N-Omniglot: a Large-scale Neuromorphic Dataset for Spatio-Temporal Sparse Few-shot Learning

Yang Li\textsuperscript{1,2,\dag}, Yiting Dong\textsuperscript{1,3,\dag}, Dongcheng Zhao\textsuperscript{1 \dag}, and Yi Zeng\textsuperscript{1,2,3,4,5}

\textsuperscript{1}Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing, China
\textsuperscript{2}School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
\textsuperscript{3}School of Future Technology, University of Chinese Academy of Sciences, Beijing, China
\textsuperscript{4}National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
\textsuperscript{5}Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China

*Email: liyang2019@ia.ac.cn, dongyiting2020@ia.ac.cn, zhaodongcheng2016@ia.ac.cn, yi.zeng@ia.ac.cn
\dag These authors contributed equally to this work

ABSTRACT

Few-shot learning (learning with a few samples) is one of the most important capacities of the human brain. However, the current artificial intelligence systems meet difficulties in achieving this ability, so as the biologically plausible spiking neural networks (SNNs). Datasets for traditional few-shot learning domains provide few amounts of temporal information. And the absence of the neuromorphic datasets has hindered the development of few-shot learning for SNNs. Here, we provide the first neuromorphic dataset: N-Omniglot, using the Dynamic Vision Sensor (DVS). It contains 1623 categories of handwritten characters, with only 20 samples per class. N-Omniglot eliminates the need for a neuromorphic dataset for SNNs with high sparseness and tremendous temporal coherence. Additionally, the dataset provides a powerful challenge and a suitable benchmark for developing SNNs algorithm in the few-shot learning domain due to the chronological information of strokes. We also provide the improved nearest neighbor, convolutional network, SiameseNet, and meta-learning algorithm in spiking version for verification.

Background & Summary

In recent years, the large scale datasets and increased computing power make machine learning, especially deep learning, reach the level of human-like performance in many areas\textsuperscript{1–3}. However, compared with the human brain, artificial neural networks(ANN) lack of biological characteristic and interpretability, for its floating-point-based calculation and gradient-based algorithm\textsuperscript{4}. Combining computer technology and computational neuroscience related knowledge can effectively improve the current deep learning technology. Spiking neural networks (SNNs) are considered to be the third generation of artificial neural networks\textsuperscript{5}, by simulating the similar calculations and representations in the human brain, which shows strong biological interpretability. Only neurons that fire spikes will participate in the calculation of the network. Meanwhile, the sparse spike activity greatly reduces the network’s energy consumption\textsuperscript{6}. However, the lack of datasets for SNNs burden the development of SNN algorithm.

The success of deep learning can largely attribute to the introduction of datasets such as ImageNet\textsuperscript{7} and COCO\textsuperscript{8}. However, the currently widely used datasets are not suitable for SNNs. SNNs need to encode the static data into spike trains and then put them into the network\textsuperscript{9}. As a result, the information will be missing, and it will be not fair to compare with the artificial neural networks. Dynamic Vision Sensor (DVS)\textsuperscript{10} is a new neuromorphic camera. DVS only generates 0/1 events on pixels with different light intensities to achieve low latency, low redundancy, and high time resolution, which is different from the
frame-based cameras. In addition, DVS simulates the human visual nervous system in principle so that SNNs can make full use of the temporal information provided by such sensors.

To promote the development of SNNs, researchers used DVS to provide many neuromorphic datasets. N-MNIST\textsuperscript{11}, N-caltech101, DVS-CIFAR10\textsuperscript{12} are obtained through the picture or camera jitter, captured from traditional classification datasets, avoiding the damage to original image information during the encoding process. In addition to the conversion of static images, more researchers tend to obtain event data from natural environments, such as DVS-Gesture\textsuperscript{13}, N-Cars\textsuperscript{14}, etc. But the existed datasets, such as those mentioned above, have very low temporal correlation and cannot fully reflect the temporal information processing capabilities of SNN. All the characteristics are shown in Table 1. In addition to sparse coding\textsuperscript{15} to reduce energy consumption, learning new concepts rapidly from a few samples is also one of the important capabilities of the human brains. While it is an open problem in spike-based machine learning. The few-shot learning\textsuperscript{16,17} imposes tremendous challenges to the current learning methodologies of SNNs due to the lack of neuromorphic datasets\textsuperscript{18} for training and evaluating the learning ability of a few samples.

| Dataset            | # of classes | Sparsity | Difference | Object  |
|--------------------|--------------|----------|------------|---------|
| N-MNIST            | 10           | Low      | Low        | Image   |
| N-Caltech101       | 100          | Low      | Low        | Image   |
| DVS-CIFAR10        | 10           | Low      | Low        | Image   |
| N-Cars             | 2            | Low      | Mid        | Gesture |
| DVS-Gesture        | 11           | Low      | Mid        | Cars    |
| N-Omniglot         | 1623         | High     | High       | Stroke  |

Table 1. The Characteristics of the neuromorphic datasets.

To tackle the problems and fulfill this gap, we propose the first neuromorphic dataset for few-shot learning: N-Omniglot. The original Omniglot dataset\textsuperscript{19} is the most commonly used dataset in the field of few-shot learning. It consists of 1623 handwritten characters from 50 different languages. Each character has only 20 different samples. It is usually recognized as a static character image, while the rich temporal information of the writing process is ignored. Therefore, we reconstruct the

![Figure 1. Some examples of N-Omniglot. High temporal correlation and spatio-temporal sparsity can be seen in these examples.](image-url)
writing process of strokes and use DVS to obtain the event records to get the neuromorphic version of Omniglot (N-Omniglot). We provide several improved classic few-shot learning algorithms to adapt to SNN, showing that N-Omniglot diverses in time-dimension, providing more temporal information, and support many tasks. We hope it can provide a benchmark for SNN-based few-shot learning and provide a competitive environment for the research community to promote SNN’s temporal and spatial feature extraction ability and sparse representation learning.

**Methods**

In this work, we first use the stoke temporal information to reconstruct the writing process of Omniglot into videos. For the convenience of capturing, we merge the writing strokes of 20 samples of each character into a video file, and a blank sequence is inserted between each sample. Second, we use the DVS acquisition platform to shoot videos which are played on the monitor, and use the Robotic Process Automation (RPA) software to automatically collect the data. Finally, the corresponding sample data will be split out. Figure 2 shows the entire construction process of the dataset.

![Figure 2.](image)

**Figure 2.** Complete process of data generation. Phase A constructs the video for recording, including a1: preprocessing the original data and a2: reconstructing the video. Phase B is the actual capture stage, including b1: building the equipment environment, and b2: recording with the RPA software. Phase C performs post-processing, including c1: labeling the beginning and end of the characters, and c2: segmentation using time stamp.

**Stroke preprocessing and reconstruction**

Each image in Omniglot has a corresponding stroke data in milliseconds. In order to present the entire writing process to the DVS, we first reconstruct the text record of strokes as a video of writing tracks. Also, because of the difference in acquisition equipment and writing habits, we delete the interval generated when each stroke is written. The linear interpolation algorithm is used to complete the data in milliseconds to reconstruct the character writing as accurately as possible. Due to the inconstant frequency of sampling and the jitter during the writing process, some strokes have only one or a few points, and the refresh rate of the display screen is 60Hz, so strokes less than 17ms in the reconstructed video may not be displayed. Therefore, we linearly interpolate them to a sufficiently long length, here we use 34ms.

**Automated capture using davis346**

We use Davis346 as our acquisition device due to its good time resolution. We design a black box to cover the screen and DVS camera to prevent external light changes from interfering with the experimental data collection. In the experiment, the DVS...
software is used to process the captured event data. We set the background activity time parameter in the DV software to 4000 to better filter out the background noise from the input, such as the low-frequency noise displayed on the monitor. As well as, the exposure parameter is fixed to 8000 to keep the brightness stable. In order to avoid frequent software operations that cause major changes in the relative position of the device, we use the Robotic Process Automation (RPA) software UiBot to automatically collect and record data. As shown in the Figure 2, we first read the address of the reconstructed stroke video. During the event conversion process, the record and stop buttons are pressed at the beginning and end of the video playback. Finally, the recorded .aedat4 files are saved to the present directory.

**Segmentation and preprocess for usage**

Each character in Omniglot contains 20 samples. To avoid unnecessary software operations and make the collected DVS event data more stable and efficient, we combine the 20 reconstructed videos into a long one, with a 500ms gap between each sample, the beginning and the end of the video. After converting the stroke video to the event file, we separate the events corresponding to each sample. Specifically, when the events change from sparse spatial-temporal property to concentrated in a certain position for a long time, the sample event starts. And we use the frame number of the original video to assist in finding the corresponding time node to determine the end of the sample event. We save the event data with the form of (x, y, t, p), where the first two items x, y are the pixel coordinates of the event, the third item t is the timestamp of the event, and the fourth item p is the polarity with value 1 and 0 indicates the increase or decrease of brightness separately. In our paper, considering that SNN is always clock-driven at runtime and cannot perform asynchronous calculations on all neuron units like FPGAs, so the event data will be processed into image data to input into the network. We process the events within a period of time into an image with a resolution of DVS346 (346*260). The two polarities are represented by two image channels, and the pixels without events are filled with 0. We mainly used two methods: the OR operation and the firing rates, the details will be shown in the technical validation section.

| Statistic | Mean  | Std  | Statistic | Mean  | Std  |
|-----------|-------|------|-----------|-------|------|
| X-max     | 321.95| 25.75| ON events | 33412.47| 12882.29|
| X-min     | 29.49 | 27.54| OFF events| 31606.19| 12111.33|
| Y-max     | 242.35| 17.65|#Language | 50    | -    |
| Y-min     | 8.55  | 16.03|#Character | 1623  | -    |
| Time (µs) | 4040930 | 2788808|#Sample | 32460 | -    |

**Table 2.** The statistical properties of the N-Omniglot dataset.

**Visual Analysis of N-Omniglot**

To better analyze the property of N-Omniglot, we calculate the statistical characteristics of N-Omniglot, as shown in the Table 2. The maximum and minimum values of the horizontal and vertical coordinates are used to crop the image to remove unnecessary input. Also, the average writing time of the strokes of each sample is 4s, which is realistic. In order to better illustrate the difference between the N-omniglot dataset and other neuromorphic datasets, we visualize some samples in NMNIST, DVS-Gesture, DVS-CIFAR10 and N-Omniglot, as shown in the Figure 3. We use OR operation to compress all event data into 12 image frames, and select 3 frames of processed images. It can be seen that for the first three neuromorphic datasets, the three image frames are very similar, and the activity is very dense, whose difference is small compared with input the static images. However, the N-omniglot dataset has large differences between frames and very sparse activities, which provides a greater challenge for building a high-performance spatio-temporal information processing algorithm, and the temporal information of the stroke order is crucial for the recognition and generation of characters. Significantly, this can also provide a good benchmark for the few-shot generation tasks.
Figure 3. Comparison of N-Omniglot with other datasets. Four neuromorphic datasets are encoded into 12 frames, and frames 2, 5 and 8 are shown above. Below is a 3D visualization of the sample.

Data Records
The N-Omniglot dataset can be downloaded in Figshare\textsuperscript{20} (https://doi.org/10.6084/m9.figshare.16821427). Because the data is collected using DV software, the stored file is aedat4 files. In addition, to minimize the additional operations of the acquisition process, we combine 20 samples into one video, using the method mentioned above to segment them. We save the beginning and ending timestamps of each sample in a CSV file. In order to better show the progress of our data compared with the previous data, we have displayed the event data on the homepage. We separate the files into different directories to facilitate maintenance and use while maintaining the same structure as Omniglot. As shown in the Figure 4, the dataset contains 1623 characters in 50 languages, and each character consists of 20 samples. The figure shows the retrieval process of the first character in the training set Alphabet_of_the_Megi. The aedat4 file records the entire event of the stroke instead of saving the color image frame. We also provide a preprocessed code at https://bii.ia.ac.cn/braincog/dataset/N-Omniglot/ to adapt it to the current algorithm application for easy use.

Technical Validation
We notice that SNN is still in its infancy in the field of neuromorphic few-shot learning, and there is almost no suitable algorithm to support this task. In order to prove the effectiveness of N-Omniglot and the potential of providing new challenges for the training of SNN algorithms, experiments are conducted on four SNN algorithms, including two general classic pattern classification methods and two few-shot learning algorithms. Also, to demonstrate the difference of N-Omniglot and the encoded Omniglot, all the experiments are performed on both of them. Then, we compare the differences between the encoded static dataset and the neuromorphic dataset.
Encoding and Preprocessing
Static images do not contain temporal information. Therefore, in order to match the characteristics of SNN in processing spatio-temporal information, static images are usually processed into spike trains. As shown in Figure 5 (a) and (c), we use Poisson coding and constant coding\textsuperscript{21, 22} as the encoding strategy, which are the most commonly used in the deep spiking neural networks. For the constant coding, the samples are fed directly into the network and the first layer can be thought as the encoding layer to generate the spike trains. On the other hand, for N-Omniglot, our dataset itself has temporal information and does not need to be encoded. However, neuromorphic datasets are acquired by DVS with high temporal resolution, and the excessively long timeline is a huge burden for current Clock-driven SNN algorithms. Therefore, datasets captured by DVS need to be merged in the temporal dimension. In our experiment, we divide the data by time average and combine them in two ways: the OR operation and spike firing rate, as shown in Figure 5 (b) and (d).

Figure 4. The structure of the N-Omniglot dataset.

Figure 5. Encoding Method for Static Image and Preprocessing for Event Data. (a) Constant Coding and (c) Poisson Coding are used to encode static images, while (b) Event Frame and (d) Firing Rate are used to preprocess neuromorphic data.
Figure 6. Four few-shot learning baseline methods for the proposed data descriptors. The image sequence represents the encoded frame. (a) The LSTM method is applied to ANN networks, and the corresponding SNN version replaces the LSTM layer with the fully connected layer. (c) Nearest Neighbor method calculates the Euclidean distance between samples as the classification basis. (b) Siamese Net compares the spiking convolution feature representation of the two samples and use the fully connected regression to classify their differences. (d) MAML method optimizes the learning process from the perspective of the gradient to make the classifier have stronger generalization.

Nearest Neighbor
As a classical pattern classification method, the nearest neighbor\cite{23, 24} (NN) method can evaluate the separability of samples to a certain extent and provide a benchmark for other algorithms to compare. As shown in Figure 6 (c), the NN method compares the input sample with each in the training set and finds the sample closest to the input according to the given distance measurement function. Then the category of the input sample can be decided by the neighbor. In the experiment, we use the Euclidean distance between different samples as the distance measurement function.

Classification Directly
The most considerable difficulty of few-shot learning lies in a large number of classification categories and the small number of samples per category. But even still, such tasks can be handled directly as general classification problems. Therefore, we construct a spiking convolutional neural network with LIF neurons to process the dataset. It consists of two convolution layers and two fully connected layers. Each convolutional layer is followed by an average pooling layer with step size 2. Due to the non-differentiable character of SNNs, we use the approximate gradient method of STBP\cite{25} for training. Even if it is not specially designed for few-shot learning tasks, the algorithm can still obtain relatively efficient results. On the other hand, as a comparison, we also design an ANN network model, as shown in Figure 5 (a), which uses the convolution layer as a feature extractor, while Long Short-Term Memory (LSTM)\cite{26}, which is a machine learning algorithm commonly used to extract time features, is employed to combine the information of Time dimension.
Siamese Net

The Siamese Net is a classical few-shot learning algorithm based on metrics. Because the original Siamese Net could not handle the neuromorphic dataset, we improve it by using SNN with LIF neurons as the basic network to add the ability to process temporal information for the model. The Siamese Net inputs two samples at the same time. If the sample pairs belong to the same category, they are marked as 1; otherwise, they are marked as 0. The network compares the two samples to determine whether they belong to the same category. As shown in Figure 6 (b), the two samples share the first half of the network structurally, and the difference between the two feature maps is input into the later fully connected layer. During the test phase, the given query set is compared with the samples of the support set one by one, and the category with the largest probability value is output as the classification result.

MAML

MAML (Model-Agnostic Meta-Learning) is another classic few-shot learning algorithm based on optimization. For the same reason, we improve MAML into SNN version, exploiting the SNN with LIF neurons as the network backbone to make the model capable of processing neuromorphic datasets. MAML tries to gain the ability to converge after a few iterations quickly. First, a fixed number of classes are randomly drawn from the dataset, with a fixed number of samples from each category, including support and query sets. Then the weights are copied and updated several times over the support set in the training phase. The loss calculation is performed with the copied weights on the query set, and the original network weight is updated with the corresponding gradient. While the same operation is performed on the support set as in the training phase and directly outputs the classification results on the query set in the test phase.

Experimental Result

Table 3 shows the classification accuracy results of Omniglot and N-Omniglot under the four methods. Different preprocessing methods and simulation time lengths were used to compare the experimental results. The last four columns of the table represent four typical configurations for few-shot learning. N-way K-shot indicates that the support set sampled from the dataset consists of N classes, and each class consists of K samples during the testing phase. Similar to Omniglot, the result of NN and direct classification methods on N-Omniglot is much better than random guesses, proving the validity of the proposed data descriptor. As shown in the table, the performance of the four methods on N-Omniglot is lower than the results on Omniglot. The first reason is that the proposed dataset is more sparse in the spatial dimension. The similarity of data in the temporal dimension is lower than the input based on static image or Poisson coding, which brings a new challenge to SNN learning. Another reason is the lack of preprocessing methods for neuromorphic datasets. It can be seen from the table that the preprocess method based on the event frame has better performance than that based on firing rate. When using firing rate, due to data sparseness in the spatial and temporal dimension, the difference of floating-point values synthesized by preprocessing is significant and not conducive to spikes’ generation and transmission. Therefore, new requirements are needed for the preprocessing method of the spiking train. It is worth noting that we simultaneously test the identification accuracy of the two classical few-shot learning methods at different simulation times. The results show that the longer the simulation time, the lower the accuracy. It is because the longer the simulation time, the more frames the event is divided into, and the more difficult it is to connect information between frames. It indicates that the data descriptor is essential for improving SNN’s ability to extract more important spatio-temporal features. Therefore, the N-Omniglot proposed in this paper can be considered an effective, robust, and challenging dataset.

Usage Notes

We provide three data interfaces for N-Omniglot to meet the requirements of different algorithms for data loading. In addition, four improved SNN learning methods, used in the technical verification section for N-Omniglot, can be found on https://bii.ia.ac.cn/braincog/dataset/N-Omniglot/. The researcher needs to download the dataset from https://doi.org/10.6084/m9.figshare.16821427 and merge the folders into two (dvs_background and dvs_evaluation). The program will first read the aedat4 file and split the file contents when first using the code. Then a folder is
| Dataset     | Methods | Structures | Encoding | Time | 20w1s     | 20w5s     | 5w1s     | 5w5s     |
|-------------|---------|------------|----------|------|-----------|-----------|-----------|-----------|
| Omniglot    | NN      | -          | Constant | 12   | 17.6±0.3% | 33.1±0.5% | 37.5±0.3% | 54.9±0.4% |
|             |         |            | Poisson  | 12   | 17.6±0.6% | 32.3±0.7% | 37.4±0.5% | 55.1±0.9% |
|             | Direct  | SCNN       | Constant | 12   | 63.2±2.2% |            |          |           |
|             |         |            | Poisson  | 12   | 64.6±0.7% |            |          |           |
|             | LSTM    |            | Constant | 12   | 66.3±0.8% |            |          |           |
|             |         |            | Poisson  | 12   | 67.0±0.8% |            |          |           |
| Siamese Net | SCNN    | 1          | Constant | 12   | 74.9±0.5% | 90.4±0.8% | 89.1±0.5% | 97.0±0.3% |
|             |         |            | Poisson  | 12   | 75.3±0.4% | 90.0±0.4% | 89.3±0.4% | 97.1±0.1% |
|             | Siamese Net | SCNN 3    | Constant | 12   | 80.7±0.3% | 94.8±0.5% | 91.5±0.1% | 97.4±0.3% |
|             |         |            | Poisson  | 12   | 83.3±0.7% | 94.8±0.2% | 93.0±0.2% | 97.8±0.1% |
|             | MAML    | SCNN       | Constant | 12   | 74.9±0.5% | 90.4±0.8% | 89.1±0.5% | 97.0±0.3% |
|             |         |            | Poisson  | 12   | 75.3±0.4% | 90.0±0.4% | 89.3±0.4% | 97.1±0.1% |
|             | NN      | -          | Event    | 12   | 7.4±0.4%  | 10.6±0.2% | 26.7±0.2% | 32.2±0.5% |
|             |         |            | Frequency| 12   | 7.3±0.3%  | 10.6±0.3% | 23.3±0.4% | 27.6±0.3% |
| N-Omniglot  | Direct  | SCNN       | Event    | 12   | 62.6±0.8% |            |          |           |
|             |         |            | Frequency| 12   | 60.8±0.9% |            |          |           |
|             | LSTM    |            | Event    | 12   | 46.2±0.3% |            |          |           |
|             |         |            | Frequency| 12   | 43.3±0.3% |            |          |           |
| Siamese Net | SCNN    | 3          | Event    | 4    | 54.0±0.7% | 76.5±0.6% | 73.1±0.6% | 89.0±0.4% |
|             |         |            | Frequency| 8    | 51.4±0.8% | 73.6±0.3% | 70.4±0.5% | 87.2±0.4% |
|             |         |            | Event    | 12   | 50.8±1.2% | 73.0±1.2% | 69.9±0.9% | 87.0±1.0% |
|             |         |            | Frequency| 4    | 53.3±1.6% | 74.9±1.6% | 72.6±0.9% | 88.4±1.0% |
|             |         |            | Frequency| 8    | 50.8±0.4% | 72.2±0.8% | 69.8±0.3% | 86.6±0.7% |
|             | MAML    | SCNN       | Event    | 4    | -         |          | 77.2±0.4% | 91.5±0.5% |
|             |         |            | Frequency| 8    | -         |          | 72.8±0.7% | 89.3±0.7% |
|             |         |            | Event    | 12   | -         |          | 70.7±0.4% | 88.9±0.6% |
|             |         |            | Frequency| 4    | -         |          | 74.4±0.7% | 90.7±0.1% |
|             |         |            | Frequency| 8    | -         |          | 71.1±0.3% | 88.6±0.4% |
|             |         |            | Event    | 12   | -         |          | 70.3±0.9% | 87.3±0.3% |

1. SCNN: 15c5-AP2-40c5-AP2-300-1623
2. ANN: (32c3-MP2)*2-LSTM128-1623
3. SCNN: 64c10-128c7-4096-1
4. SCNN: 15c5-AP2-40c5-AP2-300-5/20

Table 3. Accuracy of various modified classical methods on N-Omniglot.

created with the same structure as Omniglot, containing all the NumPy format samples, so that no more time will be spent here later. According to the different preprocessing methods, data with different frames can be obtained and directly input into the neural network. Researchers can directly use these four improved SNN few-shot learning algorithms to process datasets, or develop new algorithms to preprocess N-Omniglot datasets and propose novel few-shot learning algorithms suitable for neuromorphic datasets.

**Code availability**

Preprocessing code for the dataset and few-shot learning algorithms to verify its quality can be found here: https://bii.ia.ac.cn/braincog/dataset/N-Omniglot/. The code uses Python3 and PyTorch platforms, and the Torchvision package version is expected to be higher than 0.8.1. Please refer to the Usage Notes section and ReadMe file to run the code.
References

1. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778 (2016).
2. Hirschberg, J. & Manning, C. D. Advances in natural language processing. Science 349, 261–266 (2015).
3. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G. & Ogata, T. Audio-visual speech recognition using deep learning. Appl. Intell. 42, 722–737 (2015).
4. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. nature 323, 533–536 (1986).
5. Maass, W. Networks of spiking neurons: the third generation of neural network models. Neural networks 10, 1659–1671 (1997).
6. Shen, G., Zhao, D. & Zeng, Y. Backpropagation with biologically plausible spatio-temporal adjustment for training deep spiking neural networks. arXiv preprint arXiv:2110.08858 (2021).
7. Deng, J. et al. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, 248–255 (Ieee, 2009).
8. Lin, T.-Y. et al. Microsoft coco: Common objects in context. In European conference on computer vision, 740–755 (Springer, 2014).
9. Zhang, T. et al. Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks. Sci. Adv. 7, eabh0146 (2021).
10. Gallego, G. et al. Event-based vision: A survey. arXiv preprint arXiv:1904.08405 (2019).
11. Orchard, G., Jayawant, A., Cohen, G. K. & Thakor, N. Converting static image datasets to spiking neuromorphic datasets using saccades. Front. neuroscience 9, 437 (2015).
12. Li, H., Liu, H., Ji, X., Li, G. & Shi, L. Cifar10-dvs: an event-stream dataset for object classification. Front. neuroscience 11, 309 (2017).
13. Amir, A. et al. A low power, fully event-based gesture recognition system. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7243–7252 (2017).
14. Sironi, A., Brambilla, M., Bourdais, N., Lagorce, X. & Benosman, R. Hats: Histograms of averaged time surfaces for robust event-based object classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1731–1740 (2018).
15. Zylberberg, J., Murphy, J. T. & DeWeese, M. R. A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields. PLoS computational biology 7, e1002250 (2011).
16. Kadam, S. & Vaidya, V. Review and analysis of zero, one and few shot learning approaches. In International Conference on Intelligent Systems Design and Applications, 100–112 (Springer, 2018).
17. Wang, Y., Yao, Q., Kwok, J. T. & Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. ACM Comput. Surv. (CSUR) 53, 1–34 (2020).
18. Taherkhani, A. et al. A review of learning in biologically plausible spiking neural networks. Neural Networks 122, 253–272 (2020).
19. Lake, B. M., Salakhutdinov, R. & Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. Science 350, 1332–1338 (2015).
20. Li, Y., Dong, Y., Zhao, D. & Zeng, Y. N-omniglot: a large-scale dataset for spatio-temporal sparse few-shot learning. *figshare* [https://doi.org/10.6084/m9.figshare.16821427](https://doi.org/10.6084/m9.figshare.16821427) (2021).

21. Ding, J., Yu, Z., Tian, Y. & Huang, T. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654* (2021).

22. Li, Y., Zeng, Y. & Zhao, D. Bsnn: Towards faster and better conversion of artificial neural networks to spiking neural networks with bistable neurons. *arXiv preprint arXiv:2105.12917* (2021).

23. Peterson, L. E. K-nearest neighbor. *Scholarpedia* 4, 1883 (2009).

24. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13, 21–27 (1967).

25. Wu, Y., Deng, L., Li, G., Zhu, J. & Shi, L. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. neuroscience* 12, 331 (2018).

26. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* 9, 1735–1780 (1997).

27. Koch, G., Zemel, R., Salakhutdinov, R. *et al.* Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, vol. 2 (Lille, 2015).

28. Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135 (PMLR, 2017).

**Acknowledgements**

This study is supported by National Key Research and Development Program (2020AAA0107800), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100).

**Author contributions statement**

L.Y., Z.D. and Z.Y. generated the idea, L.Y. and D.Y. performed the experiment and analysed the results. L.Y. wrote the manuscript. D.Y., Z.D. and Z.Y. reviewed the manuscript.

**Competing interests**

The authors declare no competing interests.