

TECHNICAL NOTE • OPEN ACCESS

NeuroMorse: a temporally structured dataset for neuromorphic computing

To cite this article: Ben Walters *et al* 2025 *Neuromorph. Comput. Eng.* **5** 027001

View the [article online](#) for updates and enhancements.

You may also like

- [BAM-SLDK: biologically inspired attention mechanism with spiking learnable delayed kernel synapses](#)
Mario Chacón-Falcón, Alberto Patiño-Saucedo, Luis Camuñas-Mesa et al.
- [Spike-based computation using classical recurrent neural networks](#)
Florent De Geeter, Damien Ernst and Guillaume Drion
- [Accelerating spiking neural networks with parallelizable leaky integrate-and-fire neurons](#)
Sidi Yaya Arnaud Yarga and Sean U N Wood



TECHNICAL NOTE

OPEN ACCESS

RECEIVED
9 January 2025REVISED
24 April 2025ACCEPTED FOR PUBLICATION
2 May 2025PUBLISHED
13 May 2025

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



NeuroMorse: a temporally structured dataset for neuromorphic computing

Ben Walters¹ , Yeshwanth Bethi² , Taylor Kergan⁴ , Binh Nguyen⁴, Amirali Amirsoleimani³, Jason K Eshraghian⁴ , Saeed Afshar² and Mostafa Rahimi Azghadi^{1,*} ¹ College of Science and Engineering, James Cook University, Townsville, Australia² International Centre for Neuromorphic Systems, Western Sydney University, Sydney, Australia³ Department of Electrical Engineering and Computer Science, York University, Toronto, Canada⁴ Department of Electrical and Computer Engineering, University of California, Santa Cruz, The United States of America

* Author to whom any correspondence should be addressed.

E-mail: mostafa.rahimiazghadi@jcu.edu.au

Keywords: neuromorphic computing, benchmarks, spiking neural networks

Abstract

Neuromorphic engineering aims to advance computing by mimicking the brain's efficient processing, where data is encoded as asynchronous temporal events. This eliminates the need for a synchronisation clock and minimises power consumption when no data is present. However, many benchmarks for neuromorphic and spiking algorithms primarily focus on spatial features, neglecting the temporal dynamics that are inherent to most sequence-based tasks. This gap may lead to evaluations that fail to fully capture the unique strengths and characteristics of neuromorphic systems. In this paper, we present NeuroMorse, a temporally structured dataset designed for benchmarking spiking learning algorithms. NeuroMorse converts the top 50 words in the English language into temporal Morse code spike sequences. Despite using only two input spike channels for Morse dots and dashes, complex information is encoded through temporal patterns in the data. The proposed benchmark contains feature hierarchy at multiple temporal scales that test the capacity of spiking algorithms to decompose input patterns into spatial and temporal hierarchies. We demonstrate that our training set is challenging to categorise using a linear classifier and that identifying keywords in the test set is difficult using conventional methods. The NeuroMorse dataset is available at <https://doi.org/10.5281/zenodo.12702379>, with our accompanying code at <https://github.com/jc427648/NeuroMorse>.

1. Introduction

Neuromorphic computing draws upon brain-inspired principles in order to develop low power computing paradigms. Compared to current artificial neural networks (ANNs), spiking neural networks (SNNs) rely on the dynamic generation and transmission of binary spikes between neuronal nodes. This is significantly different from the data-driven approaches that are in use today [1, 2]. However, this shift from data-driven to event-driven computation requires new methods to train and test spiking models.

SNNs encode information across both spatial and temporal domains using precisely-timed binary-valued signals, making them well-suited for event-based tasks. However, benchmarking these networks effectively requires datasets that contain this rich spatio-temporal structure. Many current benchmarks fail to leverage temporal data aspects adequately [3]. Instead, they often rely on static datasets converted into spatio-temporal spike trains or event-based datasets that lack a focus on the hierarchical structure of spike/event times. Developing benchmarks that provide a comprehensive understanding of all model elements is crucial for evaluating and improving SNNs [4–8].

Table 1 provides a comparison of various datasets used to benchmark neuromorphic systems. In this table, we provide a description of each dataset and a qualitative guide to the spatial or temporal hierarchy information contained in each dataset. Here, 'High' implies that the data is almost explicitly encoded in this

Table 1. Comparison of various datasets used to verify neuromorphic computing architectures.

Dataset	Description	Spatial hierarchies	Temporal hierarchies
Static datasets			
MNIST [9, 10]	Grayscale images of handwritten digits (0–9)	High	None
Fashion-MNIST [11]	Grayscale images of clothing items (10 categories)	High	None
E-MNIST [12]	Grayscale images of handwritten alphanumeric characters	High	None
CIFAR [13]	Colour images of various objects	High	None
Caltech [14]	Colour images of various objects	High	None
Event-based datasets			
N-MNIST [15]	Spiking version of MNIST dataset	High	Low
Poker DVS [16]	Poker pip symbols recorded through DVS cameras	High	Low
DVS Gestures [17]	Hand gestures recorded by DVS cameras	High	Medium
ASL DVS [18]	American Sign Language letters recorded via DVS cameras	High	Medium
SSC [19]	Spiking version of the speech commands dataset	Medium	Medium
SHD [19]	Spike-based audio classification of spoken digits	Medium	Medium
NHP [20]	Prediction of electro-physiological motor signals	Medium	Medium
MSWC [21]	Audio keyword detection for multiple languages	Medium	Medium
Prophesee 1MP Gen4 automotive [22]	Object detection using event-based camera	High	Medium
Mackey-Glass [23]	Prediction of low-dimensional time signal	Low	Medium
Braille Letter Reading [24]	Learning event-based tactile sensor data for braille characters	Low	Medium–High
NeuroMorse	Top 50 English words converted to Morse code spike sequences	Low	High

domain, and virtually impossible to decipher by using other means. ‘Medium’ implies that structures and dependencies in this domain do exist, but temporal and spatial features carry similar importance. Event-based gesture datasets are an example of this, where the spatial information outweighs the importance of the temporal information, given that such tasks can be solved without recurrent models. Audio classification, as in Spiking Heidelberg Digits (SHDs)/Spiking Speech Commands (SSC), have stronger dependence on time-varying features, though the number of channels is often similar to or greater than that of the number of sequence steps when training classification models. ‘Low’ implies that some of the information is encoded in this domain, however, the majority of the information is encoded in another. An example of this is the N-MNIST [15] dataset that simply converts the MNIST dataset into a spiking dataset via an event based camera. Finally, ‘None’ implies that no data is encoded in this domain.

As shown in table 1, some common datasets used to investigate neuromorphic architectures are static image datasets, such as MNIST [9, 10], Fashion-MNIST [11], Extended-MNIST [12], CIFAR [13] and Caltech [14]. However, these datasets require a transformation from static images to temporal spike

sequences. Oftentimes, this is performed by either rate or latency encoding, neither of which emphasises a hierarchy of events in time. Instead, these datasets prioritise learning the spatial hierarchies encoded within the data. In response to these static datasets, some research has been dedicated to developing spatio-temporal datasets that are more suited for neuromorphic computing. This includes an event-based version of MNIST [15], dynamic vision sensor (DVS) gestures [16, 17, 25] and spiking audio datasets like the SHD dataset [19]. However, the timing information, even in these event-based datasets, is not crucial to the success of the downstream task. In fact, events can be binned at the input channel with minimal effect on the final result. Networks trained using these datasets often show minimal drop in performance even when the temporal information is completely removed from the spiking datasets as shown by [3].

Recognizing the lack of standardized spiking benchmarks for neuromorphic computing, researchers introduced the NeuroBench framework [4]. This framework provides a suite of evaluation metrics and datasets, allowing researchers to compare networks and algorithms from an algorithmic and systematic perspective. A wide variety of algorithmic metrics such as correctness, connection and activation sparsity, footprint and synaptic operations are discussed as important parameters. The NeuroBench algorithmic track includes four initial datasets (with additional datasets continually being added) which are: the Multilingual Spoken Word Corpus [21] (for few-shot class incremental learning), the Non-Human Primate (NHP) reaching dataset [20] (for motor control prediction), the Prophesee IMP Automotive dataset [22] (for object detection), and the Mackey-Glass time series dataset [23] (for time-series prediction). Despite their utility, these datasets primarily focus on spatial or simple temporal patterns, failing to adequately test the ability of spiking networks to process complex temporal hierarchies. The Braille letter reading dataset [24] stands out as an exception. In this work, tactile sensor data, converted to spatio-temporal representations using sigma-delta encoding, captures temporal dependencies inherent in reading Braille characters. However, the restricted set of embedded sequences present a significant constraint.

In this paper, we present NeuroMorse, a dataset where the temporal aspects of spike-based learning are more critical than the spatial aspects. We reduce the number of input channels, i.e. the spatial aspect of the data, to just two, whilst encoding the data in the form of spike sequences. To the best of our knowledge, NeuroMorse is the only spiking dataset that prioritises temporal data representation over spatial or channel-wise representations. This focus could play a crucial role in advancing the development of more effective spiking learning algorithms and architectures. To enhance the dataset's scalability and generalisation, we provide both the dataset and the dataset generation code. This enables future researchers to extend the dataset by incorporating additional words, languages, or more complex sentence structures, thereby broadening its applicability in practical scenarios.

The structure of the paper is as follows: section 2 describes the generation and formatting of the NeuroMorse dataset and details its specific nature. Section 3 discusses our preliminary analysis of the dataset, while section 4 provides a summary of the paper and our findings.

2. Methodology

To begin creating our NeuroMorse dataset, we took inspiration from a Morse code dataset [26]. Morse code has two input representations: dots and dashes, where the sequence of dots and dashes represents alphanumeric characters, as shown in figure 1(a). Thus, these sequences are easily transferable to the temporal domain, where two input channels represent dots and dashes. Figure 1(b) illustrates this by displaying the top five most frequently used English words translated into NeuroMorse. Figure 1(c) presents more details at the sentence and word encoding levels. Here, Δt_1 , Δt_2 and Δt_3 represent the timestep interval between consecutive dots and dashes (5 timesteps), between consecutive characters (10 timesteps) and between consecutive words (15 timesteps) respectively.

The NeuroMorse training set includes the top 50 most frequently used words in the English language, which have been transformed into spike sequences. Each entry in the dataset provides the channel, spike time, and labels corresponding to each word. For the test set, spike sequences were generated from a corpus of 50 441 Simple Wikipedia articles (with punctuation removed for convenience). The primary metric for this dataset is identifying occurrences of the top 50 words (referred to as keywords) in the test set. Additionally, to aid in keyword identification, the dataset includes start and end times for each keyword, along with the spike time, channel, and label data.

We initially explored incorporating null classes into the training set. However, this was deemed inconsistent with the dataset's primary goal: enabling networks to identify salient spike sequences within larger sequences, irrespective of the characteristics of non-target segments. Given the difficulty of implementing null class filtering without any training examples, we opted to create a dedicated validation set. This strategy allows researchers to train their models to effectively filter out null classes without requiring explicit consideration during training. The validation set was constructed from a small text file containing

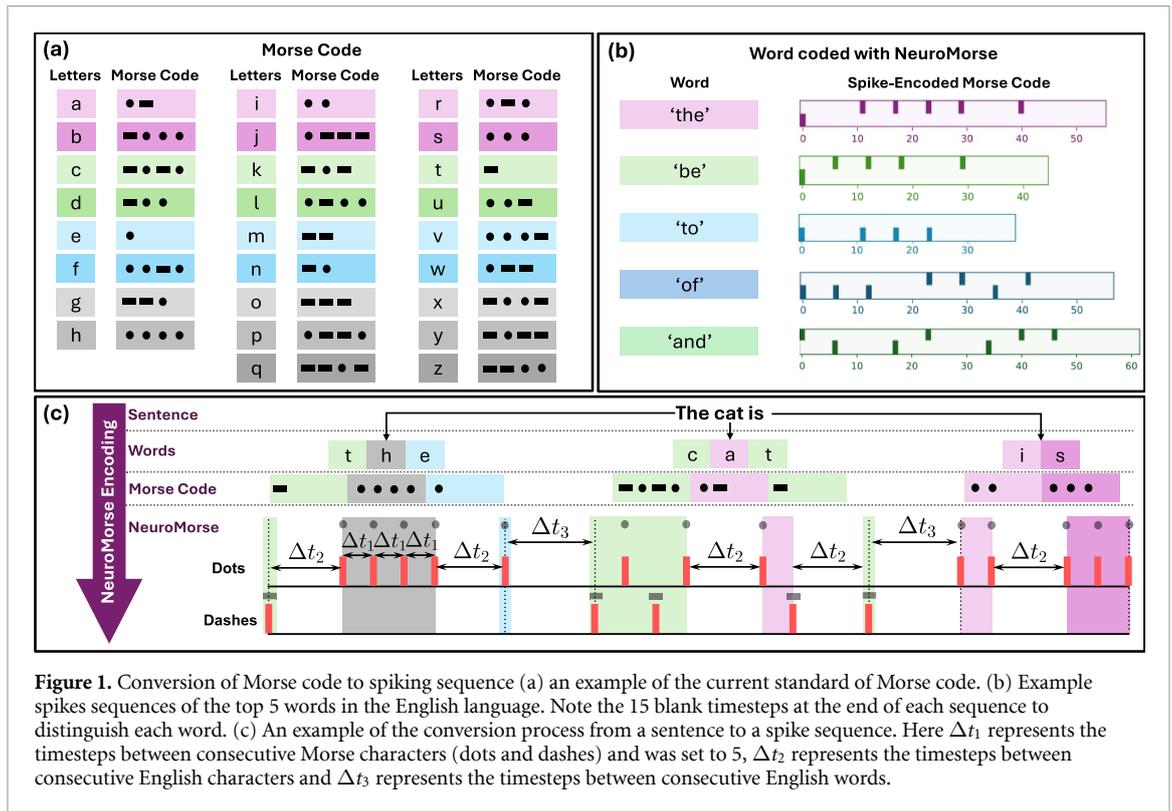


Figure 1. Conversion of Morse code to spiking sequence (a) an example of the current standard of Morse code. (b) Example spikes sequences of the top 5 words in the English language. Note the 15 blank timesteps at the end of each sequence to distinguish each word. (c) An example of the conversion process from a sentence to a spike sequence. Here Δt_1 represents the timesteps between consecutive Morse characters (dots and dashes) and was set to 5, Δt_2 represents the timesteps between consecutive English characters and Δt_3 represents the timesteps between consecutive English words.

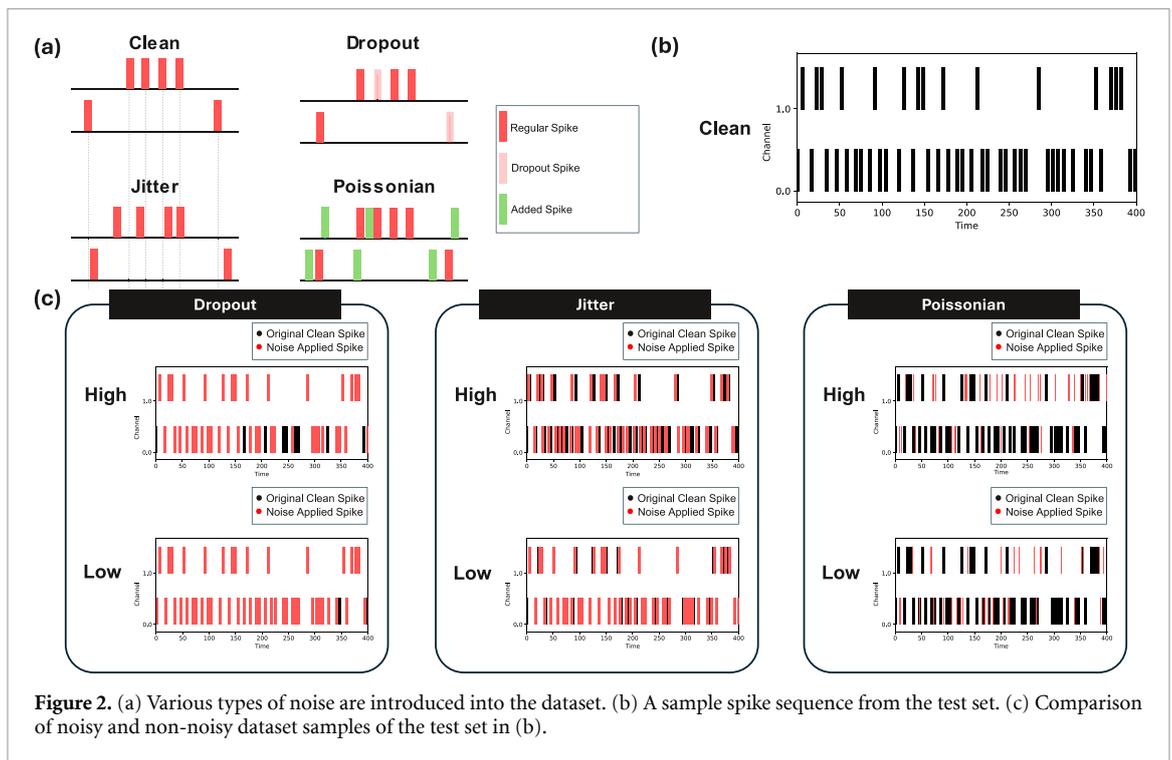
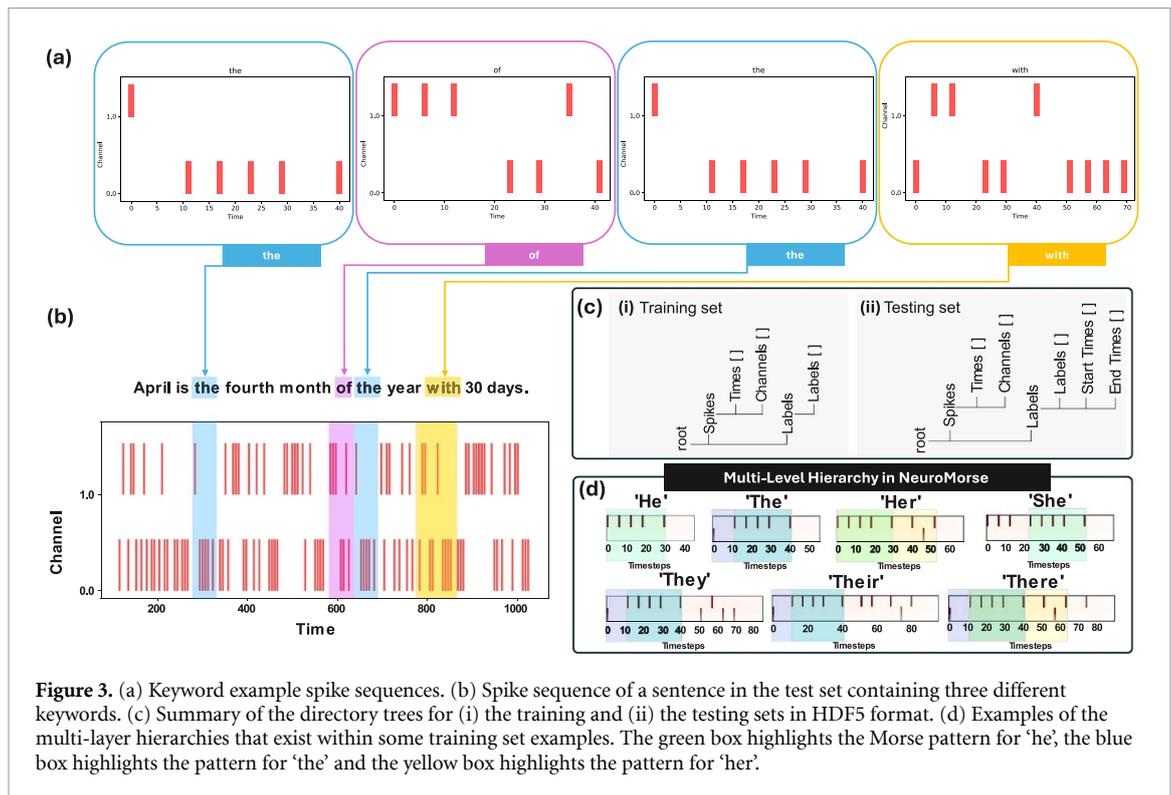


Figure 2. (a) Various types of noise are introduced into the dataset. (b) A sample spike sequence from the test set. (c) Comparison of noisy and non-noisy dataset samples of the test set in (b).

one instance of each keyword, interspersed with multiple instances of null classes, and subsequently formatted to be structurally identical to the test set.

To increase the challenge of the dataset, we introduced noisy versions of both the training and test sets. Three types of noise were applied, as illustrated in figure 2(a). The first type, spike removal (dropout), involved randomly omitting spikes with fixed probabilities of 3.33% and 6.67% for the low and high noise cases, respectively. The second type, jitter, was introduced by adding noise sampled from a Gaussian distribution (with standard deviations of 1 and 2 timesteps for low and high cases) to each spike. The final type involved adding Poissonian spike trains with rates of 0.05 and 0.1 per timestep for low and high noise



cases. Thus, the three possible cases of noise level (None, Low and High) for each of the three types of noise results in 27 total datasets. Examples of clean and noisy data can be seen in figures 2(b) and (c). Furthermore, we provide examples of keywords being identified in a sentence in the test set in figures 3(a) and (b).

The training and testing datasets are provided in Hierarchical Data Format 5 (HDF5) format for efficient storage and access. Figure 3(c)(i) illustrates the hierarchical structure of the training set. The ‘Spikes’ category contains the time stamps and channel information for each spike, while the labels are stored in a separate group. The test set follows a similar structure as shown in figure 3(c)(ii). Additionally, to facilitate evaluation, the labels group in the test set includes the start and end times of each keyword.

NeuroMorse is designed to evaluate the capability of neuromorphic computing architectures to recognise temporal patterns and hierarchical structures within spike sequences. Figure 3(d) presents sample spike sequences from the training set, highlighting embedded patterns across different words. The overlap of these patterns is particularly pronounced in the test set, which features a more extensive vocabulary. This complexity, arising from hierarchical structures, poses a significant challenge for spiking networks, demanding their ability to accurately identify both individual spike sequences and their relationships within larger patterns. Another complexity of the dataset that needs to be addressed is the unbalanced frequencies of each keyword. Table 2 shows that some keywords occur more frequently by several orders of magnitude. As such, algorithms that tackle this dataset need to be ready to handle this lopsided count.

3. Dataset tests

To show the difficulty in spike sequence learning using standard spiking architectures, we ran our training set through a linear classifier. For this, each dash or dot channel spike sequence was connected to one leaky integrate and fire neuron, as shown in figure 4(a). Equation (1) shows the formulation of this neuron model

$$U[t + 1] = \beta U[t] + X[t]. \quad (1)$$

Here, β is a leak term set to 0.95, $U[t]$ represents the membrane potential at time t and $X[t]$ represents the sum of the input spike channels at time t (where a spike is represented by the value of 1 and 0 otherwise). SnnTorch [27] was used to simulate the network. The neuron’s threshold was set arbitrarily high to ensure no spiking occurred within the network. Thus, the membrane potential was monitored and recorded at the end of each keyword presentation. Each input was also encoded with a one-hot vector. Thus the final value of membrane potential after each keyword corresponds to a one-hot vector, which could then be passed through an ordinary least-squares regression module to obtain a β value, as shown in figure 4(a). Once beta was calculated, the original membrane potential vector x was then multiplied by β to produce approximate

Table 2. Number of keyword instances for each label in the test set.

Number	Label	Count	Number	Label	Count
1	'the'	398 449	26	'they'	23 969
2	'be'	20 136	27	'we'	1960
3	'to'	99 718	28	'say'	1564
4	'of'	190 969	29	'her'	7155
5	'and'	138 885	30	'she'	8446
6	'a'	145 727	31	'or'	26 804
7	'in'	166 335	32	'an'	21 667
8	'that'	42 578	33	'will'	4765
9	'have'	15 464	34	'my'	605
10	'I'	2368	35	'one'	14 914
11	'it'	59 406	36	'all'	8351
12	'for'	38 823	37	'would'	4292
13	'not'	16 982	38	'there'	12 819
14	'on'	34 320	39	'their'	11 654
15	'with'	27 835	40	'what'	3320
16	'he'	34 707	41	'so'	5568
17	'as'	35 499	42	'up'	5565
18	'you'	1968	43	'out'	4619
19	'do'	4476	44	'if'	4923
20	'at'	20 342	45	'about'	10 929
21	'this'	20 556	46	'who'	11 351
22	'but'	14 010	47	'get'	2766
23	'his'	20 094	48	'which'	15 791
24	'by'	33 694	49	'go'	2117
25	'from'	26 787	50	'me'	421
				Total	1768 231

encoding vectors \hat{y} . Accuracy was then determined by finding the maximum value in \hat{y} and which one-hot vector in y is associated with that value.

For this task, we ignored the threshold of the neuron and only monitored the membrane potential, which was recorded after each of the 50 training input presentations. The membrane potential vector for all input words was then passed through an ordinary least squares regression module, where each input in the training set was one-hot encoded. We achieved a classification accuracy of 2/50 or 4%, i.e. double the random baseline.

We also evaluated the difficulty of detecting spike sequences within the test set. Given that the test set has additional null classes, we trained a basic network as shown in figure 4(b), where fifty neurons were connected to the 2 input channels in an all-to-all fashion via weight dependant spike timing dependant plasticity (STDP) synapses. The same LIF neurons used in the linear classifier were again used, with β set to 0.95. However, the threshold was initialised to one, allowing the network to generate spiking activity. The synaptic weights were initially randomised between values of 0 and 1, with hard-bounds set on them. These synapses only consider causal relationships i.e. pre-synaptic events that occur before post-synaptic events for weight update. Equation (2) illustrates the rule used

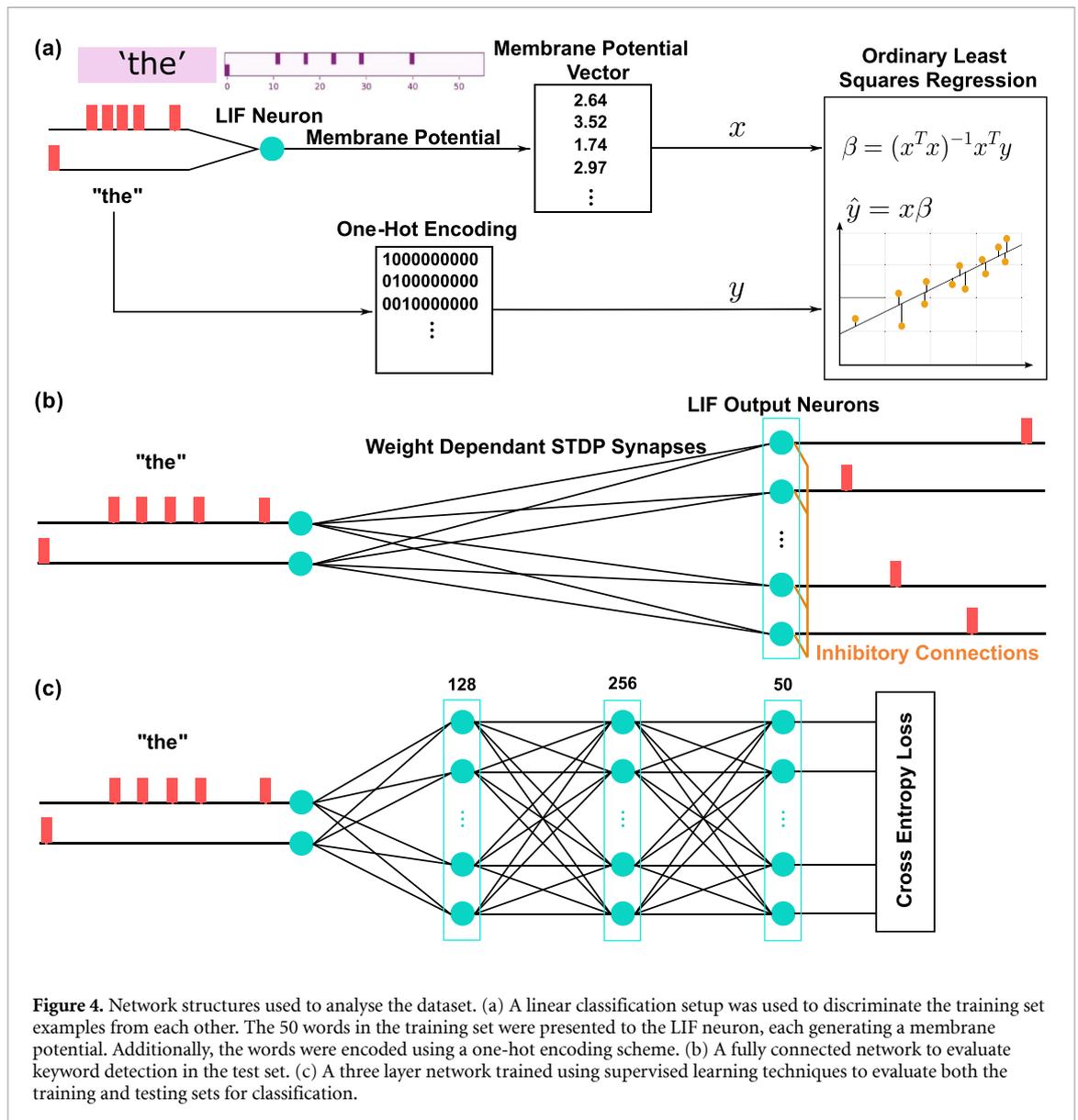
$$\Delta w = \eta \left(1 - \frac{\Delta t}{20} - w \right), \quad (2)$$

where Δw represents the change in synaptic weight, η is a scaling factor set to 0.1, Δt is the time difference between pre- and post-synaptic spikes and w is the current value of synaptic weight. This rule had previously been utilised in [26] for unsupervised learning of hierarchical relationships in input data.

Lateral inhibition was added to facilitate a winner-takes-all mechanism, allowing the neurons to discern different patterns in the training set. Furthermore, we included a threshold regulation scheme to ensure all neurons exhibit equal activity levels. The equation that governs this threshold regulation scheme is shown in equation (3),

$$\Delta V_{th} = A_{th}S[t] - \tau_{th} \quad (3)$$

where V_{th} is the threshold voltage, A_{th} is a positive fixed value of threshold increase, $S[t]$ is one if the neuron spikes and 0 otherwise and τ_{th} is a fixed positive value for threshold decay. Here we opted for values of 1×10^{-1} and 1×10^{-4} for A_{th} and τ_{th} , respectively, and did not perform any parameter optimisation.



During training, the spike sequences of the 50 training words were presented to the network in a random order for 50 epochs. Subsequently, an assignment stage was conducted to assign each of the 50 output neurons to a specific class. This involved processing each training sample again (without updating weights or thresholds) and recording the spiking output of each neuron. The output neuron with the highest spike count was assigned the corresponding class. When presented with the training set again, it was found that 3 of the 50 classes were classified correctly after training, which is slightly improved from the linear classifier. As keyword detection is a slightly different problem than classification, determining detection accuracy requires an alternative decoding method. In this work, we opted for the simplest approach that compares output spike timing with keyword timing. For each output neuron, the timing of its spikes was compared to the end times of the correct keywords in the test set. A spike was considered correct if its time matched the end time of the associated keyword. Otherwise, it was classified as incorrect. Our evaluation revealed a total of 245 correct spikes and 27 423 527 incorrect spikes, indicating a detection accuracy of only 0.00 089%. Given that the test data contains 1768 231 instances of keywords (as shown in table 2), this highlights a significant challenge: the erroneous detection of null classes. The imbalance is also a likely factor for this low accuracy, as more frequent classes in the test set can be erroneously detected.

Additionally, we performed one more test using a more complex SNN to evaluate the complexity of NeuroMorse. Here we utilised spiking backpropagation using surrogate gradients, on a deeper 3-layer network as shown in figure 4(c). This supervised learning scheme was used to evaluate the classification accuracy of the training and test set (and not for keyword detection.) By using a deeper network, more complex features can be extracted than the previously utilised shallow networks. To train this network, all

Table 3. Network parameters for supervised spiking neural network.

Network structure	
Layers	3
Layer 1 size	128
Layer 2 size	256
Layer 3 size	50
Input channels	2
Neuron parameters	
Neuron model	Leaky integrate and fire
β	0.8
Surrogate gradient	Fast sigmoid. Slope = 15
Learnable threshold	TRUE
Learnable β	TRUE
Training parameters	
Loss function	Cross entropy loss
Optimiser	Adam
Batch size	50
Training epochs	2000
Learning rate	0.001

Table 4. Summary of noisy supervised network results.

		Top-1 classification accuracy											
		Jitter			None			Low			High		
		Poisson	None	Low	High	None	Low	High	None	Low	High		
Dropout	None		12.27%	4.37%	2.48%	6.22%	2.49%	1.69%	5.70%	2.07%	1.66%		
	Low		11.81%	4.30%	2.40%	6.26%	2.42%	1.65%	5.76%	2.00%	1.60%		
	High		11.34%	4.09%	2.30%	6.16%	2.35%	1.70%	5.84%	2.06%	1.57%		
		Top-5 classification accuracy											
		Jitter			None			Low			High		
		Poisson	None	Low	High	None	Low	High	None	Low	High		
Dropout	None		30.67%	16.04%	9.18%	22.65%	13.06%	8.12%	20.61%	12.51%	7.91%		
	Low		28.41%	15.73%	9.24%	21.43%	13.19%	8.10%	19.60%	12.40%	7.95%		
	High		26.57%	15.52%	9.22%	20.28%	12.88%	8.24%	18.86%	12.51%	8.05%		

inputs were padded to the same length, and the fast sigmoid function with a slope of 15 was used for the surrogate gradients. The full list of network parameters are provided in table 3. Given the complexity of the NeuroMorse dataset, we have also measured the top-5 classification accuracy, which is also used for the ImageNet dataset [28]. This measure provides some guidance as to whether the model is converging to the correct result.

After training the network for 2000 epochs, we first presented the training set one more time to identify the keywords. The network achieved a top-1 classification accuracy of 82.00% and a top-5 accuracy of 94.00% when evaluated on the training set. While these high results were expected due to the network being exposed to the same data, there is still room for improvement. On the test set, the model correctly classified keywords 12.27% of the time, with a top-5 accuracy of 30.67% and no noise as shown in table 4. Performance under noisy variations of the test set is further summarized in table 4, demonstrating the detrimental impact of noise on the network performance.

We compared our results obtained on the NeuroMorse dataset with two commonly used event-based datasets: DVS Gestures [17] and the SHD [19]. The same network architectures and algorithms used for NeuroMorse were applied to these datasets, with adjustments only to the input size based on the data. For the STDP networks, the classification results were 21.97% and 6.1% for the DVS and SHD dataset respectively, compared to 0.000 89% on NeuroMorse. Note that for the STDP networks, we performed keyword detection on NeuroMorse whilst only classification was performed for the DVS and SHD datasets, which potentially

contributes to the difference in performance. In contrast, the supervised network achieved 81.71% on DVS and 78.72% on SHD, compared to 12.1% top-1 classification accuracy on NeuroMorse. These results confirm the complexity of NeuroMorse and show that whilst conventional STDP and supervised networks are sufficient for some event-based datasets, they are insufficient for learning temporal hierarchies.

Hence, with three different dataset tests, we show the limited ability to learn temporal hierarchies in current spiking architectures. We have shown that our training set keywords are difficult to discriminate using a basic linear classifier. We also demonstrate that a basic STDP network is unable to learn and pick out the keywords in the test set effectively. Finally, we have shown that even supervised approaches struggle to discriminate and classify keywords in the training and testing sets.

4. Conclusion

This paper introduced a novel spiking dataset designed to prioritize temporal structures and hierarchies over spatial hierarchies. Our initial experiments demonstrated the dataset's non-linearity, as evidenced by the low accuracy (4%) achieved with ordinary least squares regression. Furthermore, basic spiking networks struggled on the complex NeuroMorse test set, reaching very low accuracies. Given the dataset's emphasis on temporal patterns and hierarchical relationships, it presents a valuable resource for researchers investigating the capabilities of neuromorphic architectures in handling complex temporal data. We encourage the neuromorphic community to utilize this dataset in conjunction with other benchmarks to explore the full potential of their models in tasks such as temporal sequence recognition and hierarchical feature extraction.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://github.com/Ben-E-Walters/NeuroMorse>.

ORCID iDs

Ben Walters  <https://orcid.org/0000-0001-5464-8468>

Yeshwanth Bethi  <https://orcid.org/0000-0002-0713-0903>

Taylor Kergan  <https://orcid.org/0009-0004-6277-8018>

Jason K Eshraghian  <https://orcid.org/0000-0002-5832-4054>

Mostafa Rahimi Azghadi  <https://orcid.org/0000-0001-7975-3985>

References

- [1] Sun H, Shen L, Zhong Q, Ding L, Chen S, Sun J, Li J, Sun G and Tao D 2024 *Neural Netw.* **169** 506–19
- [2] Chen Y, Xia R, Yang K and Zou K 2024 *App. Soft Comput.* **154** 111392
- [3] Iyer L R, Chua Y and Li H 2021 *Front. Neurosci.* **15** 608567
- [4] Yik J et al 2025 *Nat. Commun.* **16** 1545
- [5] Ostrau C, Klarhorst C, Thies M and Rückert U 2022 *Front. Neurosci.* **16** 873935
- [6] Davies M 2019 *Nat. Mach. Intell.* **1** 386–8
- [7] Milde M B et al 2022 *Front. Neurosci.* **16** 813555
- [8] Stewart T C, Schuman C, Sandamirskaya Y, Furber S and Indiveri G 2024 *Neuromorph. Comput. Eng.* **4** 040202
- [9] Deng L 2012 *IEEE Signal Process. Mag.* **29** 141–2
- [10] LeCun Y 1998 (available: <http://yann.lecun.com/exdb/mnist/>)
- [11] Xiao H, Rasul K and Vollgraf R 2017 arXiv:1708.07747
- [12] Cohen G, Afshar S, Tapson J and van Schaik A 2017 EMNIST: extending MNIST to handwritten letters 2017 *Int. Joint Conf. on Neural Networks (IJCNN)* pp 2921–6
- [13] Krizhevsky A and Hinton G 2009 Learning multiple layers of features from tiny images *Master's thesis* University of Toronto
- [14] Li F F, Andreeto M, Ranzato M and Perona P 2022 Caltech 101
- [15] Orcharad G, Jayawant A, Cohen G K and Thakor N 2015 *Front. Neurosci.* **9** 437
- [16] Serrano-Gotarredona T and Linares-Barranco B 2015 *Front. Neurosci.* **9** 481
- [17] Amir A et al 2017 A low power, fully event-based gesture recognition system 2017 *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 7388–97
- [18] Bi Y, Chadha A, Abbas A, Bourtsoulatzé E and Andreopoulos Y 2019 Graph-based object classification for neuromorphic vision sensing 2019 *IEEE Int. Conf. on Computer Vision (ICCV)* (IEEE)
- [19] Cramer B, Stradmann Y, Schemmel J and Zenke F 2020 *IEEE Trans. Neural Netw. Learn. Syst.* **33** 2744–57
- [20] O'Doherty J E, Cardoso M M B, Makin J G and Sabes P N 2020 Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology (<https://doi.org/10.5281/zenodo.3854034>)
- [21] Mazumder M et al 2021 Multilingual spoken words corpus 35th *Conf. on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*
- [22] Perot E, De Tournemire P, Nitti D, Masci J and Sironi A 2020 *Advances in Neural Information Processing Systems* vol 33 pp 16639–52
- [23] Mackey M C and Glass L 1977 *Science* **197** 287–9

- [24] Müller-Cleve S F *et al* 2022 *Front. Neurosci.* **16** 951164
- [25] Li H, Liu H, Ji X, Li G and Shi L 2017 *Front. Neurosci.* **11** 309
- [26] Bethi Y, Xu Y, Cohen G, Van Schaik A and Afshar S 2022 *IEEE Access* **10** 97912–29
- [27] Eshraghian J K, Ward M, Neftci E O, Wang X, Lenz G, Dwivedi G, Bennamoun M, Jeong D S and Lu W D 2023 Training spiking neural networks using lessons from deep learning *Proc. IEEE* vol 111 pp 1016–54
- [28] Deng J, Dong W, Socher R, Li L J, Li K and Fei-Fei L 2009 ImageNet: a large-scale hierarchical image database 2009 *IEEE Conf. on Computer Vision and Pattern Recognition* pp 248–55