the samples, the convergence speed of the model is also significantly increased. In addition, SNNs exhibit multiple forward propagation processes in the temporal dimension, in contrast to artificial neural networks. By aggregating multiple time steps, the network can converge more smoothly instead of updating at every step.

From this, we propose a sample-temporal batches STDP algorithm (STB-STDP). A weight update is done by integrating the information of samples and temporal information simultaneously. As the Equation 10 shows:

$$\begin{aligned}\Delta w_j &= \sum_{n=0}^{N_{batch}} \sum_{t=0}^{T_{batch}} (x_{trace}^{n,t} - x_{offset})/(N_{batch}T_{batch}) \quad if \;\; x_{postspike} = 1 \\ x_{trace}^{n,t} &= \lambda_{+} x_{trace}^{n,t-1} + x_{prespike}\end{aligned} \tag{10}$$

where $N_{batch}$is the batchsize of the input, T_{batch} is the batchsize of time step.

Weights are normalized after updating, preventing them from diverging or shifting. As shown in Equation 11. Different normalization methods are used for convolutional layers and fully connected layers. A_{fc}^{-}, A_{conv}^{-} is scale factor to control normalization. Additionally, a spike normalization module is added between each layer to stabilize the inputs and outputs, which limits the input range of the spike. This module captures spikes and outputs a firing frequency of up to one.

$$fc: w_j^{(t)} = A_{fc}^{-} \frac{w_j^{(t)}}{mean(w_j^{(t)})} \quad conv: w_{ij}^{(t)} = A_{conv}^{-} \frac{w_{ij}^{(t)} - mean(w_{ij}^{(t)})}{std(w_{ij}^{(t)})} \tag{11}$$

5. Experiments

To demonstrate the effectiveness of our model, we conduct experiments on the commonly used datasets, MNIST LeCun et al. (1998) and FashionMNIST Xiao et al. (2017), CIFAR10 Krizhevsky et al. (2009). All the experiments are based on the structure consisting of a convolutional layer followed by a 2*2 max pooling layer, a spiking normalization layer, and a fully connected layer. Since our model is an unsupervised network, we adopt the same voting strategy as in Diehl and Cook (2015) of the output of the final layer for category prediction. And the parameters of the network are trained layer-wisely. Where $A^-_{fc} = 0.01, A^-_{conv} = 1$,$\alpha_{inh} = 1.625$, $\theta_{init} = 10$, $\alpha_{sfa} = 0.4, \beta_{sfa} = 8, \alpha_{plus} = 0.001, \lambda = 0.99$, $\beta_{thresh} = 1$, $x_{offset} = 0.3$.

5.1. Experimental Results

MNIST is a digital handwriting recognition dataset widely used as a benchmark for evaluating model performance in recognition and classification tasks. The dataset contains a total of 60,000 training set samples and 10,000 test set samples. The size of each sample is 28*28 pixels. In the experiments, the examples in MNIST were normalized using direct encoding and without any form of data augmentation and preprocessing. For the MNIST dataset, we set the kernel size of the convolutional layer to 5 with 12 channels, and 6400 neurons in the fully connected layer. Timestep is 300 in our experiments. To verify the superiority of our model, we compare the results with other famous STDP-based SNN models. Un-&Supervised denotes the former layer is trained unsupervised, while the final decision layer is trained with supervised information. As shown in Table 1, Our model achieves 97.9% accuracy. Compared with Diehl and Cook (2015), which only uses the STDP, our model improves by nearly 3%. Our model has surpassed all the unsupervised STDP-based SNNs and even some SNNs with supervised information.

Table 1: The performance on MNIST dataset compared with other STDP-based SNNs.

Model	Learning Method	Type	Accuracy
Querlioz et al. (2013)	STDP	Unsupervised	93.5%
Diehl and Cook (2015)	STDP	Unsupervised	95.0%
Hao et al. (2020)	Sym-STDP + SVM	Un-&Supervised	96.7%
DCSNN Mozafari et al. (2019)	DoG + STDP + R-STDP	Un-&Supervised	97.2%
Falez et al. (2019)	DoG + STDP + SVM	Un-&Supervised	98.6%
SDNN Kheradpisheh et al. (2018)	DoG + STDP + SVM	Un-&Supervised	98.4%
SpiCNNLee et al. (2018b)	LoG+STDP	Un-&Supervised	91.1%
Tavanaei and Maida (2017)	STDP + SVM	Un-&Supervised	98.4%
Ferré et al. (2018)	STDP + BP	Un-&Supervised	98.5%
SSTDPLiu et al. (2021)	STDP + BP	Supervised	98.1%
VPSNN Zhang et al. (2018)	Equilibrium Propagation + STDP	Supervised	98.5%
CBSNN Shi et al. (2020)	VPSNN + Curiosity	Supervised	98.6%
BP-STDP Tavanaei and Maida (2019)	STDP-Based BP	Supervised	97.2%
GLSNN Zhao et al. (2020)	Global Feedback + STDP	Supervised	98.6%
Ours	ASF + ATB + ALIC + STB-STDP	Unsupervised	97.9%

FashionMNIST is more complex than MNIST within cloths and shoes as the samples. Both the shape and the size of the data are the same as those obtained by MNIST. The kernel size for the convolutional layer is set at 3 with 64 channels, and 6400 neurons in the fully connected layer. The timestep

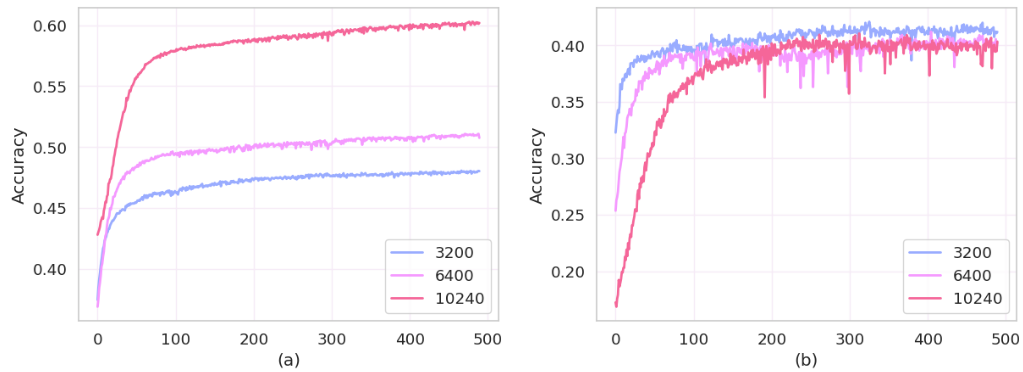

Figure 4: The accuracy curve of the model on cifar10. We conducted experiments with 3200, 6400, and 10240 neurons in the fully connected layer. We visualize (a) the accuracy curve for the training set and (b) the accuracy curve for the test set.

is the same as that used in MNIST. As seen in Table 2, our model succeeded in achieving 87.0% accuracy and outperformed most STDP-based SNNs. Although the performance of GLSNN is higher than ours, it introduces global supervised connections, while our network has no supervision information.

Table 2: The performance on FashionMNIST dataset compared with other STDP-based SNNs.

Model	Learning Method	Type	Accuracy
FSpiNN Putra and Shafique (2020)	STDP	Unsupervised	68.8%
Rastogi et al. (2021)	A-STDP	Unsupervised	75.9%
Hao et al. (2020)	Sym-STDP	Supervised	85.3%
VPSNN Zhang et al. (2018)	Equilibrium Propagation + STDP	Supervised	83.0%
CBSNN Shi et al. (2020)	VPSNN + Curiosity	Supervised	85.7%
GLSNN Zhao et al. (2020)	Global Feedback + STDP	Supervised	89.1%
Ours	ASF + ATB + ALIC + STB-STDP	Unsupervised	87.0%

5.2. Result on CIFAR10

Only a few STDP-based SNN algorithms are capable of classifying the FashionMNIST dataset and achieving impressive results. As far as we know, few unsupervised STDP-based models run experiments on more complex datasets, such as CIFAR10. Trying to maintain high performance on more complex datasets will be our exploration direction. First, we used RGB data and the image size is 32x32. We use a 5x5 convolution kernel with 3 input channels and 64 output channels in the convolutional layer. Meanwhile, 3200