Joint-Scnn: Joint Spiking Convolution Network for Direction of Arrival Estimation

Zhixuan Zhang, Qi Liu, *Senior Member, IEEE*, and Zhenao Wei

***Abstract*—Direction of arrival (DOA) refers to finding direction information of propagating waves from the received antennas equipped with several sensors. Recently, we have witnessed an enrichment of source data prompting us to design more robust DOA estimator, where artificial neural network (ANN)-based DOA estimators have shown their superior performance as compared to the traditional subspace-based DOA estimation methods. However, these data-driven DOA estimation methods tend to rely on parameters that are computationally intensive for efficient processing/running with the limitation of hardware resources. Thus, we propose an event-driven spiking neural network (SNN) model, namely, Joint-Scnn, for DOA in the presence of various imperfections, which consists of ANN-based and SNN-based modules with weights sharing. The former not only contributes to assist sparse SNN-based module to learn latent information, but also enhances its robustness via self-learning. The superior estimation performance and lower power consumption have been verified via experimental results. The success of Joint-Scnn is partially attributed to the teacher-student tandem learning scheme.**

***Index Terms*—Direction of arrival, spiking neural network, tandem learning, transfer learning.**

I. Introduction

DIRECTION of arrival (DOA) estimation, as one of the central problems in radar localization and tracking application, has attracted a lot of research interests over the past decades. The DOA estimators are commonly used in signal processing for determining DOA of a signal at a receiver array, which can be classified into conventional optimal algorithms and popular deep learning (DL)-based methods.

In the past few decades, traditional DOA algorithms have been proposed and widely used in various radar communication fields, including MUSIC [1], ESPRIT [2], Root-MUSIC [3], MVDR [4], ML [5]. The MUSIC algorithm [1] is one of the most popular DOA estimation methods, which used the eigenvalue decomposition of the covariance matrix to estimate the DOAs of the sources. [2] proposed a high-resolution DOA estimation method called ESPRIT, utilizing the rotational invariance property of uniform linear arrays to estimate the DOAs of the sources. In [3], a variant of the MUSIC algorithm called Root-MUSIC, was designed that estimated the DOAs of the sources by finding the roots of a polynomial function derived from the covariance matrix. Then, by minimizing the variance of the beamformer's output power under distortionless response constraints, the DOA of a source was estimated using the MVDR algorithm in [4]. ML algorithm [5] estimated DOAs of the sources by maximizing the likelihood function of the received signals.

With the rise of artificial neural networks (ANNs) and the increase of the radar sensor data, researchers turn their insights into neural networks to solve these kinds of massive computing problems [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. In [6], an encoder-decoder framework was proposed to address the DOA estimation problem, composed of a multi-task autoencoder to decompose the input into multiple components in different spatial sub-regions and a series of parallel multilayer classifiers. In [7], a deep convolution network (DCN) that learned the inverse transformation, was introduced, where the columns of the array covariance matrix were formulated as under-sampled linear measurements of the spatial spectrum, corrupted by noise. Reference [8] applied a CNN-based method for the DOA estimation, and estimated the DOAs by discretizing the spatial domain into grids. In [9], a synthetic dataset for angle classification was shown under the presence of additive noise, propagation attenuation, and delay. [10] proposed an offline and an online DNN approaches for the DOA estimation in the massive multiple-input multiple-output system. In [11], multiple CNNs were designed to extract features from the MUSIC spectrum of the received signal.

Nevertheless, with the development of wider and deeper neural networks, ANN-based DOA estimators comes at the cost of computational complexity, which renders to develop robust and lightweight DOA estimation method. Motivated by the SNNs as the third generation of neural networks, with the properties of sparse firing and spike-based information transmission, we propose to estimate DOAs under the framework of SNNs. SNNs typically can be categoried into three folds: 1) due to the discreteness of spikes and the intrinsic non-differentiability of spike firing function impeding the direct applicability of the traditional back-propagation (BP), surrogate-based BP algorithm thus can be derived by replacing the spiking function with differential functions or adding some limitation or clipping to the BP process [17], [18], [19], [20]. 2) Based on

Manuscript received 5 September 2023; revised 19 March 2024; accepted 13 June 2024. Date of publication 17 June 2024; date of current version 7 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62202174, in part by the Fundamental Research Funds for the Central Universities under Grant 2023ZYGXZR085, in part by the Basic and Applied Basic Research Foundation of Guangzhou under Grant 2023A04J1674, and in part by the Guangdong Provincial Key Laboratory of Human Digital Twin under Grant 2022B1212010004. The review of this article was coordinated by Dr. Balasubramaniam Natarajan. *(Corresponding author: Qi Liu.)*

The authors are with the School of Future Technology, South China University of Technology, Guangzhou 511442, China (e-mail: zhangzhixuan77@gmail.com; drliuqi@scut.edu.cn; wza@scut.edu.cn).

Digital Object Identifier 10.1109/TVT.2024.3415438

spike-timing-dependent-plasticity (STDP) algorithm, SNN models learn each module separately [21], [22], [23], [24], [25]. 3) Different from the above SNN algorithms, the ANN-to-SNN algorithms assume that SNNs have equal efficiency as the counterpart ANNs [26], [27], [28]. Our proposal belongs to the latter.

Recently, a novel ANN-to-SNN algorithm [29] has been proposed, called spiking tandem learning. Different from other ANN-to-SNN algorithms, each layer of this framework is composed of an ANN-based module and a weight-sharing spike-based module with the similar structure. The spike module as a student is utilized to obtain the exact neural representation, while the ANN-based module as a teacher is devised to estimate the surrogate gradients on the spike-train level and helps the spike-based module to learn, resulting in the lower latency and almost comparable performance to the counterpart ANN. However, there are few drawbacks on the spiking tandem learning algorithm: 1) Due to the character of weight sharing, the error of the ANN-based module will increase when error exists in the spike module. To other word, "a bad student will affect on a worse teacher to some extent". 2) In the original framework, the ANN-based module aims to deal with the non-differentiability problem of spike function, and ignores its self-learning to enhance the whole performance.

To address these problems, we propose the Joint-Scnn method for DOA estimation task. Our work makes the notable contributions as follows:

- To the best of our knowledge, it is the first time to successfully apply the SNN model for DOA in the presence of various imperfections.
- A novel spiking tandem learning framework is proposed, named Joint-Scnn, to increase the whole performance via ANN self-learning.
- The proposed Joint-Scnn method enjoys better DOA estimation accuracy as well as lower power consumption.

The remainder of the paper is organized as follows. In Section II, DOA estimation with imperfections is modeled. The proposed method is introduced in Section III. In Section IV, simulation results show that our method outperforms the benchmarks. Finally, conclusions are drawn in Section V.

II. Problem Statement

Assume that K narrowband far-field signals $\mathbf{s}(t)$ impinge onto a linear array of M omnidirectional sensors, whose locations md are described by an integer set $\mathcal{D} = \{D_1, D_2, \ldots, D_M\}$, $m \in \mathcal{D}$ and $d = \lambda/2$ denotes the half wavelength. The incident directions of signals are represented by $\{\theta_k\}_{k=1}^{K}$. In the presence of additive Gaussian noise $\mathbf{n}(t) \in \mathbb{C}^{M\times 1}$, the output of received array at time instant t is modeled as below:

$$\mathbf{y}(t) = \mathbf{A}(\theta)\mathbf{s}(t) + \mathbf{n}(t),\; t = 1, \ldots, N \tag{1}$$

where $\mathbf{A}(\theta) = [\mathbf{a}_1(\theta), \mathbf{a}_2(\theta), \ldots, \mathbf{a}_K(\theta)] \in \mathbb{C}^{M\times K}$ with the steering vector $\mathbf{a}(\theta_k) = [e^{v_{D_1}}, e^{v_{D_2}}, \ldots, e^{v_{D_M}}]^T$, $v_{D_i} = -j2\pi d \sin(\theta_k)/\lambda$ and $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_K(t)]^T \in \mathbb{C}^{K\times 1}$. N is the number of snapshots. Assuming that the Gaussian noise is distributed with a mean of zero and a certain variance σ^2, and all signals are uncorrelated with the noise.

In practice, various imperfections exist in sensor arrays for real-world radar applications to directly affect $\mathbf{a}_K(\theta)$ with deviations [6], such as mutual coupling, sensor position errors, gain and phase inconsistencies, etc. With the effect of imperfections, the array output vector is re-written as:

$$\mathbf{y}(t) = \mathbf{A}(\theta, \epsilon)\mathbf{s}(t) + \mathbf{n}(t). \tag{2}$$

As for the diversity of array imperfections, it becomes challenging for traditional optimization methods to completely figure out these problems together because the resulting optimization approaches are generally designed based on one specific imperfection. The covariance matrix of the output vector array $\mathbf{y}(t)$ can be expressed as:

$$\mathbf{R} = E\{\mathbf{y}(t)\mathbf{y}^H(t)\} = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma^2\mathbf{I}_M \tag{3}$$

where $\mathbf{R}_s = E\{\mathbf{s}(t)\mathbf{s}^H(t)\} = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_K^2\}$.

As for the massive amount of learnable parameters, deep ANN is endowed with better capability in fitting training data than conventional DOA estimators. Similar to [6], following the rules of RBF [30] and SVR [31] -based methods, we build the off-diagonal upper right matrix elements $\mathbf{R}_{ur}$ as the input vector to feed ANN:

$$\mathbf{R}_{ur} = [\mathbf{R}_{1,2}, \mathbf{R}_{1,3}, \ldots, \mathbf{R}_{1,M}, \mathbf{R}_{2,3}, \ldots, \mathbf{R}_{2,M}, \ldots, \mathbf{R}_{M-1,M}]$$

$$\mathbf{r} = \left[Real\{\mathbf{R}_{ur}^T\}, Image\{\mathbf{R}_{ur}\}^T\right]^T / ||\mathbf{R}_{ur}||_2 \tag{4}$$

where $Real\{\cdot\}$ and $Image\{\cdot\}$ denote the real and imaginary components of the complex-valued matrix, respectively. This is also commonly used by all ANN-based DOA estimation approaches, which is due to the existence of abundant feature information in the covariance matrix. The motivation behind (4) is that bottom left and upper right parts of the covariance matrix are conjugate, and there exists unknown noise variance laying on its diagonal elements.

III. Methodology

In this section, we introduce the proposed model framework and learning algorithm in detail, respectively.

A. Model Framework

From the viewpoint of energy consumption, we apply the spike-based computational method to devise a spiking deep convolutional neural network model for the task of DOA estimation. As shown in Fig. 1, the proposed model is composed of an encode layer, three spiking blocks, and a decoder layer. Details of the proposed model are described below.

1) Neuron Models: In our proposed neural network model, there exists two neurons: the ReLU neuron and the integrate-and-fire (IF) neuron. ReLU neurons are frequently built in ANNs, which directly clip the activation below zero. As a category of spike neurons, IF neurons are inspired by the spike-based information transmission in real biological neurons, used in SNNs. The synaptic transmission from presynaptic to postsynaptic neurons occurs through the positive correlation between the firing

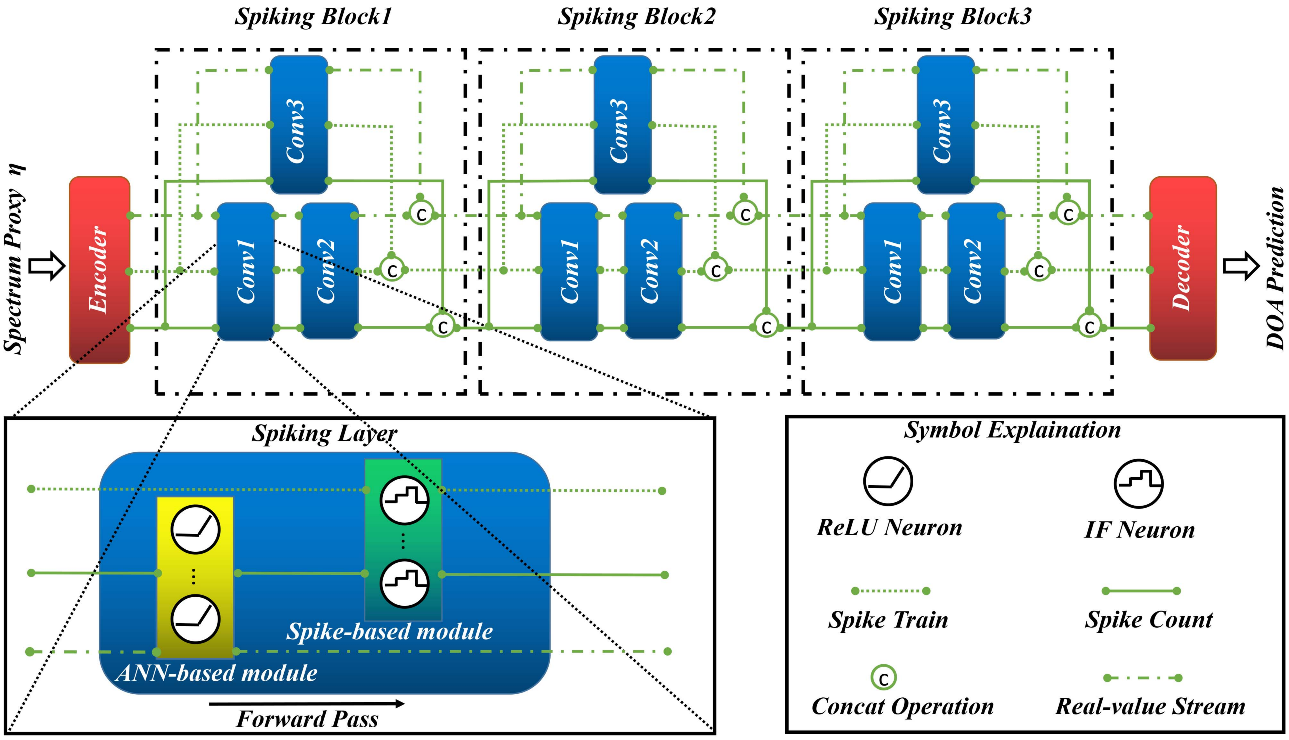


Fig. 1. Architectures of the proposed neural networks. For each Spiking layer, the ANN-based module and the spike-based module share the weights. It is worth noting that the ANN-based module is not needed for the inference and 50 time steps is totally required to SNN learning for each frame.

time of presynaptic spikes and their corresponding membrane potentials. In a simulation time window N_t, the incoming spikes to neuron j at layer l are integrated into subthreshold membrane potential V_j^l per time step t. Therefore, the membrane potential of neuron j at layer l is modeled as [24], [28]:

$$V_j^l[t] = V_j^l[t-1] + RI_j^l[t] - \vartheta S_j^l[t-1] \tag{5}$$

in which $I_j^l[t] = \sum_i w_{ji}^{l-1} S_i^{l-1}[t] + b_j^l$ and $S_j^l[t] = \Theta(V_j^l[t] - \vartheta)$ with the indicator function

$$\Theta(x) = \begin{cases} 1, \text{ if } x \geq 0 \\ 0, \text{ otherwise.} \end{cases} \tag{6}$$

The synaptic weight w_{ji}^{l-1} contributes to connecting presynaptic neuron i from layer $l-1$ and b_j^l is a constant injecting current. Additionally, $S_j^l[t-1]$ denotes the instance of a spike generated by the afferent neuron i at time t, and $I_j^l[t]$ is the resulting synaptic current from incoming spike sequences. At the arrival time t, the membrane potential $V_j^l(t)$ surpasses the pre-specified threshold ϑ (typically $\vartheta = 1$), triggering the generation of an action potential, commonly referred to as a spike. That is:

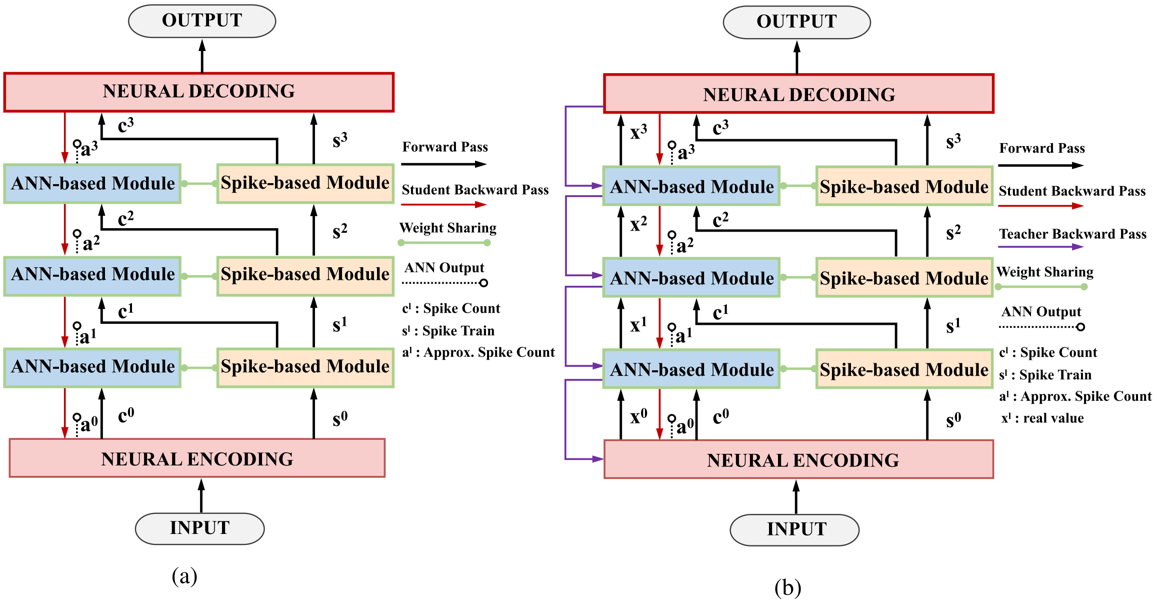
$$V_j^l(t) \geq \vartheta, \; \frac{dV_j^l(t)}{dt} > 0. \tag{7}$$

After firing, the $V_j^l(t)$ resets to the rest potential V_{rest} while staying at the refractory period for a duration of time. The synaptic weight, also known as the conductance of a synapse, varies based on the activities of the presynaptic and postsynaptic neurons. The ability of the neuron to learn is credited to synaptic plasticity, which is influenced by these activities. Moreover, the encoding methods can be divided into spike-rate-based related to spike counts in a time interval, and spike-time-based encoding related to every detail spike time. We introduce our used encoding method in the next subsection.

2) Encoder Layer & Decoder Layer: The encoder layer and decoder layer represent the input module and the output module in our proposed model. The time-dependent input currents are considered as the real-valued inputs and are directly applied in (5) at every time step. This neural encoding approach overcomes the sampling error of the rate code, thereby facilitating accurate and prompt inference, as demonstrated in previous studies [17], [32]. As shown in Fig. 1, the spiking layer receives spike trains and spike counts as input.

In order to enable pattern classification, the SNN back-end needs to decode the output spike trains into pattern classes. Decoding can be achieved from the SNN output layer using either discrete spike counts or continuous free aggregate membrane potentials (without spiking). The use of free aggregate membrane potentials results in a smoother learning curve as it enables the derivation of continuous error gradients at the output layer [29]. Thus, we take a 1D-convolution layer as the decoder layer in our proposed model.

3) Spiking Block: The design for the spiking block is inspired by the superiority of supplying additional information via skip-layer connections derived from the residual structure of ResNet. The spiking block mainly consists of three spiking layers, namely Conv1, Conv2, and Conv3. Every spiking layer is composed of a real-valued convolution module and spike-based



(a)

(b)

Fig. 2. (a) Original tandem learning in SNN vs (b) our proposed joint tandem learning in SNN.

convolution module with the sharing parameters (e.g. convolution kernel), in which the real-valued convolution module works on completing the parameter training and spike-based convolution module is responsible for the inference phase (testing or validating). Conv1 connects Conv2 with the same convolution kernel, while connecting Conv3 with a 1x1 convolution kernel, as shown in Fig. 1. Given the input channel *in_ch* and output channel *out_ch*, the input channel of Conv2 and the output channel of Conv1 and Conv3 are set as *out_ch* / 2.

B. Joint Spiking Tandem Learning

As shown in Fig. 2(a), ANN-based module plays the part of teacher to supervise SNN-based counterpart to learn on the assumption that the spike counts from SNN-based module can be approximated by the real values from ANN-based module. However, there exists approximation loss to result in performance degradation especially in the noisy circumstance. Moreover, it performs not so good to generalize to unseen data drawn from the same distribution. To address that, as shown in Fig. 2(b), a learning pathway is designed to independently pass over every ANN layers via real-valued data x^l. This is motivated by the multi-task learning mechanism [33] to enhance the robustness. The main task is to train spike-based module by sharing weights from ANN-based module, while the related task is to train the ANN-based module itself.

For DOA estimation problem, our goal is to minimize the mean squared error (MSE) between actual and estimated outputs. Given data $x_1, x_2, \ldots, x_N$ with labels $y_1, y_2, \ldots, y_N$, the model outputs are denoted as $\hat{y_1}, \hat{y_2}, \ldots, \hat{y_N}$. Then loss function $\mathcal{L}_{MSE}$ can be formulated as:

$$\mathcal{L}_{MSE}(\hat{y}, y) = \frac{1}{2}\sum_{i=1}^{N} \|\hat{y_i} - y_i\|^2 \tag{8}$$

As mentioned above, to achieve the multi-task learning, the joint loss function $\mathcal{L}_{joint}$ is applied, where $\hat{y_c}$ and $\hat{y_x}$, represent the outputs of main task (along the pathway of s^i and c^i) and related task (along the pathway of x^i), respectively. The hyperparameter α is equal to 0.9:

$$\mathcal{L}_{main} = \mathcal{L}_{MSE}(\hat{y_c}, y) \tag{9}$$

$$\mathcal{L}_{related} = \mathcal{L}_{MSE}(\hat{y_x}, y) \tag{10}$$

$$\mathcal{L}_{joint} = \alpha\mathcal{L}_{main} + (1-\alpha)\mathcal{L}_{related} \tag{11}$$

As for the highly discrete and non-differentiable nature of SNNs in spike generation, the error BP method cannot be directly applicable to the training process. To that end, we apply the ANN-to-SNN conversion learning rule, namely, weight updating between ANN and SNN, by spike count approximation. Next, we compute the spike count from neuron i at layer l as:

$$c_i^l = \sum_{t=1}^{N_t} S_i^l[t] \tag{12}$$

To build the relationship between ANN's activation functions and SNN's spike counts, they are connected via shared weights. This is mathematically modeled by

$$\Delta t = \rho\left(\frac{\vartheta}{V_j^l / N_t}\right) \tag{13}$$

in which the aggregated action potential of neuron j in layer l is written as $V_j^l = \sum_i w_{ji}^{l-1} c_i^{l-1} + b_j^l N_t$. ϑ, ρ and Δt denote the firing threshold, the non-linear transformation of ReLU, and the time step, respectively. Hence, the approximated spike count a_j^l is obtained as:

$$a_j^l = \frac{N_t}{\Delta t} = \frac{1}{\vartheta}\rho\left(\sum_i w_{ji}^{l-1} c_i^{l-1} + b_j^l N_t\right) \tag{14}$$

Algorithm 1: Joint Spiking Tandem Learning.

```
Input: Spectrum proxy $\eta$; DOA label $L$; Total epoch $\mathbf{E}$; Jonit-Scnn $\mathbf{F}$={Encoder layer $\mathbf{F}_{en}$, spiking blocks $\mathbf{F}_{s_i}$(i=1,2,3), Decoder layer $\mathbf{F}_{de}$} with parameter $\theta_{\mathbf{F}}$; Learning rate $\gamma$; Hyperparameter $\alpha$.
Output: Model parameters $\theta^{*}_{\mathbf{F}}$.
for $e \leftarrow 1$ to E do
    for $i \leftarrow 1$ to $len(\eta)$ do
        // Extract training data
        $x \leftarrow \eta[i]$, $y \leftarrow L[i]$
        // In encoder layer
        $s^0$, $c^0$, $x^0 \leftarrow \mathbf{F_{en}}(x)$ (as in Eq. 5)
        // In spiking blocks
        for $\ell \leftarrow 1$ to $3$ do
            $s^{\ell}$, $c^{\ell}$, $x^{\ell} \leftarrow \mathbf{F}_{\mathbf{s}_{\ell}}(s^{\ell-1}, c^{\ell-1}, x^{\ell-1})$
        end
        // In decoder layer
        $\hat{y}, \hat{y_c} \leftarrow \mathbf{F}_{de}(s^3, c^3, x^3)$
        // Construct loss functions
        $\mathcal{L}_{main} \leftarrow Eq.9$ with input of $\hat{y_c}$ and $y$
        $\mathcal{L}_{related} \leftarrow Eq.10$ with input of $\hat{y}$ and $y$
        $\mathcal{L}_{joint} \leftarrow \alpha\mathcal{L}_{main} + (1-\alpha)\mathcal{L}_{related}$ (as in Eq. 11)
        // Update parameters of Joint-Scnn
        $\theta_{\mathbf{F}} \leftarrow \theta_{\mathbf{F}}$-$\gamma\nabla_{\theta_{\mathbf{F}}}\mathcal{L}_{joint}$
        $\theta^{*}_{\mathbf{F}} \leftarrow \theta_{\mathbf{F}}$
    end
end
```

Thus, a^l_j can be effectively determined by the ReLU activation function in the ANN-based module via inputting the spike count c^{l-1}_i and injecting the current aggregated constant $b^l_j N_t$ as the bias term. By simplifying the spike generation process, it becomes feasible to estimate the error gradients at the spike-train level from those ANN-based modules [29] according to the tandem learning rule with firing rate instead of spike timings.

The overall process of Joint Spiking Tandem Learning is shown in Algorithm 1. In a training epoch, spectrum proxy η as input to Joint-Scnn $\mathbf{F}$ is first encoded into spike trains s^0 and spike counts c^0 at encoder layer of $\mathbf{F}$ obeyed by (5). It notes that the original η is also copied as the one of the output of encoder layer, namely x^0, to pass the ANN-based module of the first spiking block. Then the resulting three spiking blocks all receive the spike trains $s^{\ell-1}$, spike counts $c^{\ell-1}$ and real-value output $x^{\ell-1}$ from the previous blocks and output the new spike trains s^{ℓ}, spike counts c^{ℓ} and real-value output x^{ℓ}. Besides, the decoder layer processes the spike counts and real-value output from the last spiking block and outputs the DOA predictions $\hat{y_c}$, $\hat{y}$ of spike-based modules and ANN-based modules. Finally, we construct the loss function using (11) and update parameters of Joint-Scnn by SGD.

IV. Experimental Results

In this section, we carry out simulations to demonstrate the predominance of our proposed model over other benchmarks. The simulations are implemented on PyTorch [34] framework, in which ANN-based models can be easily constructed and gradient computation in the training phase can be automatically finished by its embedding tools. For a fair comparison, we choose methods in [6] and [7] as benchmarks since their backgrounds are similar to ours. We first describe the experimental setting.

A. Experimental Settings

The experimental setup employs an 8-element uniform linear array (ULA) with half-wavelength inter-element spacing, in which all simulation parameters are consistent with that described in [7]. It should be noted that the numbers of training and test samples are 15840 and 3960 respectively. Additionally, we set the dense of DOA grids $\Delta\phi$ as 1. The experiment platform is a PC with one Intel i7-6700HQ CPU.

In the simulations, the architecture of DNN-DOA [6] is 56-28-56×6-(37-24-20)×6, and that of DCN-DOA [7] is 2×120-12C25-6C15-3C5-1C3. As illustrated in Fig. 1, our proposed model is composed of 3 spiking blocks, an encoder layer, and a decoder layer. Moreover, the encoder is just a no-parameter transformation method. In spiking block1, channels of input and output are set as 2 and 32, respectively, with the kernel size 31 and padding length 15. For capturing more complex and higher-dimension features, the output channel of spiking block2 increases to 96. At the same time, to obtain the finer-grained knowledge, the kernel size of spiking conv2 is reduced to 25, corresponding to a padding length 12. Then, we set the output channel of spiking block3 as 32, with kernel size 15 and padding length 7. Finally, the decoder layer consists of an ordinary 1D-convolution, whose output channel is 1 and kernel size is 1x1. The learning rate used for our proposed model is set as 0.001, and we adopt the Adam [35] as our model optimizer.

B. MSE Recovery During Training and Validation

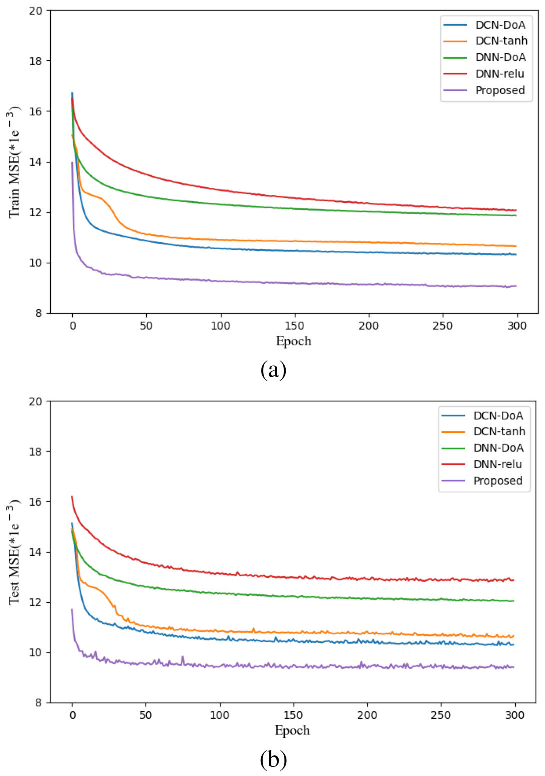
To evaluate the superiority of the proposed method, the convolution networks in Fig. 1(a), (b), (c), and (d) of [7] and the model in [6] are compared. They are shorted as DCN-DoA, DCN-tanh, DNN-relu and DNN-DoA, respectively, and Scnn using original spiking tandem learning.

Fig. 3 displays the training/testing MSE lversus the number of epochs of the proposed method. From Fig. 3, we can see that all approaches converge less than 50 epochs and ours converges faster than other benchmarks.

C. Spatial Spectra and DOA Estimation

We compare our proposal with the model in Fig. 1(a) of [7] and the model in [6], namely DCN-DoA and DNN-DoA, respectively, to evaluate the performance of spatial spectra reconstruction and DOA estimation.

1) Spatial Spectra Reconstruction Estimation: Four groups of two 0 dB narrowband signals are simulated in the far field and made to impinge on the array from two different directions (off the presumed grid). The four groups of signal angles are $[-0.9^o, 1.0^o]$, $[-2.9^o, 3.0^o]$, $[-4.9^o, 5.0^o]$, and $[-8.9^o, 9.0^o]$. Besides, we also test over three and four signals, which



(a)

(b)

Fig. 3. (a) Training and (b) testing MSE of all networks.

are $[-35.9°, -0.9°, 24.0°]$ and $[-24.9°, -4.9°, 20.0°, 45.0°]$ respectively. The reconstructed spectra are averaged by 10 trials, where the true signal locations are indicated by the red squares. The result of this experiment is illustrated in Fig. 4.

In Fig. 4, we can see that both Joint-Scnn, Scnn, and DCN-DoA achieve accurate DOA estimation at true directions, while DNN-DoA deteriorates DOA estimation results with many pseudo peaks, especially when two signals are approaching. Moreover, when the two signals get closer, our proposed Joint-Scnn model still remains better recovery results than others with sharp peaks.

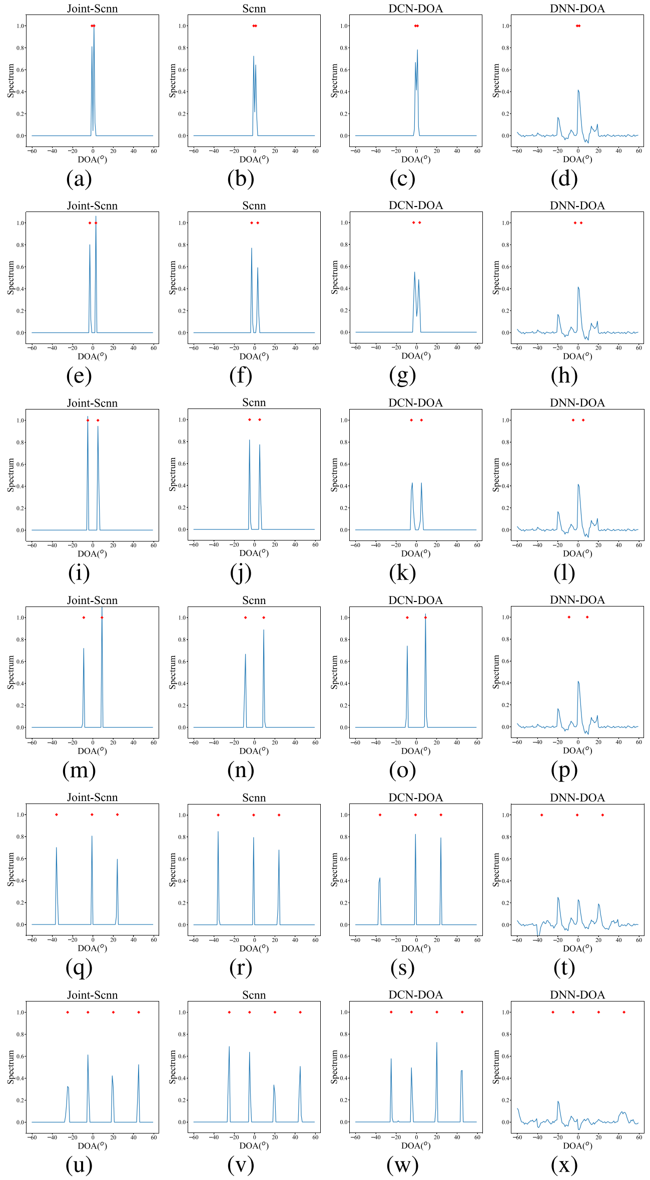
2) DOA Estimation: We further evaluate the effectiveness of the proposed model for DOA estimation via real-time DOA estimates and corresponding errors. The DOA settings are set as $\{5.5^o, 13.5^o, 20.67^o, 50^o, 60^o, 70^o\}$.

As shown in Fig. 5, it is easy to find that Joint-Scnn, Scnn, and DCN-DoA achieve a better performance than DNN-DoA with low estimation error. In addition, we observe that the proposed Joint Scnn outperforms benchmarks with all estimation errors lower than 2^o.

D. RMSE Evaluation Performance

In this subsection, the average root-mean-square error (RMSE) of all incident signals is leveraged to evaluate the statistical performance of our proposed model, Scnn, DCN-DoA and DNN-DoA. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{HK}\sum_{h=1}^{H}\|\hat{\theta}^h - \theta\|^2} \qquad (15)$$



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

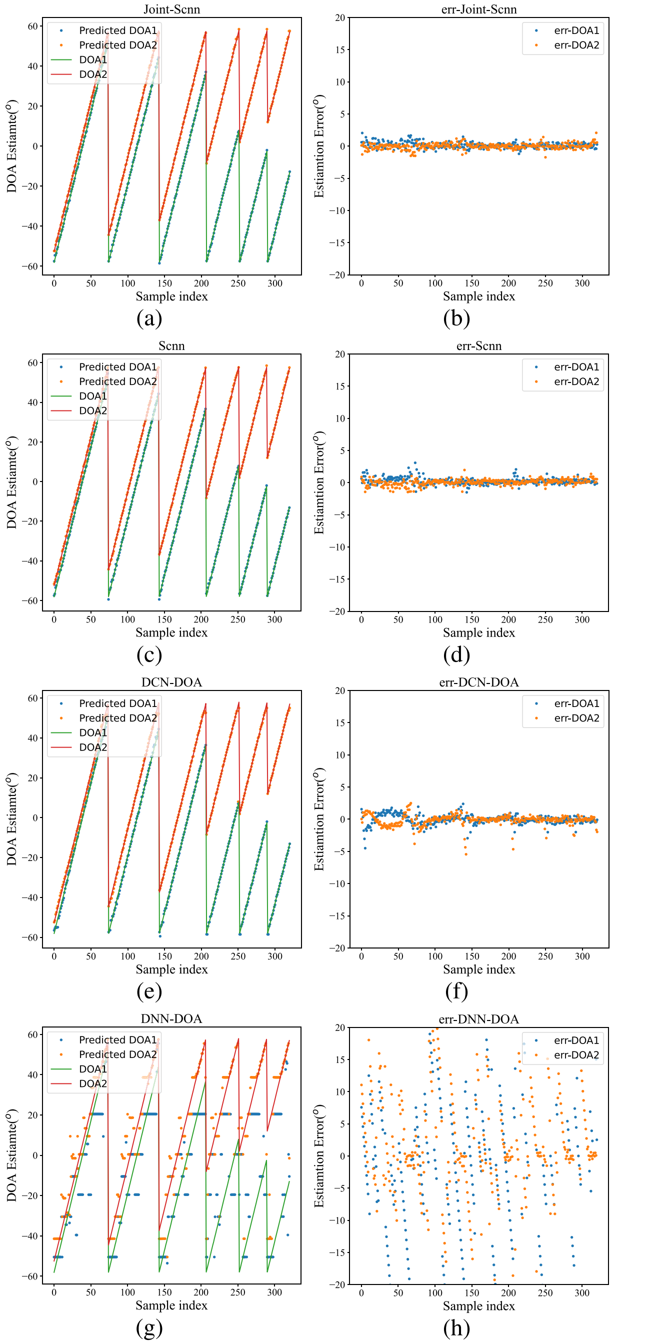
(u) (v) (w) (x)

Fig. 4. Results of reconstructed spatial spectra with two signals. First row: −0.9° and 1.0°. Second row: −2.9° and 3.0°. Third row: −4.9° and 5.0°. Fourth row: −8.9° and 9.0°. Fifth row: −35.9°, −0.9° and 24.0°. Sixth row: −24.9°, −4.9°, 20.0° and 45.0°. (a)(e)(i)(m)(q)(u): Joint-Scnn. (b)(f)(j)(n)(r)(v): Scnn. (c)(g)(k)(o)(s)(w): Methods from [7]. (d)(h)(l)(p)(t)(x): Methods from [6].

where $\hat{\theta}^h$ is the DOA estimate at the hth trial, θ is the ground truth, H is the number of Monte Carlo trials and K is the number of signals. We compare the RMSEs of the proposed Joint-Scnn with those Scnn, DCN-DoA, and DNN-DoA in terms of SNR and angle separation.

1) RMSE vs SNR: Two signals located at $-10.5°$ and $4.5°$ and SNR within [-15 dB, 15dB] are considered, and 1000 independent simulations are carried out at each SNR. As illustrated in Fig. 6(a), ours almost has the lowest RMSE on par with DCN-DoA and Scnn.

2) RMSE vs Angle Separation: Assume that two signals with SNR $= 0$ dB and angle separations within $[2^o, 15^o]$ are considered. For each angle separation $\Delta\phi$, the signal directions



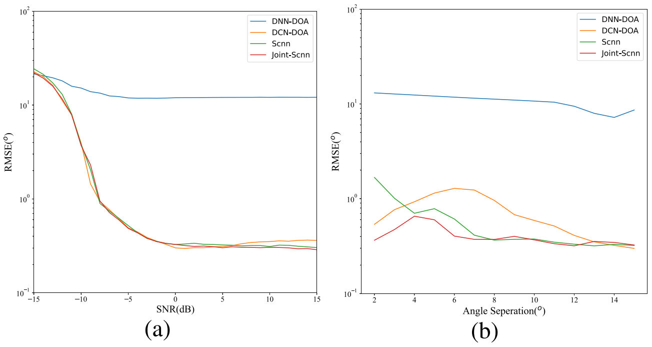
(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. Results of DOA estimates and corresponding estimation errors. (a)(b): Joint-Scnn. (c)(d): Scnn. (e)(f): Methods from [7]. (g)(h): Methods from [6].

are $0^o + \delta$ and $\delta + \Delta\phi$, respectively. δ is a random variable uniformly distributed in $(0^o, 0.2^o)$. Fig. 6(b) shows that our proposed model remains the lowest RMSE with the angle separation increasing.

E. Power Consumption Verification

As known, SNNs can represent the learned information via sparse spikes. Thus, we further illustrate the efficiency of the proposed method with the energy calculation and the heatmaps of three spike blocks' outputs, respectively.



(a) (b)

Fig. 6. Statistic Performance. (a) RMSE vs SNR. (b) RMSE vs Angle Separations.

1) Energy Calculation: The computational cost of this study relies on the total number of Floating-Point Operations (FLOPs), which is analogous to the number of matrix-vector multiplication operations and exhibits a proportional relationship. Note that we focus on the energy calculation of the SNN model in test phase, where every ANN-based module does not work. For per layer l_i, the FLOPs of ANN-based module can be described by [36],

$$FLOPs_{ANN\_Module}(l_i) = \begin{cases} k^2 \times O^2 \times C_{in} \times C_{out}, \text{ if } l_i \text{ denotes the convolutional layer,} \\ C_{in} \times C_{out}, \text{ if } l_i \text{ denotes the linear layer} \end{cases} \tag{16}$$

where k and O correspond to the sizes of kernel function and output feature map, respectively. Besides, C_{in} and C_{out} represent the channel numbers of the input and output, respectively. To compute the FLOPs of spike-based module per spiking layer, the spiking rate $R_s(l_i)$ per spiking layer l_i since SNN consumes energy only when firing spikes, can be defined as:

$$R_s(l_i) = \frac{\#\text{spikes per layer } l_i \text{ over all time steps}}{\#\text{neurons per layer } l_i} \tag{17}$$

i.e., the average firing rate per neuron. Therefore, FLOPs for spike-based module per spiking layer is:

$$FLOPs_{Spike\_Module}(l_i) = FLOPs_{ANN\_Module}(l_i) \times R_s(l_i) \tag{18}$$

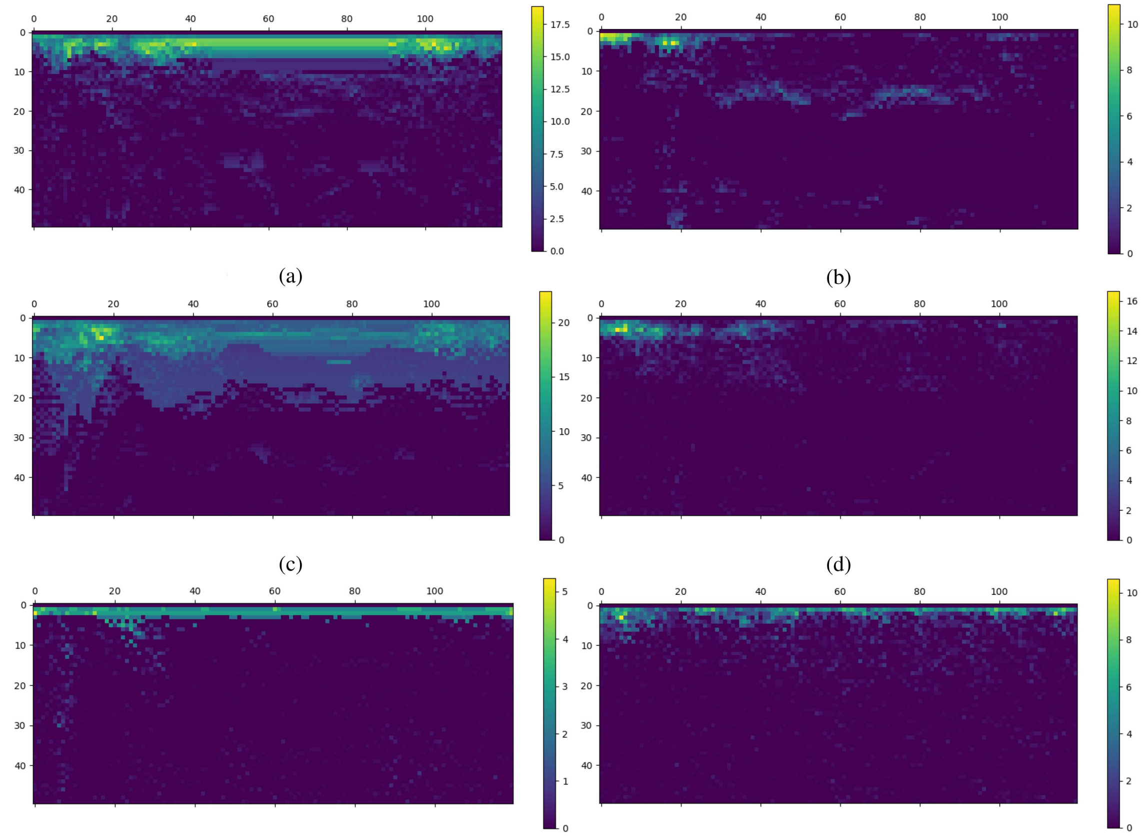
Thus, over all layers, the total inference energy consumption for ANN-based module (E_{ANN_Module}) and spike-based module (E_{Spike_Module}) are calculated as:

$$E_{ANN\_Module} = \sum_{l_i} FLOPs_{ANN\_Module}(l_i) \times E_{MAC} \tag{19}$$

$$E_{Spike\_Module} = \sum_{l_i} FLOPs_{Spike\_Module}(l_i) \times E_{AC} \tag{20}$$

where E_{AC}, E_{MAC} are obtained from a standard 45 nm Complementary Metal–Oxide–Semiconductor (CMOS) process, viz., $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ$ for 32 b FP [37].

Note that, as shown in Table 1, both DNN-DOA and DCN-DOA using simple network architecture perform inferior to ours,



(a) (b) (c) (d) (e) (f)

Fig. 7. Heatmap of output spike in every spiking block (Scnn vs. Joint-Scnn). (a) Spiking block1 in Scnn. (b) Spiking block1 in Joint-Scnn. (c) Spiking block2 in Scnn. (d) Spiking block2 in Joint-Scnn. (e) Spiking block3 in Scnn. (f) Spiking block3 in Joint-Scnn.

TABLE I
ENERGY COMPARISON

Model	**Type**	**Energy**(pJ)
DNN-DoA	real-value-based	0.025E+07
DCN-DoA	real-value-based	0.098E+07
Cnn*	real-value-based	7.45E+07
Scnn	spike-based	2.55E+07
Joint-Scnn	spike-based	1.45E+07

though they have less energy consumption. For fair comparison, we compare with Cnn, Scnn under the same network architecture, and we can see that the proposed Joint-Scnn outperforms them in terms of energy consumption. It is observed that, the energy consumption of Scnn drops by over 65% compared to that of Cnn*. Furthermore, our proposed Joint-Scnn achieves over 81% energy saving as compared to Cnn*.

2) Heatmap Comparison: As well known, the sparsity or firing counts of spike can directly reflect the activation of neurons within a module. Theoretically, it is expected that the effective neural networks activate as small and sparse part of neurons as possible in inference phase, which indicates the low energy consumption. To this end, the energy consumed by our Joint-Scnn can be presented by the spike firing heatmap at the output layer of its three spike blocks. To better visually display, we use the spike-based visual tool [38] to compute and present the heatmap. The results of heatmaps are illustrated in Fig. 7. For each heatmap, its width equals 120 representing the length of the 1D output features in every spiking block, and its height is 50 representing the length of the time window. In a heatmap, the color depth of a position denotes the activation of the feature at a specific time. When the color turns more yellow, it means that the activation of neurons at this feature position is higher over the specific time step. It should be noted that the number of yellow or green positions of a feature in the time dimension is equal to the spike count of this feature over the whole time window.

It can be observed that the spike counts of neurons in our Joint-Scnn mainly locate within the smaller time window [1,10], less than that of Scnn. It indicates that our proposed model mainly requires only 10 time steps to represent features in every spiking block via binary spikes (0 or 1) instead of real-value values. Besides, from the dimension of feature length, the number of activated neurons in our Joint-Scnn is obviously much less than Scnn. Especially on the output of the first and second block, our Joint-Scnn just activates less than half the neurons. In contrast, almost all of neurons in Scnn are in active state.

Inherently, at the begining, the neurons on spike-based module only enjoy simple addition operation until the accumulated values bypass the threshold, which can be seen from the computation of $R_s(l)$ in (17). Further, the sparse fired spikes (demonstrated in Fig. 7), as tensors composed of 0 or 1 binary values, reduce the addition operation by a large margin. Finally, our proposed approximation of spike train by real-value spike count in the whole training process successfully reduces the high cost as the traditional SNNs. That is also the reason why the proposed method enjoys lower energy consumption as compared to Scnn.

V. Conclusion

Consider the limitations of the data-driven ANN-based approaches for the DOA estimation task, we propose an event-driven spiking neural network (SNN) model via ANN-SNN weight sharing scheme. Herein, a novel spiking tandem learning framework is proposed, named Joint-Scnn, to increase the whole performance via ANN self-learning. Finally, extensive experiments demonstrate the efficiency of our proposed model over other approaches in terms of estimation performance and energy consumption.

Acknowledgment

The authors would like to thank the authors of [7] for sharing their codes online, which helps us to compare more conveniently.

References

[1] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[2] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[3] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[4] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.

[5] H.L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, NY, USA: Wiley, 2002.

[6] Z.-M. Liu, C. Zhang, and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec. 2018.

[7] L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Process. Lett.*, vol. 26, pp. 1688–1692, Nov. 2019.

[8] G.K. Papageorgiou and M. Sellathurai, and Y.C. Eldar, "Deep networks for direction-of-arrival estimation in low SNR," *IEEE Trans. Signal Process.*, vol. 69, pp. 3714–3729, 2021.

[9] R. Akter, V.S. Doan, T. Huynh-The, and D.S. Kim, "RFDOA-Net: An efficient ConvNet for RF-Based DOA estimation in UAV surveillance systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12209–12214, Nov. 2021.

[10] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[11] A. M. Elbir, "DeepMUSIC: Multiple signal classification via deep learning," *IEEE Sensors Lett.*, vol. 4, no. 4, Apr. 2020, Art. no. 7001004.

[12] H. Huang, Q. Liu, H. C. So, and A. M. Zoubir, "Low-rank and row-sparse decomposition for joint DOA estimation and distorted sensor detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4763–4773, Aug. 2023.

[13] Q. Liu, Y. Gu, and H. C. So, "DOA estimation in impulsive noise via low-rank matrix approximation and weakly convex optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 3603–3616, Dec. 2019.

[14] Y. Ma, Y. Zeng, and S. Sun, "A deep learning based super resolution doa estimator with single snapshot MIMO radar data," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4142–4155, Apr. 2022.

[15] S. Xu, A. Brighente, B. Chen, M. Conti, X. Cheng, and D. Zhu, "Deep neural networks for direction of arrival estimation of multiple targets with sparse prior for line-of-sight scenarios," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 4683–4696, Apr. 2023.

[16] R. Akter, V.-S. Doan, T. Huynh-The, and D.-S. Kim, "RFDOA-Net: An efficient convnet for RF-based doa estimation in UAV surveillance systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12209–12214, Nov. 2021.

[17] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1311–1318.

[18] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 5, pp. 54–66, 2015.

[19] S. B. Shrestha and G. Orchard, "SLAYER: Spike layer error reassignment in time," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1412–1421.

[20] R. Xiao, Q. Yu, R. Yan, and H. Tang, "Fast and accurate classification with a multi-spike learning algorithm for spiking neurons," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1445–1451.

[21] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Netw.*, vol. 99, no. 5, pp. 56–67, 2018.

[22] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, and T. Masquelier, "Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks," *Pattern Recognit.*, vol. 94, pp. 87–95, 2019.

[23] A. Tavanaei and A. Maida, "A spiking network that learns to extract spike signatures from speech signals," *Neurocomputing*, vol. 240, no. 02, pp. 191–199, 2017.

[24] Z. Zhang and Q. Liu, "Spike-event-driven deep spiking neural network with temporal encoding," *IEEE Signal Process. Lett.*, vol. 28, pp. 484–488, 2021.

[25] Q. Liu and Z. Zhang, "Ultra-low power always-on intelligent and connected SNN-Based system for multimedia IoT-Enabled applications," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15570–15577, Sep. 2022.

[26] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2015, pp. 1–8.

[27] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Front. Neurosci.*, vol. 13, 2019, Art. no. 95.

[28] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Front. Neurosci.*, vol. 11, 2017, Art. no. 682.

[29] J. Wu, Y. Chua, M. Zhang, G. Li, H. Li, and K. C. Tan, "A tandem learning rule for effective training and rapid inference of deep spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 446–460, Jan. 2023.

[30] M. Pastorino and A. Randazzo, "A smart antenna system for direction of arrival estimation based on a support vector regression," *IEEE Trans. Antennas Propag.*, vol. 53, no. 7, pp. 2161–2168, Jul. 2005.

[31] A. El Zooghby, C. Christodoulou, and M. Georgiopoulos, "A neural network-based smart antenna for multiple source tracking," *IEEE Trans. Antennas Propag.*, vol. 48, no. 5, pp. 768–776, May 2000.

[32] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking neurons," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 787–797.

[33] R. Ronan and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[35] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014, pp. 1–15.

[36] Y. Kim and P. Panda, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," *Front. Neurosci.*, vol. 15, 2020, Art. no. 773954.

[37] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2014, pp. 10–14.

[38] Y. Kim and P. Panda, "Visual explanations from spiking neural networks using inter-spike intervals," *Sci. Rep.*, vol. 11, no. 1, 2021, Art. no. 19037.