For the efficient implementation, we take another form of STDP using eligibility traces Lee et al. (2018a); Izhikevich (2007); Zenke et al. (2015). As shown in the Equation 4 , $x_{trace}^{(t)}$ accumulates presynaptic spikes and gradually decays over time.

$$\begin{aligned} &\Delta w_j = x_{trace}^{(t)} - x_{offset} \quad if \;\; x_{postspike} = 1 \\ &{x_{trace}}^{(t)} = \lambda_+ x_{trace}^{(t-1)} + x_{prespike} \end{aligned} \tag{4}$$

When the neuron fires at time t,$x_{postspike} = 1$. When recieves a spike at time t, $x_{prespike} = 1$. We denote $\lambda_+ = 1 - \frac{1}{\tau +}$.

In experiments, we improve STDP and propose a sample-temporal batch STDP (STB-STDP) algorithm. More details are in section 4.4.

4. Proposed Algorithms

To be more explicit about the problem we focus on, we built a one-layer convolutional spiking neural network and used the STDP algorithm, described in Equation 4 , to train in MNIST dataset until convergence. This convolutional layer consists of a 5x5 convolution kernel with 20 channels. As shown in Figure 2(a), we show the synaptic weights of the convolutional layers after training. It can be observed that a large number of repeated convolution kernels appear. The same convolution kernel features are boxed in the same color. Repeated convolution kernels will affect the effective feature representation Glorot and Bengio (2010).

On the other hand, neurons from different layers may also work in a disordered way. Since the algorithm lacks a global guided signal, the neuron can not judge whether it is a suitable firing rate. Therefore This will cause the adjacent next layer of neurons to receive too high or too low input current, which makes the last layer's neurons fire unstably. As shown in Figure 2(b), The membrane potential will take longer to build up if the input current is too small, leading to delays in the transmission of information. When the firing frequency is significant, the neuron fires nearly all the time, which will damage the effective information representation. Different parameter settings can easily lead to disordered spikes, making it challenging to transfer useful information.

To alleviate the above problems, we propose three adaptive algorithms. To address the problem of repetition of neuron features within a layer, we

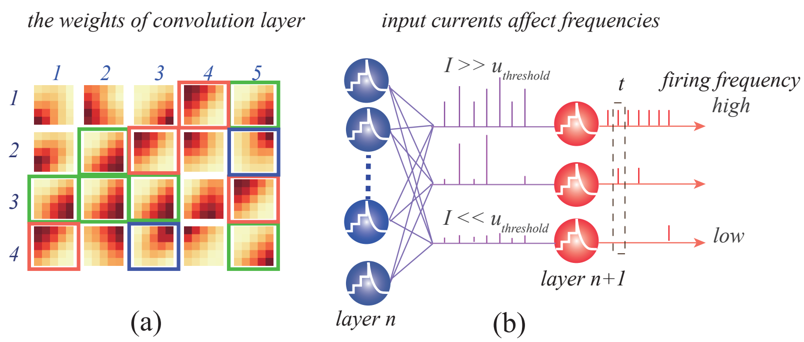

Figure 2: (a) Features of the convolutional layers trained with only STDP, the same features are marked with the same color. (b) SNNs trained only with STDP cannot properly regulate the information transmitted by the spikes, which will lead to sparse or frequent spikes.

propose an Adaptive Lateral Inhibitory Connection (ALIC), different from static lateral inhibition Diehl and Cook (2015), which provides a way to coordinate neurons by automatically selecting those that need to be inhibited. Next, we use Adaptive Threshold Balance (ATB) to solve the mismatching of input and output ranges between adjacent layers. In order to make the spikes firing more stable, we designed Adaptive Synaptic Filter (ASF) inspired by STP. Finally, we propose STB-STDP, which combines spatial and temporal information into a single batch.

4.1. Adaptive Lateral Inhibitory Connection

We introduce lateral inhibition to solve the problem of neurons in the same layer tending to have the same weights. Interaction between neurons is enabled by lateral inhibition, which allows firing neurons to dominate this input by inhibiting the firing of other neurons. The dominant neuron is more likely to experience learning according to STDP, and sensitivity to its input gradually increases. Conversely, non-dominant neurons will not be sensitive to this input. This prevents neurons from convergent towards the same weight. static lateral inhibition, however, is usually a constant set manually.

$$inh_{ij}^{(t)} = \alpha \tag{5}$$

where $inh_{ij}^{(t)}$ denotes the inhibition received by the neuron i, j. α is a parameter defining the degree of inhibition. The same inhibiting degree is maintained across all neurons and inputs. As a part of the network, the lateral inhibitory connections are static, with the same weights assigned to each neuron. However, this is not reasonable. Evidence from neuroscience suggests that for better inhibition, it does not exert inhibition on all neurons, but on those with relevant activity Linster et al. (2005); Kuffler (1953); Arevian et al. (2008). Thus, we expect a dynamic structure of lateral inhibitory connections to produce various structures for different inputs. Such a dynamic structure would help enhance coordination between neurons.

To this end, we introduce Adaptive Lateral Inhibitory Connection (ALIC), as shown in Figure 3 . We designed a dynamic structure of lateral inhibition. we determine which neurons may fire by setting a threshold and inhibit the membrane potential for these neurons. As shown in the Equation 6, we choose the maximum input current as the reference. Inhibition depends on maximum current for different inputs. Threshold is set with $i_{thresh} = \frac{i_{max}^{(t)}}{2}$

$$\begin{aligned} inh_{ij}^{(t)} &= \alpha_{inh} \; (\max_{b,c,w,h} i^{(t)})(1 - \hat{s}_{ij}^{(t)}) \\ &= \alpha_{inh} \; (\max_{b,c,w,h} \sum_{i,j} w_{ij}^{(t)} \hat{s}_{ij}^{(t)})(1 - \hat{s}_{ij}^{(t)}) \quad if \quad i_{ij}^{(t)} > i_{thresh} \end{aligned} \tag{6}$$

where $inh_{ij}^{(t)}$ is the inhibition for the neuron at the position of i, j at the time t. α_{inh} is a coefficient that adjusts the degree of inhibition. $i^{(t)}$ denotes the input current, which obtained by synaptic weight w_{ij} and spikes $\hat{s}_{ij}^{(t)}$. The maximum is selected from all the neurons in batch, where b denotes batch, c, w, h denotes the channel and size of output. $(1 - s_{ij}^{(t)})$ allows the inhibition to act only on neurons that are not firing at this moment. We adopt the winner-take-all strategy, randomly take out a firing neuron, and set the remaining spikes to 0, $\hat{s}_{ij}^{(t)} = wintakeall(s_{ij}^{(t)})$. A more detailed analysis of section 5.3 shows that ALIC actually improves the performance of the network.

4.2. Adaptive Threshold Balance

By introducing an adaptive threshold method, we are able to eliminate the mismatch between input and output between layers. The input current varies from different inputs, leading to spikes firing variability. The current may be too small to reach the threshold, delaying the transmission of information and increasing the network delay. A large value may also be well above