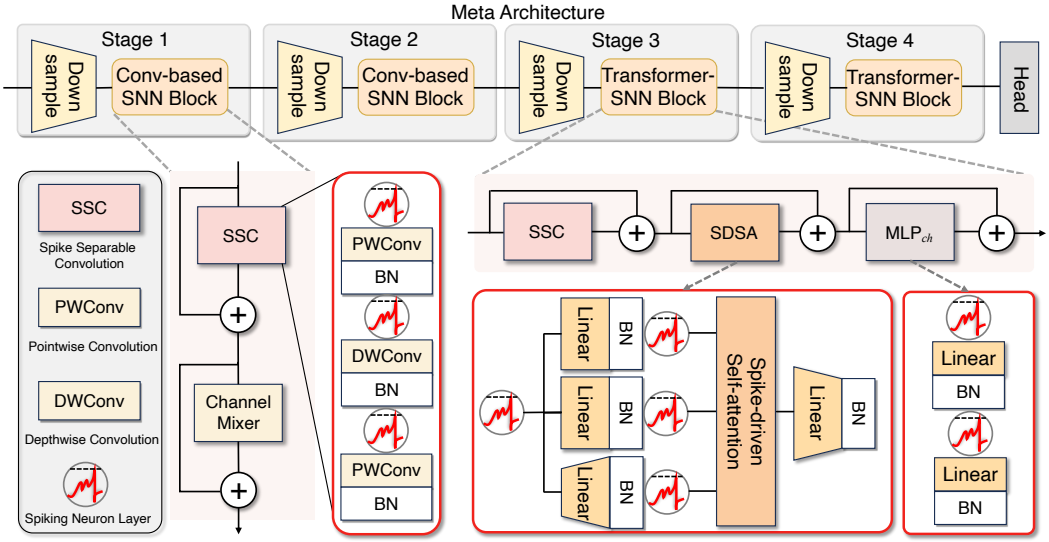

Figure 4: The overall architecture of Meta-SDT, which typically comprises four hierarchical stages. The first two stages use convolution-based SNN blocks, while the latter two adopt Transformer-SNN blocks. To strengthen the global modeling capacity of SNNs, we introduce two novel channel mixer architectures, MLPixer and SRB, to replace the convolution-based SNN blocks in the first two stages.

5.2 Overall Architecture

To further validate the effectiveness of our approach, we integrate the proposed SRB and MLPixer modules into the CAFormer [60] and Meta-SDT [33] architectures. As shown in Figure 4, these architectures adopt a multi-stage design, where the first two stages consist of Conv-based SNN blocks, and the latter two stages comprise Transformer-SNN blocks. In this work, we replace only the operations in the first two stages. Specifically, the first two stages are represented as:

$$\mathbf{X}' = \mathbf{X} + \mathrm{SSC}(\mathbf{X}), \mathbf{X}'' = \mathbf{X}' + \mathrm{Mixer}_{\epsilon}(\mathbf{X}'). \tag{10}$$

Here, $\mathrm{Mixer}_{\epsilon}(\cdot)$ is the channel mixer. In this work, we implement the approach using both the MLPixer module and the SRB module. Similar to the vanilla channel mixer, our method adopts an up-projection followed by a down-projection with a nonlinear activation in between, where $\epsilon > 1$ represents the intermediate dimensional expansion ratio. $\mathrm{SSC}(\cdot)$ is the spike-driven separable convolution block as token mixer, it is defined as follows:

$$\mathrm{SSC}(\mathbf{X}) = \mathrm{PWConv}_2(\mathbb{SN}(\mathrm{DWConv}(\mathbb{SN}(\mathrm{PWConv}_1(\mathbb{SN}(\mathbf{X})))))) \,. \tag{11}$$

$\mathrm{PWConv}_1(\cdot)$ and $\mathrm{PWConv}_2(\cdot)$ are pointwise convolutions, $\mathrm{DWConv}(\cdot)$ is depthwise convolution. $\mathbb{SN}(\cdot)$ denotes the spiking neuron layer. To maintain the spike-driven characteristics of the network, we implement membrane-shortcut residual connection mechanism. Furthermore, Transformer-SNN blocks are utilized in Stage 3 and Stage 4, following the same configuration as that of Meta-SDT-V3 [33]. We will further verify the effectiveness of the proposed method in the experimental section.

6 Experiments

In this section, we validate the effectiveness of our method through visualization and experimental analysis. First, we examine the changes in the ST-ERF after integrating the proposed modules into Meta-SDT, showing that our method achieves stronger global spatial receptive fields across all stages. Second, we evaluate its performance improvement on long-sequence modeling tasks, including object detection and semantic segmentation. Finally, we further investigate the method on complex event modeling tasks to assess its applicability in more challenging scenarios.

6.1 ST-ERF Behavior in Transformer-based SNNs

In order to study the impact of our proposed block on the receptive field of Meta-SDT, we compared temporal-averaged spatial ERFs between our two Meta-SDT variants with previous models. We initialized the central spatial feature across all channels and timesteps in the output tensor as uniform gradient stimuli (value = 1), and propagated the gradients backward through the network. Each experiment consisted of 60 iterations with input tensors randomly drawn from a standard normal distribution ($\mu = 0, \sigma^2 = 1$).

The results are illustrated in Figure 5. Surprisingly, we found that Spikformer exhibits diffuse receptive fields across all stages. The SDT-V1, Meta-SDT, and QKFormer demonstrate markedly centered distribution that gradually expand as the network deepens, all manifesting a Gaussian-like effect. Additionally, we observed dissipation of spatial ERF in SDT-V1 during the final stage. In contrast, our proposed two Meta-SDT variants establish robust global spatial receptive fields in the early stages. The MLPixer-SDT establishes a strong global spatial ERF in Stage 1. As the network deepens, its spatial ERF selectively contracts toward specific regions. The ERF behavior in SRB-SDT is slightly different, as it only begins to form a preliminary spatial ERF at Stage 2, and this distribution continues to evolve with increasing network depth.

6.2 Performance in Visual Long-sequence Modeling Tasks

We selected two challenging datasets to evaluate performance on classic visual long-sequence modeling tasks: object detection and instance segmentation on COCO 2017, and semantic segmentation on the ADE20K dataset. We choose the Meta-SDT(v3) [33] as the baseline and construct Meta-SDT variants with MLPixer($\epsilon 4$), MLPixer($\epsilon 6$) and SRB($\epsilon 4$).

Performance on COCO 2017 We evaluate the efficacy of the MLPixer and SRB on Meta-SDT and select the classic and large-scale COCO [61] dataset as our benchmark for evaluation. Following

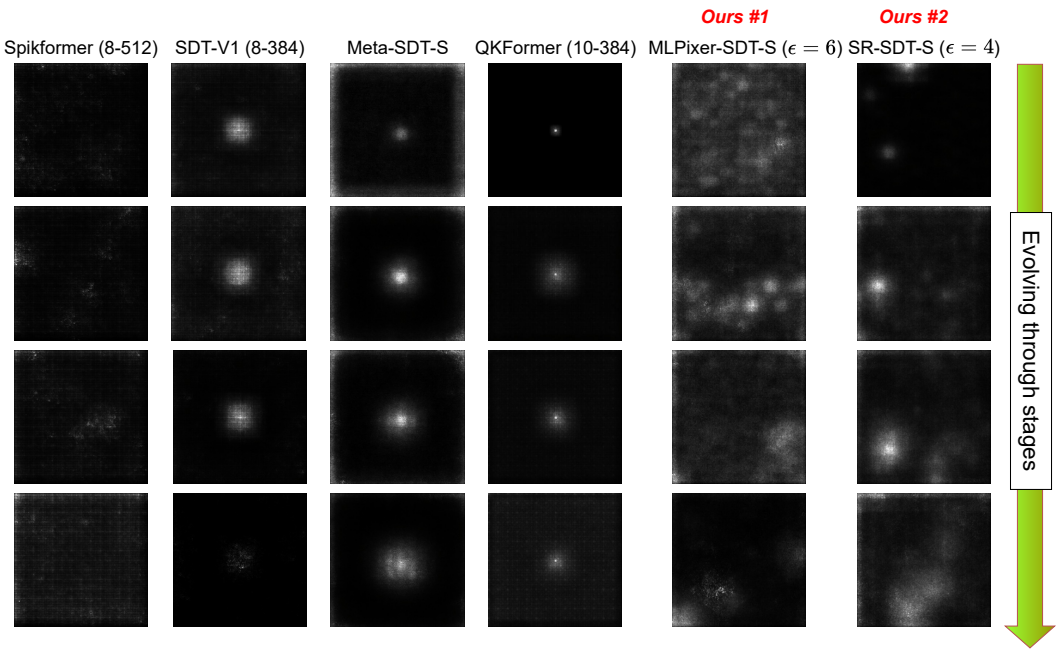

Figure 5: Comparison of temporal-averaged spatial ERF evolution across stages. From top to bottom are Stage 1 through Stage 4. Spikformer shows diffuse receptive fields across all stages. SDT-V1, Meta-SDT, and QKFormer exhibit more centered spatial distributions that gradually expand as depth increases. Our two Meta-SDT variants establish global spatial receptive fields in the early stages.

the previous work [51, 33, 52], we use the MMdetection [62] codebase with a spiking version and then deploy our model. We employ the Meta-SDT [33] with two variants as the backbone network to extract features, along with fine-tuning Mask R-CNN [63] for object detection and instance segmentation. All backbone networks are pretrained on ImageNet-1K [64], while the incremental layers are initialized following [65]. During fine-tuning, we strictly obey the $1\times$ training schedule.

The comparison results of object detection and instance segmentation are shown in Table 1. Under the same training schedule, both the MLPixer and SRB variants outperforms the baseline across all metrics. More specifically, the SRB variant exceeds the performance of SDTv3-T and SDTv3-B by 10.42% and 4.26% on the $\mathrm{AP}^{\mathrm{b}}_{50}$ metric, while maintaining almost the same model size. In conclusion, our approach demonstrates efficacy in object detection and instance segmentation, setting a new benchmark for COCO dataset in the SNN domain.

Performance on ADE20K We evaluate the performance of MLPixer and SRB on the semantic segmentation task using the challenging ADE20K dataset [66]. Similar to the COCO experiments, we utilize the spiking version of MMSegmentation [67] as our codebase and employ the Meta-SDT [33] with two variants as the backbone network. We fine-tune the Semantic FPN framework [68] for semantic segmentation. Backbone networks are initialized with ImageNet-1K pre-trained weights [64], and new layers follow the initialization scheme of [65]. All models are strictly obey the same training schedule for 160k iterations.

As shown in Table 2, both MLPixer and SRB variants surpass the baseline in terms of mIoU. The SRB variant improves performance by 3.3% and 2.6% over SDTv3-T and SDTv3-B, respectively, while reducing parameters by 0.3M and 1.2M. The MLPixer($\epsilon 4$) variant achieves the largest parameter reduction of 0.6M and 2.4M, with comparable or superior accuracy to SDTv3-T and SDTv3-B. These results highlight the effectiveness of the proposed modules in enhancing semantic segmentation on ADE20K.

6.3 Performance in Complex Event Modeling Tasks

Event-based Tracking We evaluate the effectiveness of two channel mixers in the context of event-based tracking, a highly challenging yet practically significant application domain for SNNs. Our experiments follow the SDTrack pipeline [18], which employs the Global Trajectory Prompt

Arch.	#T	#P	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
SDTv3-T[33]	4	25M	15.2	35.5	10.2	15.2	33.0	12.3
MLPixer(ϵ4)	4	24M	16.2	37.0	11.5	15.2	32.9	12.5
MLPixer(ϵ6)	4	25M	17.5	38.5	13.2	16.2	34.5	13.5
SRB(ϵ4)	4	25M	**18.2**	**39.2**	**13.8**	**17.5**	**34.8**	**14.3**
SDTv3-B[33]	4	39M	21.7	46.9	17.0	20.1	41.8	17.5
MLPixer(ϵ4)	4	36M	22.9	47.6	19.2	21.0	43.4	18.3
MLPixer(ϵ6)	4	39M	25.1	48.8	22.5	21.9	43.5	19.6
SRB(ϵ4)	4	37M	**25.8**	**48.9**	**22.8**	**22.5**	**43.9**	**20.4**

Table 1: Object detection and instance segmentation with Mask R-CNN on COCO val2017, using ImageNet-1K pre-training and $1\times$ training schedule.

Arch.	Ch. Mixer	#T	Param.(M)	mIoU(%)
SDTv3-T[33]	C2d-k3(ϵ4)	4	6.5 **BASE**	34.9 **BASE**
	MLPix.(ϵ4)	4	5.9 (↓**0.6**)	34.9 (↑**0.0**)
	MLPix.(ϵ6)	4	6.6 (↑**0.1**)	35.9 (↑**1.0**)
	SRB(ϵ4)	4	6.2 (↓**0.3**)	**38.2** (↑**3.3**)
SDTv3-B[33]	C2d-k3(ϵ4)	4	20.4 **BASE**	41.1 **BASE**
	MLPix.(ϵ4)	4	18.0 (↓**2.4**)	42.0 (↑**0.9**)
	MLPix.(ϵ6)	4	20.7 (↑**0.3**)	43.4 (↑**2.3**)
	SRB(ϵ4)	4	19.2 (↓**1.2**)	**43.7** (↑**2.6**)

Table 2: Segmentation results on ADE20K based on different mixer block, using ImageNet-1K pre-training and 160k iter.

method to convert event streams into event frames. We strictly adhere to the original training protocol, modifying only the backbone by replacing SDTrack with our proposed SDTrack+MLPixer or SDTrack+SRB variants. As presented in Table 3, extensive experiments on the FE108 [69] and VisEvent [70] datasets demonstrate that our architectures surpass the original SDTrack in several key metrics. These results confirm that both the MLPixer and SRB designs preserve the Transformers-based SNNs' performance, yet highlight opportunities for further improvement in subsequent temporal benchmarks.

Table 3: Performance comparison on event-based object tracking, a challenging yet important application for SNNs. Evaluation is conducted on two benchmark datasets, FE108 and VisEvent.

Architecture	Timesteps	Param. (M)	FE108 [69]		VisEvent [70]	
			AUC(%)	PR(%)	AUC(%)	PR(%)
SD-Track(Tiny) [18]	4×1	19.61	56.7	89.1	**35.4**	48.7
+MLPixer ($\epsilon=4$)	4×1	20.21	57.1	89.2	33.7	47.3
+MLPixer ($\epsilon=6$)	4×1	22.99	57.9	**90.1**	34.5	**48.9**
+SRB ($\epsilon=4$)	4×1	21.43	**58.2**	88.5	33.8	48.0

7 Conclusion

This paper presents ST-ERF as a novel framework for analyzing the spatial-temporal modeling behaviors in SNNs from a new perspective. Through this analysis, an inherent limitation in current Transformer-based SNN models is identified when applied to visual long-sequence modeling tasks. To address this limitation, two channel-mixer architectures, MLPixer and SRB, are proposed. Visualization of ST-ERF demonstrates that both modules enhance the global receptive field. Extensive experiments on long-sequence modeling tasks, including object detection and semantic segmentation, show that MLPixer and SRB improve overall performance, with SRB achieving an optimal balance between accuracy and model size. Furthermore, the study investigates complex event modeling tasks to assess the applicability of MLPixer and SRB in more challenging scenarios. Overall, the proposed ST-ERF framework offers valuable insights for the design and optimization of SNN architectures across a wide range of tasks.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62220106008 and 62271432), in part by the Shenzhen Science and Technology Program (Shenzhen Key Laboratory, Grant No. ZDSYS20230626091302006) and in part by the Program for Guangdong Introducing Innovative, Entrepreneurial Teams, Grant No. 2023ZT10X044, and in part by the State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Grant No. SKLBI-K2025010. This work was partially supported by UESTC Kunpeng&Ascend Center of Cultivation.

References

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

[2] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[3] Malu Zhang, Jiadong Wang, Jibin Wu, Ammar Belatreche, Burin Amornpaisannon, Zhixuan Zhang, Venkata Pavan Kumar Miriyala, Hong Qu, Yansong Chua, Trevor E Carlson, et al. Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE transactions on neural networks and learning systems*, 33(5):1947–1958, 2021.

[4] Malu Zhang, Shuai Wang, Jibin Wu, Wenjie Wei, Dehao Zhang, Zijian Zhou, Siying Wang, Fan Zhang, and Yang Yang. Toward energy-efficient spike-based deep reinforcement learning with temporal coding. *IEEE Computational Intelligence Magazine*, 20(2):45–57, 2025.

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

[6] Qiang Yu, Huajin Tang, Kay Chen Tan, and Haoyong Yu. A brain-inspired spiking neural network model with temporal encoding and learning. *Neurocomputing*, 138:3–13, 2014.

[7] Wenzhe Guo, Mohammed E Fouda, Ahmed M Eltawil, and Khaled Nabil Salama. Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Frontiers in Neuroscience*, 15:638474, 2021.

[8] Xuerui Qiu, Rui-Jie Zhu, Yuhong Chou, Zhaorui Wang, Liang-Jian Deng, and Guoqi Li. Gated attention coding for training high-performance and efficient spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(1):601–610, Mar. 2024.

[9] 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.

[10] Dehao Zhang, Shuai Wang, Yichen Xiao, Wenjie Wei, Yimeng Shan, Malu Zhang, and Yang Yang. Memory-free and parallel computation for quantized spiking neural networks. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[11] 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.

[12] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[13] De Ma, Xiaofei Jin, Shichun Sun, Yitao Li, Xundong Wu, Youneng Hu, Fangchao Yang, Huajin Tang, Xiaolei Zhu, Peng Lin, et al. Darwin3: a large-scale neuromorphic chip with a novel isa and on-chip learning. *National Science Review*, 11(5):nwae102, 2024.

[14] Wenjie Wei, Malu Zhang, Hong Qu, Ammar Belatreche, Jian Zhang, and Hong Chen. Temporal-coded spiking neural networks with dynamic firing threshold: Learning with event-driven backpropagation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10518–10528, 2023.

[15] Zhenxin Lei, Man Yao, Jiakui Hu, Xinhao Luo, Yanye Lu, Bo Xu, and Guoqi Li. Spike2former: Efficient spiking transformer for high-performance image segmentation. *arXiv preprint arXiv:2412.14587*, 2024.

[16] Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 253–272, Cham, 2025. Springer Nature Switzerland.

[17] Xuerui Qiu, Man Yao, Jieyuan Zhang, Yuhong Chou, Ning Qiao, Shibo Zhou, Bo Xu, and Guoqi Li. Efficient 3d recognition with event-driven spike sparse convolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(19):20086–20094, Apr. 2025.

[18] Yimeng Shan, Zhenbang Ren, Haodi Wu, Wenjie Wei, Rui-Jie Zhu, Shuai Wang, Dehao Zhang, Yichen Xiao, Jieyuan Zhang, Kexin Shi, Jingzhinan Wang, Jason K. Eshraghian, Haicheng Qu, Jiqing Zhang, Malu Zhang, and Yang Yang. Sdtrack: A baseline for event-based tracking via spiking neural networks, 2025.

[19] Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K. Eshraghian. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*, 2023.

[20] Xingrun Xing, Boyan Gao, Zheng Liu, David A. Clifton, Shitao Xiao, Wanpeng Zhang, Li Du, Zheng Zhang, Guoqi Li, and Jiajun Zhang. SpikeLLM: Scaling up spiking neural network to large language models via saliency-based spiking. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] Jingya Wang, Xin Deng, Wenjie Wei, Dehao Zhang, Shuai Wang, Qian Sun, Jieyuan Zhang, Hanwen Liu, Ning Xie, and Malu Zhang. Training-free ann-to-snn conversion for high-performance spiking transformer. *arXiv preprint arXiv:2508.07710*, 2025.

[22] Dehao Zhang, Shuai Wang, Ammar Belatreche, Wenjie Wei, Yichen Xiao, Haorui Zheng, Zijian Zhou, Malu Zhang, and Yang Yang. Spike-based neuromorphic model for sound source localization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[23] Shuai Wang, Dehao Zhang, Kexin Shi, Yuchen Wang, Wenjie Wei, Jibin Wu, and Malu Zhang. Global-local convolution with spiking neural networks for energy-efficient keyword spotting. *arXiv preprint arXiv:2406.13179*, 2024.

[24] Shuai Wang, Dehao Zhang, Ammar Belatreche, Yichen Xiao, Hongyu Qing, Wenjie Wei, Malu Zhang, and Yang Yang. Ternary spike-based neuromorphic signal processing system. *Neural Networks*, 187:107333, 2025.

[25] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[29] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

[30] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[31] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[32] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network, 2023.

[33] Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–18, 2025.

[34] Shuai Wang, Malu Zhang, Dehao Zhang, Ammar Belatreche, Yichen Xiao, Yu Liang, Yimeng Shan, Qian Sun, Enqi Zhang, and Yang Yang. Spiking vision transformer with saccadic attention. In *The Thirteenth International Conference on Learning Representations*, 2025.

[35] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[36] Hung Le and Ali Borji. What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? *arXiv preprint arXiv:1705.07049*, 2017.

[37] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[38] Zihao Wang and Lei Wu. Theoretical analysis of inductive biases in deep convolutional networks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[39] Yuandong Ma, Meng Yu, Hezheng Lin, Chun Liu, Mengjie Hu, and Qing Song. Efficient analysis of deep neural networks for vision via biologically-inspired receptive field angles: An in-depth survey. *Information Fusion*, 112:102582, 2024.

[40] Tomohiro Hayase and Ryo Karakida. Understanding mlp-mixer as a wide and sparse mlp. In *International Conference on Machine Learning*, 2023.

[41] Zhen Wei, Yao Sun, Jinqiao Wang, Hanjiang Lai, and Si Liu. Learning adaptive receptive fields for deep image parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2434–2442, 2017.

[42] Shanghua Gao, Zhong-Yu Li, Qi Han, Ming-Ming Cheng, and Liang Wang. Rf-next: Efficient receptive field search for convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):2984–3002, 2022.

[43] Peijie Dong, Xin Niu, Zimian Wei, Hengyue Pan, Dongsheng Li, and Zhen Huang. Autorf: Auto learning receptive fields with spatial pooling. In *International Conference on Multimedia Modeling*, pages 683–694. Springer, 2023.

[44] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 04 2017.

[45] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[46] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[47] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

[48] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 459–469, 2023.

[49] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[50] Xiangtai Li, Henghui Ding, Haobo Yuan, Wenwei Zhang, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10138–10163, 2024.

[51] Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *The Twelfth International Conference on Learning Representations*, 2024.

[52] Xuerui Qiu, Malu Zhang, Jieyuan Zhang, Wenjie Wei, Honglin Cao, Junsheng Guo, Rui-Jie Zhu, Yimeng Shan, Yang Yang, and Haizhou Li. Quantized spike-driven transformer. *arXiv preprint arXiv:2501.13492*, 2025.

[53] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022.

[54] Yuhao Wang and Wei Xi. Uniconvnet: Expanding effective receptive field while maintaining asymptotically gaussian distribution for convnets of any scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20922–20933, 2025.

[55] Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[56] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021.

[57] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 103031–103063. Curran Associates, Inc., 2024.

[58] Xiaoxia Qi, Md Gapar Md Johar, Ali Khatibi, and Jacquline Tham. Exploiting gaussian based effective receptive fields for object detection. *Scientific Reports*, 15(1):25008, July 2025.

[59] Chunyan Li, Zhiyong Li, Jianhong Sun, and Rui Li. Middle-shallow feature aggregation in multimodality for face anti-spoofing. *Scientific Reports*, 13(1):9870, 2023.

[60] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):896–912, 2024.

[61] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[62] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. arxiv 2019. *arXiv preprint arXiv:1906.07155*, 5, 1906.

[63] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[64] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[65] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[66] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127:302–321, 2019.

[67] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020.

[68] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, pages 6399–6408, 2019.

[69] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13043–13052, 2021.

[70] Xiao Wang, Jianing Li, Lin Zhu, Zhipeng Zhang, Zhe Chen, Xin Li, Yaowei Wang, Yonghong Tian, and Feng Wu. Visevent: Reliable object tracking via collaboration of frame and event flows. *arXiv:2108.05015*, 2021.

[71] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Online training through time for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:20717–20730, 2022.

[72] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[73] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly describe our contribution, the algorithm, and the experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper discusses the limitations of the work in Appendix C.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We mentioned our proposition and properties of ST-ERF and provided a whole set of proofs in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide a detailed description of our model architecture and present all the training details, including dataset processing methods and hyperparameter settings in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We mentioned our data in Appendix B. The code is compressed in the supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the full details in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Although we did provided $mean \pm std$ range to show several experiments' numerical range, we need to clarify that the numerical experiments are focused on the validation of ST-ERF properties, so we provided several independent trials with different input samples and different network initialization. The mean value of performance (e.g. the fitted curve) is solid enough to clarify our theories.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper provides sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper strictly adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work is a foundational research and not tied to particular societal applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, the paper properly credits the creators or original owners of assets (e.g., code, data, models) and explicitly mentions and respects the relevant licenses and terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets are introduced in this article.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The proposed method in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Proof of 3.3

Proof. Leaky Integrate-and-Fire (LIF) model [5, 2] can be described by the following equations:

$$\mathbf{v}^{\ell}[t] = \mathbf{h}^{\ell}[t-1] + f(\mathbf{w}^{\ell}, \mathbf{x}^{\ell-1}[t-1]), \quad \text{(Charging function)}, \tag{12}$$

$$\mathbf{s}^{\ell}[t] = \mathbf{\Theta}(\mathbf{v}^{\ell}[t] - \vartheta), \quad \text{(Firing function)}, \tag{13}$$

$$\mathbf{h}^{\ell}[t] = \begin{cases} \beta \mathbf{v}^{\ell}[t] - \vartheta \mathbf{s}^{\ell}[t], & \text{soft reset} \\ \mathbf{v}^{\ell}[t](1 - \mathbf{s}^{\ell}[t]), & \text{hard reset} \end{cases} \quad \text{(Leak-and-reset function)}, \tag{14}$$

where β is the decay constant, t is the time step, $\mathbf{w}^{\ell}$ is the weight matrix of layer ℓ, $f(\cdot)$ is the operation that stands for convolution (Conv) or fully connected (FC), $\mathbf{x}$ is the input, and $\mathbf{\Theta}(\cdot)$ denotes the Heaviside step function. When the membrane potential $\mathbf{v}$ exceeds the firing threshold ϑ, the LIF neuron will trigger a spike $\mathbf{s}$; otherwise, it remains inactive. After spike emission, the neuron invokes the reset mechanism, where the soft reset function is employed. $\mathbf{h}$ is the membrane potential following the reset function.

For the back-propagation of this neuron, we introduce the training process of SNN gradient descent and the parameter update method of spatio-temporal back-propagation (STBP) [9, 71]. The accumulated gradients of loss $\mathcal{L}$ with respect to weights $\mathbf{w}$ at layer ℓ can be calculated as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{\ell}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell+1}[t]} \frac{\partial \mathbf{s}^{\ell+1}[t]}{\partial \mathbf{v}^{\ell+1}[t]} \Big(\frac{\partial \mathbf{v}^{\ell+1}[t]}{\partial \mathbf{w}^{\ell}} + \sum_{\tau<t} \prod_{i=t-1}^{\tau} \left(\frac{\partial \mathbf{v}^{\ell+1}[i+1]}{\partial \mathbf{v}^{\ell+1}[i]} + \frac{\partial \mathbf{v}^{\ell+1}[i+1]}{\partial \mathbf{s}^{\ell+1}[i]} \frac{\partial \mathbf{s}^{\ell+1}[i]}{\partial \mathbf{v}^{\ell+1}[i]} \right) \frac{\partial \mathbf{v}^{\ell+1}[\tau]}{\partial \mathbf{w}^{\ell}} \Big), \tag{15}$$

where $\mathbf{s}^{\ell}[t]$ and $\mathbf{v}^{\ell}[t]$ represent the output spikes and membrane potential of the neuron in layer ℓ, at time t. Moreover, notice that $\frac{\partial \mathbf{s}^{\ell}[t]}{\partial \mathbf{v}^{\ell}[t]}$ is non-differentiable. To overcome this problem, Wu et al. [9] propose the surrogate function to make only the neurons whose membrane potentials close to the firing threshold receive nonzero gradients during back-propagation.

In this paper, we use the rectangle function, which has been shown to be effective in gradient descent and may be calculated by:

$$\frac{\partial \mathbf{s}^{\ell}[t]}{\partial \mathbf{v}^{\ell}[t]} = \frac{1}{a} \operatorname{sign}\left(\left| \mathbf{v}^{\ell}[t] - \vartheta \right| < \frac{a}{2} \right), \tag{16}$$

where a is a defined coefficient for controlling the width of the gradient window.

To compute $\sum_{t=1}^{T} \frac{\partial \mathbf{s}^{\ell}_{(0,0)}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]}$, we follow the chain rule with an arbitrary loss $\mathcal{L}$. Consider $\sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]}$:

$$\begin{aligned} \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]} &= \sum_{t=1}^{T} \sum_{\hat{i},\hat{j}} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]} \frac{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]} \\ &= \sum_{\hat{i}} \sum_{\hat{j}} \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]} \frac{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]} \\ &= \sum_{\hat{i}\neq 0} \sum_{\hat{j}\neq 0} \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]} \frac{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]} + \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(0,0)}[t]} \frac{\partial \mathbf{s}^{\ell}_{(0,0)}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]}. \end{aligned} \tag{17}$$

When the following conditions are met:

$$\forall t \in T, \frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[t]} = \begin{cases} 1 & \hat{i} = 0, \hat{j} = 0, \\ 0 & otherwise \end{cases}. \tag{18}$$

We can get:

$$\sum_{t=1}^{T}\frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]}=\sum_{t=1}^{T}\frac{\partial \mathbf{s}^{\ell}_{(0,0)}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]},\quad \frac{1}{T}\sum_{t=1}^{T}\frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]}=\frac{1}{T}\sum_{t=1}^{T}\frac{\partial \mathbf{s}^{\ell}_{(0,0)}[t]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[t]}. \tag{19}$$

The spatial ERF at position (i, j) can thus be calculated by summing the gradients of the loss with respect to all timesteps.

For the temporal ERF, we need to compute $\sum_{i,j}\frac{\partial s^{\ell}_{(0,0)}[T]}{\partial s^{\ell-1}_{(i,j)}[T-\tau]}$. We consider $\sum_{i,j}\frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[T-\tau]}$. By applying the chain rule:

$$\sum_{i,j}\frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[T-\tau]}=\sum_{i,j}\sum_{\hat{i},\hat{j}}\frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[T]}\frac{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[T]}{\partial \mathbf{s}^{\ell-1}_{(i,j)}[T-\tau]} \tag{20}$$

When the following conditions are met:

$$\forall \hat{i},\hat{j},\frac{\partial \mathcal{L}}{\partial \mathbf{s}^{\ell}_{(\hat{i},\hat{j})}[T]}=1 \tag{21}$$

We can simplify:

$$\sum_{i,j}\frac{\partial \mathcal{L}}{\partial s^{\ell-1}_{(i,j)}[T-\tau]}=\sum_{\hat{i},\hat{j}}\sum_{i,j}\frac{\partial s^{\ell}_{(\hat{i},\hat{j})}[T]}{\partial s^{\ell-1}_{(i,j)}[T-\tau]} \tag{22}$$

The temporal ERF at delay τ can thus be calculated by summing the gradients of the loss with respect to all spatial positions at timestep $T-\tau$. □

B Details in Detection and Segmentation Experiments

On ImageNet-1K pretraining, we employ three scales of Meta-SDT with three different channel mixer design ((1): Conv-Mixer; (2): MLPixer; (3): SRB) in Table 4 and utilize the hyper-parameters in Table 5 to pre-train models in our paper for further fine-tuning on COCO 2017 and ADE20K datasets. Note that ϵ represents the channel expand ratio (CHW $\rightarrow \epsilon$ CHW $\rightarrow$ CHW).

For COCO 2017 dataset, We utilize the MMDetection [72] framework to implement the existing models and our method. The object detection and instance segmentation framework strictly follows Mask R-CNN, with a training schedule of $1\times$ (12 epochs). We use a total batch size of 4/GPU, utilize the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 0.05. Images are resized and cropped into 1333 × 800 for training and testing and maintain the ratio. Random horizontal flipping and resize with a ratio of 0.5 was applied for augmentation during training. This pre-training fine-tuning method is a commonly used strategy in ANNs.

For ADE20K dataset, we utilize the MMSegmentation [73] framework. The training configuration strictly encompasses for 160,000 iterations. The batch size is set to 4/GPU, and the AdamW optimizer is used. The learning rate and weight decay parameters are tuned to 2×10^{-4} and 0.05, respectively. To speed up training, we warm up the model for 1.5k iterations with a linear decay schedule. All the experiments are conducted on 4 NVIDIA-A100 80GB GPUs.

Table 4: Configurations of different Meta-SDT Variants.

Stage	# Tokens	Layer Specification			Tiny	Medium	Base
1	$\frac{H}{2}\times\frac{W}{2}$	Downsampling		Conv	7x7 stride 2		
				Dim	16	24	32
		Conv-based SNN block	SepConv	DWConv	7x7 stride 1		
				MLP ratio	2		
			Channel Mixer	(1)Conv+Conv	$\epsilon = 4$		
				(2)MLP+MLP	$\epsilon = 4/6$		
				(3)MLP+Conv	$\epsilon = 4$		
	$\frac{H}{4}\times\frac{W}{4}$	Downsampling		Conv	3x3 stride 2		
				Dim	32	48	64
		Conv-based SNN block	SepConv	DWConv	7x7 stride 1		
				MLP ratio	2		
			Channel Mixer	(1)Conv+Conv	$\epsilon = 4$		
				(2)MLP+MLP	$\epsilon = 4/6$		
				(3)MLP+Conv	$\epsilon = 4$		
2	$\frac{H}{8}\times\frac{W}{8}$	Downsampling		Conv	3x3 stride 2		
				Dim	64	96	128
		Conv-based SNN block	SepConv	DWConv	7x7 stride 1		
				MLP ratio	2		
			Channel Mixer	(1)Conv+Conv	$\epsilon = 4$		
				(2)MLP+MLP	$\epsilon = 4/6$		
				(3)MLP+Conv	$\epsilon = 4$		
			# Blocks		2		
3	$\frac{H}{16}\times\frac{W}{16}$	Downsampling		Conv	3x3 stride 2		
				Dim	128	192	256
		Transformer-based SNN block	SDSA	RepConv	3x3 stride 1		
			Channel MLP	MLP ratio	4		
			# Blocks		6		
4	$\frac{H}{16}\times\frac{W}{16}$	Downsampling		Conv	3x3 stride 1		
				Dim	192	240	360
		Transformer-based SNN block	SDSA	RepConv	3x3 stride 1		
			Channel MLP	MLP ratio	4		
			# Blocks		2		

Table 5: Hyper-parameters for pre-training on ImageNet-1K

Hyper-parameter	Settings	Hyper-parameter	Settings
Model size	T/M/B	Timestemp	4
Epochs	200	Resolution	224*224
Batch size	1568	Optimizer	LAMB
Base learning rate	6e-4	Learning rate decay	Cosine
Warmup eopchs	10	Weight decay	0.05
Random augment	9/0.5	Mixup	None
Cutmix	None	Label smoothing	0.1

C Limitations

This work presents several avenues for future exploration, such as how neuronal dynamics parameters influence ST-ERF in more dynamic and diverse SNNs. Given that one of SNN's major successes stems from its inherent membrane potential memory update mechanism, this represents a particularly worthwhile direction for deeper investigation. We will further explore the interactions between spiking neurons' neurodynamics and the networks' temporal response in the future. Nevertheless, this work provides a viable analytical framework for understanding SNN model behavior, with practical implications for architectural design across various levels of SNNs.