The Spike Gating Flow: A Hierarchical Structure Based Spiking Neural Network for Spatiotemporal Computing

Zihao Zhao [1] **Yanhong Wang** [1] **Qiaosha Zou** [1] **Xiaoan Wang** [2] **C.-J. Richard Shi** [1] **Junwen Luo** [2]

Abstract

Current deep learning faces major challenges for action recognition tasks because of: 1) the huge computational cost and 2) the inefficient learning. Hence, we develop a novel Spiking Neural Network (SNN) titled Spiking Gating Flow (SGF) for such a dilemma. The developed system consists of multiple SGF units which assembled in a hierarchical manner. A single SGF unit involves three layers: a feature extraction layer, an event-driven layer, and a histogram-based training layer. By employing a dynamic visions sensor gesture dataset, the results indicate that we can achieve 87.5% accuracy which is comparable with Deep Learning (DL), but at smaller training/inference data number ratio 1.5:1. And only a single training epoch is required during the learning process. At last, we conclude the few-shot learning paradigm of the developed network: 1) a hierarchical structure-based network design involves human prior knowledge; 2) SNNs for content based global dynamic feature detection.

1. Introduction

Deep Learning (DL) nowadays exerts a substantial impact on a wide range of computer vision tasks such as face recognition (Hu et al., 2015) and image classifications (Krizhevsky et al., 2012). But it is still facing major challenges when processing information with high dimensional spatiotemporal dynamics such as video action recognition. This is because of the huge computational cost: the deep neural networks have to capture dynamic information across another timing dimensions, which requires significant computational resources for the training stage (He et al., 2016). One promising technology of sparsity (Liu et al., 2015; Wen et al., 2016; Liu et al., 2021) can relieve the first issue of the intensive computing to some extend, but the training cost is still enormous.

Spiking Neural Networks (SNNs) is an alternative candidate to perform spatiotemporal related tasks (Lobo et al., 2020) with few-shot learning capability. Employing SNNs for action recognition remains challenging since it lacks an efficient learning algorithm. Recently SNN based learning systems can be classified into three levels: a micro-level, a middle-level and a macro-level system. A micro-level based systems emphasis on utilizing low-level spiking neuron computing characters such as a temporal process and an integration-and-fire manner (Wu et al., 2018; Amir et al., 2017; Lee et al., 2016; Zhang & Li, 2019; Caporale & Dan, 2008). For instances, (Wu et al., 2018) illustrates a Convolution Neural Network (CNN) based SNN for gesture classification. By employing an event-driven sensor and a TrueNorth neuromorphic chip, the system shows 178.8mW power consumption and 96.49% accuracy. However, the SNN higher-level computing entities such as attractor dynamics are missed in the system, which results in inefficient learning.

A middle-level system indicates SNNs apply global dynamic behaviors on the learning process (Eliasmith, 2005; Bekolay et al., 2014; Voelker et al., 2019; Chilkuri et al., 2021; Luo & Chen, 2020; Sussillo & Abbott, 2009). A Neural Engine Framework (NEF) develops a method to build dynamic systems based on spiking neurons (Bekolay et al., 2014). Such an approach leverages neural non-linearity and weighted synaptic filter as computational resources.

A macro-level system includes both micro-level and middlelevel system's advantages (Sussillo & Abbott, 2009; Imam & Cleland, 2019). It combines detailed spiking neuron characters and network dynamics together to form a unique learning system. (Wu et al., 2022) proposed a spike-based hybrid plasticity model for solving few-shot learning, continual learning, and fault-tolerance learning problems, it combines both local plasticity and global supervise information for multi-task learning.

In this work we develop a novel macro-level system titled Spike Gating Flow (SGF) for action recognition as shown in Fig. 1. The system consists of multiple SGF units that

[1]Fudan University, China [2]BrainUp Research Lab, China. Correspondence to: Junwen Luo <luojunwen@naolubrain.com>.

DynN workshop at the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, 2022.

connect in a hierarchical manner. An SGF unit consists of three layers: 1) a feature extraction layer for global dynamic feature detection; 2) an event-driven layer for generating event global feature vectors; 3) a supervise-based histogram training layer for online learning. By employing a Dynamic Vision Sensor (DVS) (Posch et al., 2011) based gesture dataset (Amir et al., 2017), the results demonstrate that the developed SGF has great learning performance: the system can approximately achieve the same level accuracy of 87.5% as the DL but at a training/inference sample ratio 1.5:1 condition. More importantly, only one epoch is required during the training. In summary, the contributions are as follows:

- **Algorithm aspect**: We developed an efficient few-shot learning system for gesture recognition, which behaves like the biological intelligence: few-shot learning, energy efficient and explainable.
- **Learning theory aspect**: We conclude one few-shot learning paradigm: 1) a hierarchical structure-based network design involves with human prior knowledge; 2) SNNs for global dynamic feature detection.

2. The Spike Gating Flow

The Spike Gating Flow (SGF) is a new dynamic network to achieve online few-shot training entities, which is inspired from the Neural Engineering Framework (NEF) (Paulin, 2004) and brain assemble theories (Papadimitriou et al., 2020). In brief, the few-shot learning capabilities rely on prior knowledge embedded in the hierarchical architecture and global feature computing. While the online computing benefits from using dynamic spike pattern to encode both data and control flow. Therefore, network different level nodes are served as gates to pass or stop input data information, and spikes are served as gate control signals. We have concluded the key principles of SGF as below:

- **Global feature representations**: Network representations are defined by the combination of different SNNs global movement features rather than pixel local features.
- **Tailor designed hierarchical network structure**: A hierarchical structure-based network for conditional data-path execution. Depending on inputs, SGF unit spike patterns are served as gates command to manipulate data-paths.
- **Histogram based training algorithms**: A global feature-based histogram training adjusts output layer weights based on history information.

Based on such principles, we design three SGF units and carefully connect them into a two-level network. Each SGF

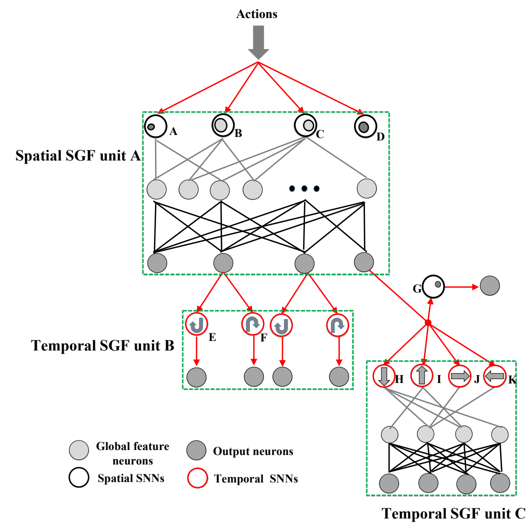

Figure 1. The SGF network architecture. It mainly consists of three SGF units: a spatial SGF unit A and temporal SGF units B and C. A spatial SGF unit A has four SNNs with feature ID index A-D (A: intensive activities at constrained left areas; B: mild activities at plateau left areas; C: mild activities at plateau right areas D: intensive activities at constrained right areas). A temporal SGF unit B has two SNNs with feature ID index E-F (F: clockwise movement; G: clockwise counter movement). A temporal unit C has four SNNs with feature ID index H-K (H: top-down; I: bottom-up; J: left-right; D: right-left). Also, the developed network has 10 output neurons corresponding to 10 action types.

unit has several corresponding spatial SNNs and temporal SNNs, which targets on detecting different global features (features with index A-I are shown at Fig. 1). Next there is an event-driven layer that connects SNNs outputs to the global feature neurons. This layer responses for generating event feature vectors for the next layer training. Typically, an event class will have several feature vectors types due to the spatiotemporal variations. A feature vector can be defined as a combination of active SNNs' feature index, which are represented by connecting active SNNs to one global neuron. Therefore, for each action type, global feature neuron number is equal to the action type feature vector type number. At last, an SGF unit has a fully connected histogram-based training layer, in which each output neuron connects to its all global feature neurons. After each training trail, feature vector histogram numbers will be updated and converted into corresponding weights. And the conversion is a normalization process.

At an inference stage, a test sample generated feature vector will be sent into all output neurons for calculating final scores, which follows the equation as below:

$$S_m = \sum_j \frac{\left(T_j^m * V\right)}{L_v} \times w_m^j \tag{1}$$

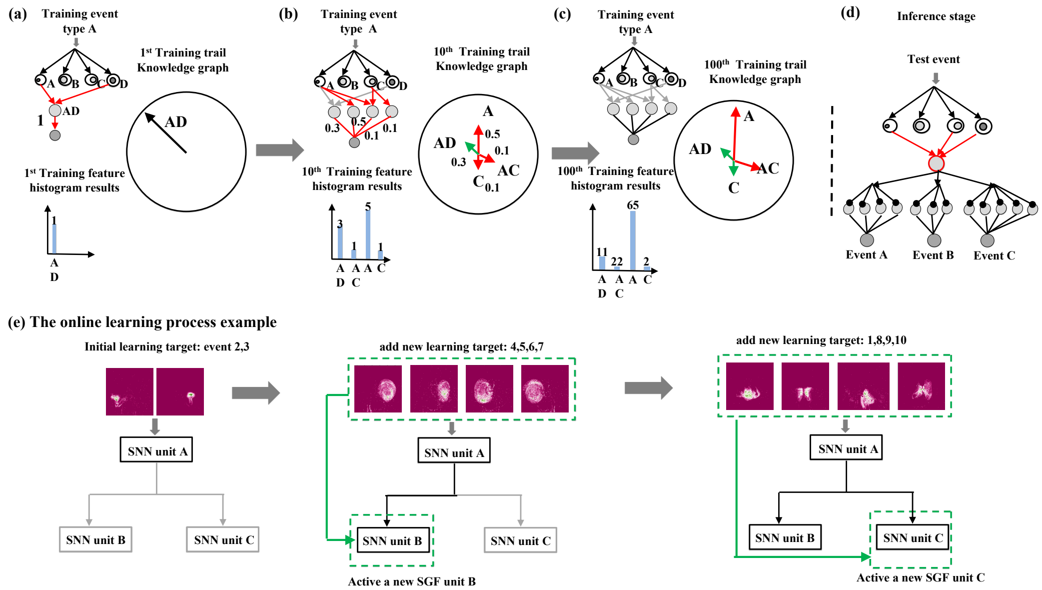

Figure 2. (a)-(d) The histogram-based training example of an event type A; (e) The system online learning process example.

Where S_m is a testing sample score at m^{th} output neuron; w_m is the j^{th} feature vector weights of the m^{th} output neuron; T_j^m is the j^{th} feature vector of the m^{th} output neuron; and V is the feature vector of the testing samples. The symbol $*$ is a bit-wise NOR operation, and L_v is the bit length of the feature vector. The key advances of such a learning algorithm are that each data sample only requires one training time and tiny computational resources for updating weights, which enables rapid online learning behaviors.

A detailed example is illustrated at Fig.2. SNNs with feature index A and D are active at the first training trail, which forms a feature vector$[A - D]$. Hence a corresponding global feature neuron is generated that connects to SNNs with feature index A and D (connected with red lines). And a feature vector histogram is also displayed on the Fig. 2(a) left. After that, the feature vector$[A - D]$ histogram values will be converted into event type A output neuron weights. It is clearly seen that the weight is one since there is only one feature vector type (Fig. 2(a)). Meanwhile a knowledge graph of event type A is produced for quantitative analysis feature vector distributions (Fig. 2(a) right). At a $10th$ training trail, there are three more feature vectors generated $[A - C, A, C]$ (Fig. 2(b) left red lines). This indicates that there are in total four types of feature vector in the event type A. Identically, corresponding feature vectors histogram numbers $[3, 1, 5, 1]$ will be transformed into event A neuron outputs weights via a training layer. The feature vectors distribution is also updated in the knowledge graph: a vector with green lines indicates histogram values are decreased, while a vector with red lines indicates histogram values are raised. At the end of a 100th training trail, there is no new feature vector appeared, which results of the same global feature neuron number as the 10th training trail. The event A output neuron weights are updated based on the current histogram numbers as a final result. Similarly, the other event types B and C follow the same training procedures.

At the inference stage, a test sample is given into the trained network, which will generate a corresponding test feature vector. And it will go through all the output neurons to calculate the final scores. As shown at Fig. 2(d), there is a trained network which contains three output neurons, whose inference classification result is the maximum one among these output neuron scores.

The online learning process example is shown at Fig. 2(e). At an initial stage, event group A [3: right hand wave; 2: left hand wave] is sent into the network for training. Since event group A contains significant spatial features, only a spatial SGF unit A is active and responsible for generating feature vectors. After finishing learning event group A, event group B [4: right arm clockwise, 5: right arm counter clockwise, 6: left arm clockwise, 7: left arm counter clockwise] is sent into the network for sequential online learning. Identically, a temporal SGF unit B is active for recognizing clockwise/counter clockwise movements. At last, event group C [1: hand clap, 2: left hand wave, 8: arm rolls, 9: air drum, 10: air guitar] is sent into network that contains complex combinations of vertical and horizontal movements. The rest of SGF unit C is active for learning such features. As it can be seen, the final network architecture varies depending on the learning targets.

Name	Type	Learning method	Learning style	Model information				Training cost		Accuracy
				Size	Diff($\times$)	OPs	Diff($\times$)	Epoch	T/I ratio	
Reservoir CSNN (George et al., 2020)	SNN	STDP	Offline	3.17MB	$88.7\uparrow$	-		-	3.8:1	65.0%
Heterogeneity Network (Perez-Nieves et al., 2021)	SNN	SGD	Offline	125KB	$3.4\uparrow$	-		-	3.8:1	82.1%
SCRNN (Xing et al., 2020)	ANN2SNN	BPTT	Offline	732.34KB	$20.0\uparrow$	81.91M	$9.9\uparrow$	100	4.1:1	96.59%
SLAYER (Shrestha & Orchard, 2018)	ANN2SNN	BP	Offline	1034.8KB	$28.3\uparrow$	79.8M	$9.6\uparrow$	739	3.8:1	93.64%
Converted SNN (Kugele et al., 2020)	ANN2SNN	BP	Offline	500KB	$13.7\uparrow$	651M	$78.7\uparrow$	10	3.8:1	96.97%
ConvNet (Amir et al., 2017)	DNN2SNN	BP	Offline	16.3MB	$456\uparrow$	946.82M	$114\uparrow$	250	3.8:1	96.5%
PointNet++ (Qi et al., 2017)	DNN2SNN	BP	Offline	3.50MB	$98\uparrow$	440.0M	$53.2\uparrow$	250	3.8:1	97.08%
This work	**SNN**	**SGF**	**Online**	**36.58KB**		**8.27M**		**1**	**1.5:1**	**87.5%**

”-” indicates the data can not be calculated or not mentioned in the corresponding paper.

Table 1. The comparison between state-of-the-art methods and the proposed SGF network.

3. SNNs Design

We have developed three SNN types that are SpatioTemporal (ST) cores, spatial SNNs and temporal SNNs.

Spatiotemporal Core The equation of spatiotemporal core is shown as below:

$$ST_m^t = \left[\int_{t-\Delta ST_t}^{t} \left[\sum_{i}^{i+\Delta ST_s} d_m^t \right]^{\theta_s} dt \right]^{\theta_t} \quad (2)$$

Where ST_m^t is the outputs of the m^{th} ST core at frame t period; d_m^t is the outputs of the m^{th} DVS sensor pixel at frame t, which equals to -1 or +1; ΔST_s is a ST core spatial detection range. The function $[S]^{\theta_s}$ equals 1 if S over spatial thresholds θ_s. Regarding the temporal computations, ΔST_t is an integration window and θ_t is a temporal threshold. The function $[T]^{\theta_t}$ equals 1 if T is over spatial thresholds θ_t. As a result of this, by adjusting above four parameters, we can configure ST core filtering behaviors properly.

Spatial SNNs The spatial SNNs' equation is as following:

$$SP_m = \left[\sum_{i}^{i+\Delta SP_s} \left[\int_{t=0}^{t=T} ST_m^t dST \right]^{\theta_i} \right]^{\theta_a} \quad (3)$$

Where T is the total frame number of an event. ΔSP_s is the detection size and SP_m is the m_{th} spatial SNN outputs. And the outputs can be a single bit or multiple bits. θ_i is an intensity gate neuron threshold, θ_a is an area gate neuron threshold.

Temporal SNNs The temporal SNNs‘ equation is shown below:

$$TE_m^t = \left[\sum_{i}^{n^{t-\Delta TE_t}} \left[l_m^t - l_i^{t-\Delta TE_t} \right]^{\theta_l} \right]^{\theta_{te}} \quad (4)$$

Where TE_m^t is the outputs of the m^{th} temporal neuron at frame t; l_i^t is the location of the i^{th} temporal active neuron at frame t, the location can be either vertical or horizontal information depends on temporal SNN types. ΔTE_t is the comparison frame window; θ_l is the location index threshold, $n^{t-\Delta TE_t}$ is the active neuron number at frame $t - \Delta TE_t$. θ_{te} is the temporal neuron spiking threshold.

4. Results

System accuracy: A DVS gesture dataset (Amir et al., 2017) (10 different gesture actions) is employed to verify the system performance. In Tab. 1, we first compare the developed network with two typical SNN-based gesture recognition networks, a STDP based SNN (George et al., 2020) and a SGD based SNN (Perez-Nieves et al., 2021). Regarding ANN/DNN converted SNN, the developed network can reach the same level of accuracy as a SLAYER (Shrestha & Orchard, 2018), but a slight lower than ConvNet (Amir et al., 2017) 96.5%, SCRNN (Xing et al., 2020) 96.59%, Converted SNN (Kugele et al., 2020) 96.97% and PointNet++ (Qi et al., 2017) 97.08%. However, the network model size can be reduced by 456 times compared to the ConvNet (Amir et al., 2017), and number of operations can be reduced by 53 times compared to the PointNet++ (Qi et al., 2017). **Few-shot learning:** Last but not the least, the developed SGF only requires 1 training epoch at a condition of training/inference ratio 1.5:1, which DL networks typical require hundreds training epochs at a condition of 3.8:1. This indicates the system training cost is significantly lower than the DL based networks.

5. Conclusion

In this work we first employed a gesture classification task as a proof of concept. The developed network can achieve the same level of accuracy with the DL under a condition of the training/inference data ratio 1.5:1. Also, only one training epoch is required during the learning periods. At last, although the developed system capability has a considerable distance compared to the current DL network, the system shows the essential biological intelligence (e.g. few-shot learning, energy efficient, explainable).

Acknowledgements

This work was supported by the project of National Key Research and Development Plan under Grants No. 2018YFB2202604.

References

Amir, A., Taba, B., Berg, D. J., Melano, T., McKinstry, J. L., di Nolfo, C., Nayak, T. K., Andreopoulos, A., Garreau, G., Mendoza, M., Kusnitz, J., DeBole, M., Esser, S. K., Delbrück, T., Flickner, M., and Modha, D. S. A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 7388–7397. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.781. URL https://doi.org/10.1109/CVPR.2017.781.

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T., Rasmussen, D., Choo, X., Voelker, A., and Eliasmith, C. Nengo: a python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48, 2014. ISSN 1662-5196. doi: 10.3389/fninf.2013.00048. URL https://www.frontiersin.org/article/10.3389/fninf.2013.00048.

Caporale, N. and Dan, Y. Spike timing–dependent plasticity: A hebbian learning rule. *Annual Review of Neuroscience*, 31(1):25–46, 2008. doi: 10.1146/annurev.neuro.31.060407.125639. URL https://doi.org/10.1146/annurev.neuro.31.060407.125639. PMID: 18275283.

Chilkuri, N., Hunsberger, E., Voelker, A., Malik, G., and Eliasmith, C. Language modeling using lmus: 10x better data efficiency or improved scaling compared to transformers. *CoRR*, abs/2110.02402, 2021. URL https://arxiv.org/abs/2110.02402.

Eliasmith, C. A Unified Approach to Building and Controlling Spiking Attractor Networks. *Neural Computation*, 17(6):1276–1314, 06 2005. ISSN 0899-7667. doi: 10.1162/0899766053630332. URL https://doi.org/10.1162/0899766053630332.

George, A. M., Banerjee, D., Dey, S., Mukherjee, A., and Balamurali, P. A reservoir-based convolutional spiking neural network for gesture recognition from DVS input. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pp. 1–9. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9206681. URL https://doi.org/10.1109/IJCNN48605.2020.9206681.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W. J., Li, S. Z., and Hospedales, T. M. When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. In *2015 IEEE International Conference on Computer Vision Workshop, ICCV Workshops 2015, Santiago, Chile, December 7-13, 2015*, pp. 384–392. IEEE Computer Society, 2015. doi: 10.1109/ICCVW.2015.58. URL https://doi.org/10.1109/ICCVW.2015.58.

Imam, N. and Cleland, T. A. Rapid online learning and robust recall in a neuromorphic olfactory circuit. *CoRR*, abs/1906.07067, 2019. URL http://arxiv.org/abs/1906.07067.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Kugele, A., Pfeil, T., Pfeiffer, M., and Chicca, E. Efficient processing of spatio-temporal data streams with spiking neural networks. *Frontiers in Neuroscience*, 14: 439, 2020.

Lee, J. H., Delbruck, T., and Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10:508, 2016. ISSN 1662-453X. doi: 10.3389/fnins.2016.00508. URL https://www.frontiersin.org/article/10.3389/fnins.2016.00508.

Liu, B., Wang, M., Foroosh, H., Tappen, M. F., and Pensky, M. Sparse convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 806–814. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298681. URL https://doi.org/10.1109/CVPR.2015.7298681.

Liu, S., Zhao, Z., Wang, Y., Zou, Q., Zhang, Y., and Shi, C. R. Systolic-array deep-learning acceleration exploring pattern-indexed coordinate-assisted sparsity for real-time on-device speech processing. In Chen, Y., Zhirnov, V. V., Sasan, A., and Savidis, I. (eds.), *GLSVLSI '21: Great Lakes Symposium on VLSI 2021, Virtual Event, USA, June 22-25, 2021*, pp. 353–358. ACM, 2021. doi: 10.1145/3453688.3461530. URL https://doi.org/10.1145/3453688.3461530.

Lobo, J. L., Ser, J. D., Bifet, A., and Kasabov, N. K. Spiking neural networks and online learning: An overview and perspectives. *Neural Networks*, 121:88–100, 2020. doi: 10.1016/j.neunet.2019.09.004. URL https://doi.org/10.1016/j.neunet.2019.09.004.

Luo, J. and Chen, J. An internal clock based space-time neural network for motion speed recognition. *CoRR*, abs/2001.10159, 2020. URL https://arxiv.org/abs/2001.10159.

Papadimitriou, C. H., Vempala, S. S., Mitropolsky, D., Collins, M., and Maass, W. Brain computation by assemblies of neurons. *Proceedings of the National Academy of Sciences*, 117(25):14464–14472, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2001893117. URL https://www.pnas.org/content/117/25/14464.

Paulin, M. G. Neural engineering: Computation, representation and dynamics in neurobiological systems: Chris eliasmith, charles anderson; MIT press (december 2003), ISBN: 0262050714. *Neural Networks*, 17(3):461–463, 2004. doi: 10.1016/j.neunet.2004.01.002. URL https://doi.org/10.1016/j.neunet.2004.01.002.

Perez-Nieves, N., Leung, V. C. H., Dragotti, P. L., and Goodman, D. F. M. Neural heterogeneity promotes robust learning. *Nature Communications*, 2021. doi: 10.1038/s41467-021-26022-3. URL https://www.biorxiv.org/content/early/2021/03/22/2020.12.18.423468.

Posch, C., Matolin, D., and Wohlgenannt, R. A QVGA 143 db dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE J. Solid State Circuits*, 46(1):259–275, 2011. doi: 10.1109/JSSC.2010.2085952. URL https://doi.org/10.1109/JSSC.2010.2085952.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 77–85. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.16. URL https://doi.org/10.1109/CVPR.2017.16.

Shrestha, S. B. and Orchard, G. Slayer: Spike layer error reassignment in time. 2018.

Sussillo, D. and Abbott, L. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2009.07.018. URL https://www.sciencedirect.com/science/article/pii/S0896627309005479.

Voelker, A., Kajic, I., and Eliasmith, C. Legendre memory units: Continuous-time representation in recurrent neural networks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15544–15553, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/952285b9b7e7a1be5aa7849f32ffff05-Abstract.html.

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. Learning structured sparsity in deep neural networks. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2074–2082, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/41bfd20a38bb1b0bec75acf0845530a7-Abstract.html.

Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12: 331, 2018. ISSN 1662-453X. doi: 10.3389/fnins.2018.00331. URL https://www.frontiersin.org/article/10.3389/fnins.2018.00331.

Wu, Y., Zhao, R., Zhu, J., Chen, F., Xu, M., Li, G., Song, S., Deng, L., Wang, G., Zheng, H., et al. Brain-inspired global-local learning incorporated with neuromorphic computing. *Nature Communications*, 13(1):1–14, 2022.

Xing, Y., Caterina, G. D., and Soraghan, J. A new spiking convolutional recurrent neural network (scrnn) with applications to event-based hand gesture recognition. *Frontiers in Neuroscience*, 14:590164, 2020.

Zhang, W. and Li, P. Spike-train level backpropagation for training deep recurrent spiking neural networks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 7800–7811, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f42a37d114a480b6b57b60ea9a14a9d2-Abstract.html.