neurons are used in the fully connected layer. After training, our model achieved 42.36% accuracy on the CIFAR10 dataset. We visualize the accuracy curves of our model with 3200, 6400, and 10240 neurons in the fully connected layer. The highest accuracy was obtained for the test set at the number of neurons of 3200.

5.3. Discussion

In order to determine what causes misclassified samples, we visualized the classification confusion matrix of our model on the MNIST and FashionMNIST datasets, respectively. As shown in the Figure 5. There is not much distinction between the categories on MNIST's matrix. However, FashionMNIST's matrix showed lower accuracy on four categories of 0,2,4,6, corresponding to 't-shirt', 'pullover', 'coat', and 'shirt', respectively. There is a high degree of similarity between images in these four categories, which makes it more difficult for the model to classify them accurately.

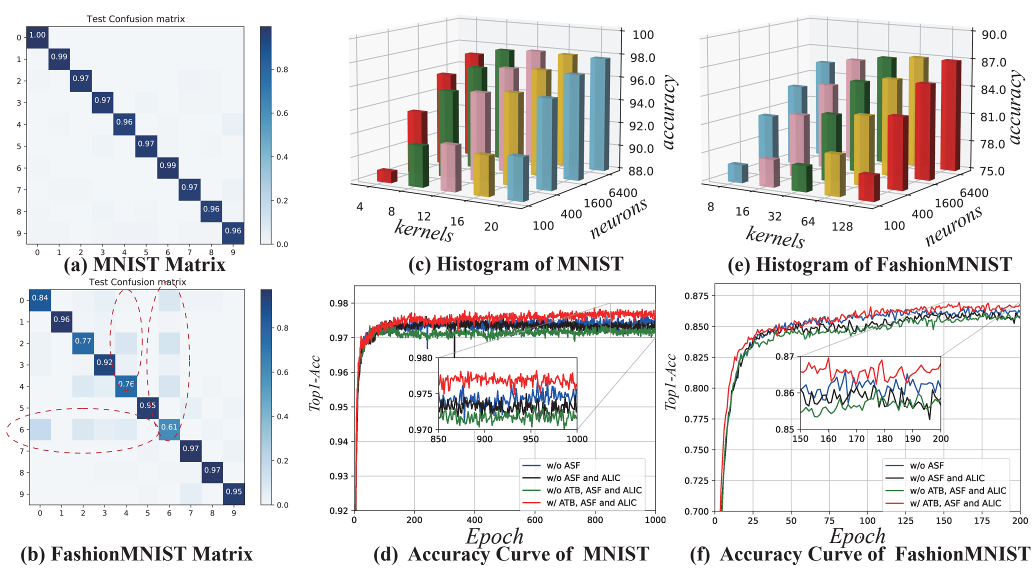

Figure 5: (a,b) The confusion matrix of our model in MNIST and FashionMNIST. (c,e) The histograms of the performance of our model on MNIST and FahsionMNIST with the different numbers of convolutional kernels and voting neurons. (d,f) The test accuracy curves of our model with and without the modules we design.

We explore the impact of different parameter settings on network performance. In convolutional layers, we changed the number of kernels, while in fully connected layers, we changed the number of neurons. As shown in Figure 5. The different number of channels and the number of neurons in the fully connected layer are combined, respectively. The height of the histogram represents the final accuracy of the model. There is a direct correlation between performance and the number of voting neurons set. Because more voting neurons will provide more predictions, combining all predictions, the network will produce more accurate results. However, this correlation does not exist for the setting of the number of convolution kernels. The best performance for MNIST is achieved when the number of kernels is set to 12.

In contrast, FashionMNIST requires more convolution kernels due to its complexity, and it works best when the number of kernels is set to 64. More convolution kernels cannot extract more features, because repeated features will appear in large numbers. In addition, the fully connected layer also needs more connections due to the increased number of kernels because more connections increase the complexity of the network and the difficulty of convergence.

To fully illustrate the impact of each module of our model on the results, we conduct ablation experiments on two datasets separately. We conducted experiments with the settings "w/o ASF", "w/o ASF and ALIC", and "w/o ASF, ALIC, and ATB", respectively. "w/o" denotes that we remove the relevant module. We show the convergence curve of the model on the test set, as shown in Figure 5. All experiments included STB-STDP to ensure the efficiency of the experiment. It would take a much longer time for the model to run without it. Adding each module can assist in improving the model's learning ability and its ability to achieve higher accuracy on the MNIST and FashionMNIST datasets.

To better illustrate the role of lateral inhibition, we added "softmax" for comparison. "softmax" is a commonly used function that can serve as a lateral inhibition. For the experiments, we removed the lateral inhibition module and performed the experiments on the MNIST dataset with the same model structure. The convolutional layer contains 12 5x5 convolutional kernels and 6400 neurons in the fully connected layer. We have trained the model with the same experimental setup. The model reached 28.26%. Then we added the softmax layer after the input currents in the convolutional layer and the fully connected layer. Also, to ensure that the currents are of the same scale, we multiplied by the maximum value of the current. After training, the model obtained 78.16%. This shows that the softmax has a certain effect. However, lateral inhibition can help the model to achieve the inhibition in a better way.

5.4. Experiment for Small Samples

Table 3: The performance of our model compared with ANN on MNIST dataset with different training samples.

samples	200	100	50	10
ANN	79.77%	71.40%	68.72%	47.12%
Ours	81.45%	75.44%	72.88%	51.45%
	1.68%	4.04%	4.16%	4.33%

In contrast to the backpropagation algorithm, our network trained with the STDP unsupervised algorithm is more adaptable to different number of samples, especially in small-sample task, and shows superior performance compared to supervised algorithms with the same structure. There are only a very small number of samples in the whole train set in small-samples task, which differs from few-shot learning. To this end, to illustrate the ability of our model to train on small samples tasks, a very small number of samples are randomly selected from the MNIST dataset. The number of samples per class ranges from 20 to 10 to 5, in the most extreme case, to 1. At the same time, we also designed an artificial neural network model with the same structure for comparison. The model consists of 5x5 convolutions with 12 channels, a maxpool layer, a fully connected layer of 6400 neurons, and a final fully connected classifier. The backpropagation algorithm is used to train this ANN model.

As shown in the Table 3, our model has better performance than the ANN model. As the number of training samples gradually decreases, the performance gap between our model and ANN gradually increases. When there is only 1 training sample per class, our model outperforms ANN by 4.43%. It fully illustrates that our model requires only a small number of samples to achieve high performance compared to artificial neural networks that require a large amount of labeled training data.

5.5. Visualization

To illustrate the feature extraction capability of our model, we visualize the weights of different layers. Figure 6 a and b shows the weight of the convolutional layer on the MNIST and FashionMNIST dataset respectively.