ViT-LCA: A Neuromorphic Approach for Vision Transformers

Sanaz Mahmoodi Takaghaj
Department of Computer Science and Engineering, and
School of Engineering Design and Innovation
Penn State University
University Park, PA USA
sxm788@psu.edu

***Abstract*—The recent success of Vision Transformers has generated significant interest in attention mechanisms and transformer architectures. Although existing methods have proposed spiking self-attention mechanisms compatible with spiking neural networks, they often face challenges in effective deployment on current neuromorphic platforms. This paper introduces a novel model that combines vision transformers with the Locally Competitive Algorithm (LCA) to facilitate efficient neuromorphic deployment. Our experiments show that ViT-LCA achieves higher accuracy on ImageNet-1K dataset while consuming significantly less energy than other spiking vision transformer counterparts. Furthermore, ViT-LCA's neuromorphic-friendly design allows for more direct mapping onto current neuromorphic architectures.**

***Index Terms*—Vision Transformers (ViT), Sparse Coding, LCA, Encoder-Decoder Architecture, SNNs**

I. Introduction

Neuromorphic computing represents a paradigm shift in computing, characterized by its low-power processing capabilities and brain-inspired architectures [1]–[7]. This approach emulates biological neural networks through the use of Spiking Neural Networks (SNNs). One of the primary advantages of neuromorphic chips lies in their capacity for highly parallel and energy-efficient computations. By performing operations asynchronously and maintaining proximity between synapses and weight calculations, these systems significantly reduce data movement, thereby enhancing overall computational efficiency. These platforms integrate many-core systems capable of instantiating large populations of spiking neurons, enabling information processing that mimics the dynamics of biological neural systems. Additionally, by utilizing crossbar arrays and memristores [8]–[10] to store multi-bit quantities as conductance values, neuromorphic computing is particularly well-suited for efficiently evaluating matrix-vector-multiplications, which are fundamental to deep learning algorithms.

A particularly interesting model in neuromorphic computing is the Locally Competitive Algorithm (LCA) [11], [12], which is a computational model and learning algorithm that iteratively updates neuron activity to achieve a sparse representation of input data. This computational model has been implemented on recent neuromorphic platforms [2], [13], [14] for image reconstruction. The competitive mechanism inherent in LCA ensures that only a limited number of neurons become active at any given time, facilitating efficient coding of high-dimensional data. One proposal for leveraging the LCA in neuromorphic computing is the Exemplar LCA-Decoder [15]. Functioning as a single-layer encoder-decoder, this computational model iteratively updates neuron activity to identify a sparse representation of the input data (i.e, encoding) and then uses these neuron activities for classification tasks (i.e, decoding).

Recently, the Transformer architecture [16] and its variants have demonstrated impressive performance across a range of tasks, including natural language processing [17], [18] and computer vision [19]–[21]. This success is largely due to their ability to effectively capture long-range dependencies, a capability primarily attributed to the self-attention mechanism. Given the enormous computational requirements of transformer architectures, deploying these models on devices with limited resources remains a significant challenge. As a result, integrating transformer architectures with neuromorphic computing represents a promising research avenue. In particular, the combination of transformer architectures and LCA-based learning could lead to more efficient and biologically inspired artificial intelligence systems. However, this area remains largely unexplored.

This paper presents ViT-LCA, which leverages Vision Transformers (ViT) [19] to extract self-attention representations and incorporates these representations into an LCA-based SNN. This algorithm effectively addresses the challenges of deploying transformer models on energy-constrained neuromorphic platforms. The self-attention representations are extracted once and stored in non-volatile memory elements, enabling in-memory computation on neuromorphic systems that emphasize specialized operations and energy efficiency. Our approach consists of two stages. In the first stage, a transformer encoder generates self-attention representations from the input image. In the second stage, these representations are processed by a single-layer SNN that employs a LCA encoder-decoder architecture for classification tasks. In this study, we evaluate ViT-LCA on CIFAR-10 [22], CIFAR-100 [23] and ImageNet-1k [24] datasets and assess the effectiveness of integrating ViT's self-attention representation with the efficiency of sparse coding through LCA for deployment on neuromorphic systems. By inputting self-attention representations (contextual

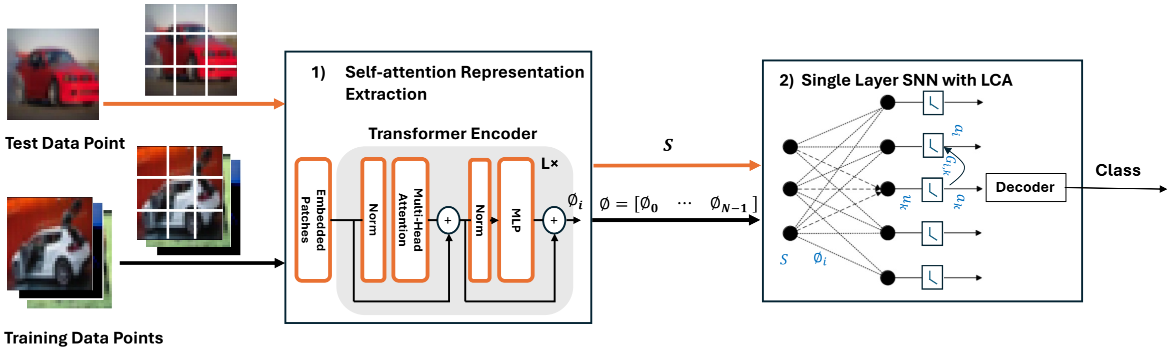

Fig. 1: ViT-LCA Architecture: Self-attention representations (ϕ_i) are extracted once and stored in the synaptic weights of a single-layer SNN for inference. The orange arrows indicate the inference process that follows the completion of training.

embeddings) derived from ViT into a single-layer SNN model, we achieved high classification accuracy while ensuring low computational overhead and high energy efficiency.

II. Related Work

There has been a growing interest in developing methods to reduce the computational requirements of Transformer models by integrating Transformer architectures with SNNs [25]–[28], which serve as brain-inspired counterparts to deep neural networks (DNNs). Spikformer [25] and the Spike-driven Transformer [26] introduce a spiking self-attention mechanism that utilizes spike-based representations for the Query, Key, and Value components, replacing traditional multiplication operations with low-energy addition operations. SpikingResformer [28] proposes a novel Dual Spike Self-Attention mechanism to address the challenges faced by previous methods in effectively extracting local features. Wang et al. [27] utilized the ANN-to-SNN conversion method and introduced the Random Spike Masking technique to improve the performance and energy efficiency of the Spiking Transformers.

ViT-LCA leverages transformer encodings within a novel Exemplar LCA-Decoder architecture [15] to deploy them on neuromorphic platforms. The ViT-LCA architecture differs from the standard ViT transformers, with a pre-trained ViT transformer being used to construct the dictionary employed by LCA-Decoder. Given the memory and computation demands of training transformers, ViT-LCA allows compute- and energy-restricted neuromorphic platforms to leverage advancements in state-of-the-art Transformer architectures.

III. ViT-LCA Encoder-Decoder

Fig. 1 illustrates the architecture of ViT-LCA. The training data undergoes pre-processing before being processed by the Vision Transformer (ViT) [19], where it passes through attention layers to extract self-attention representations. These representations are stored in the synaptic weights of a single-layer spiking neural network using LCA encoding, which are then employed to classify unseen test inputs via a decoder.

A. Extracting self-attention representations

To extract self-attention representations from an image using the Vision Transformer (ViT), the image is first split into fixed-size patches (Tokens). Each token is then embedded into a vector representation, which is augmented with positional encodings to retain spatial information. An additional learnable "classification token" is appended to the sequence of tokens. The resulting sequence of embeddings is then input into the Transformer Encoder as shown in Fig. 1. The extracted self-attention representations, denoted as ϕ_i, are subsequently used to construct a dictionary ϕ as in Eq. 5.

B. Single-Layer SNN with LCA Encoder and Decoder

We first provide an overview of the Exemplar LCA-Decoder algorithm [15], which utilizes the sparse coding algorithm [12] and the LCA algorithm [11] to represent the input signal S:

$$S = \sum_{i=0}^{M-1} \phi_i a_i + \varepsilon \tag{1}$$

where ϕ is a dictionary of self-attention representations (ϕ_i) and a_i represents the activation of the LIF neuron i. The term ε represents Gaussian noise. The membrane potential u_i of the LIF neuron is governed by a driving excitatory input b_i and an inhibition matrix (Gramian) G. The Gramian matrix enables stronger neurons to inhibit weaker neurons from activating, leading to a sparse representation.

$$\tau \dot{u}_i[k] + u_i[k] = b_i - \sum_{m \neq i}^{M-1} G_{i,m} a_m[k] \tag{2}$$

$$b_i = S\phi_i \tag{3}$$

$$G = \phi^T \phi \tag{4}$$

$$\phi = [\phi_0, \phi_1, \ldots, \phi_{M-1}] \tag{5}$$

In ViT-LCA, each ϕ_i represents a self-attention representation learned from a data point in the training dataset. This approach eliminates the need for dictionary learning, which differs from the original LCA, where Stochastic Gradient Descent (SGD) is used to learn and update the dictionary for each batch of input data. ViT-LCA benefits from the inherent optimization performed through LCA, without requiring any explicit and computationally expensive error backpropagation.

The thresholding function is:

$$\begin{aligned} a_i[k] &= T_\lambda(u_i[k]) \\ &= \begin{cases} u_i[k] - \lambda\,\mathrm{sign}(u_i[k]), & |u_i[k]| \geq \lambda \\ 0, & |u_i[k]| < \lambda \end{cases} \end{aligned} \tag{6}$$

Here, the threshold λ refers to the level that the membrane potential must exceed for the neuron to become active. Next, a decoder is designed to decode the sparse codes a_i and map them to C distinct classes from the training dataset. Later, the same decoder is used to generate class predictions for the unseen test data points.

Next, any (unseen) test data $S = S_{test}$ is mapped to the resulting M-dimensional space of self-attention embeddings (dictionary atoms ϕ_i). The corresponding activation codes a_i that approximate the new input are then determined, as outlined in Eq. 2 to Eq. 6.

Given a sufficient number of training data points *M*, each test data point will have a sparse coding representation $a = [a_0, a_1, a_2, ..., a_{M-1}]$, with the majority of the a_i values being zero. Ideally, for any test input S_{test}, the non-zero entries in the vector a would correspond exclusively to the dictionary atoms ϕ_i associated with class *c*, where *c* ranges from 1 to *C*. However, modeling errors and input signal noise may introduce small non-zero entries that are associated with multiple classes. To address this issue and better harness the relevance of neuron activations for each class, the "Maximum Sum of Activations" decoder is proposed. In this approach, the ℓ_1 norms of the activations for each class *c* are summed and the class with the highest value is determined:

$$Predicted\,Class = \underset{c}{argmax}(\sum_i \left| a_i^{(c)} \right|) \tag{7}$$

IV. Experiment Setup

We evaluated ViT-LCA using PyTorch [29] and the following datasets: CIFAR-10 [22], CIFAR-100 [23] and ImageNet-1K [24]. All datasets and the pre-trained Vision Transformer model were obtained using Torchvision [30].

A. Dataset Pre-Processing

After resizing the images to a uniform size of 224 x 224 pixels, the dataset is normalized to have a zero mean and a standard deviation of one to prepare it for input into the Vision Transformer.

B. CIFAR-10 and CIFAR-100

The CIFAR-10 and CIFAR-100 datasets each comprise a total of 60,000 images, with 50,000 designated for training and 10,000 for testing. Each image is represented in RGB format and has dimensions of 32 x 32 pixels. The CIFAR-10 dataset contains 10 classes, while the CIFAR-100 dataset is more complex, featuring 100 classes.

C. ImageNet-1K

The ImageNet-1K dataset comprises approximately 1.28 million training images and 50,000 validation images. For the purpose of constructing the dictionary, the training dataset was randomly split, and a subset of 50,000 samples was selected. The ILSVRC2012 validation dataset was exclusively used for testing, ensuring that it was not utilized in any capacity prior to evaluation. This dataset contains 1000 classes.

V. Evaluation

Table I presents the workload and energy estimates, along with accuracy performance across all tested datasets. All experiments were conducted using the hyperparameters specified in Table II, which were selected to achieve nearly 100% training accuracy on all datasets; the results reflect the corresponding test accuracy. Further optimization of these hyperparameters per dataset may enhance accuracy even further. Additionally, Table I indicates that the "Maximum Sum of Activations" decoder outperforms the "Maximum Activation" decoder across all datasets. The energy estimates were calculated based on the floating-point operations performed during inference and the anticipated energy consumption per floating-point operation, which is discussed further in the following section.

A. Workload and Energy Efficiency Analysis

In this section, we conduct a comprehensive evaluation of the workload and energy efficiency of ViT-LCA. We assess workload efficiency by estimating the number of Floating-Point Operations (FLOPs) involved, followed by a detailed discussion of energy consumption.

Computing b_i as described in Eq. 3 requires $N * M$ multiplications and $(N-1)*M$ addition, where N is the size of the self-attention representation and M is the size of the dictionary (which also corresponds to the number of neurons). The inhibition signal calculation in Eq. 2 involves $M^2 - M$ multiplications and $M^2 - 2M$ additions. Additionally, the leakage term introduces M multiplication operations, while $3M$ additions are required to combine these terms and update the neurons' membrane potentials, as specified in Eq. 2. Furthermore, computing the Gramian matrix G in Eq.4 entails $\frac{M(M+1)N}{2}$ multiplications and $\frac{M(M+1)(N-1)}{2}$ additions.

$$\underset{(Training)}{FLOPs} = \frac{M(M+1)(2N-1)}{2} \tag{8}$$

The Gramian matrix computation, which is considered part of the training cost and performed once per task (dataset), is excluded from the inference cost. Thus, the total floating-point operations required per time step K for inference is given by:

$$\underset{(Inferenec)}{FLOPs} = K(\frac{(2N-1)M}{K} + 2M^2 + M) \tag{9}$$

Note that b_i is computed once per input data and remains constant across iterations. Finally, the expected sparsity through LCA is factored in by redefining M, the number of active neurons, as $\hat{M}$, which represents the average number of spiking neurons whose $a_m \neq 0$.

$$\underset{(Inferenec)}{FLOPs} = K(\frac{(2N-1)M}{K} + 2M\hat{M} + M) \tag{10}$$

The estimated training and inference FLOPs for ViT-LCA are presented in Table I. "TFLOPs" denotes the Tera FLOPs required for training (computing the Gramian matrix), while

TABLE I: Top-1 Test Accuracy Scores and Workload/ Energy Estimates for ViT-LCA.

					Accuracy	
					Decoding Methods	
Dataset	**Transformer Model**	**Training TFLOPs**^{a}	**Inference GFLOPs**^{b}	**Energy**	$\underset{i}{max}\,\{a_i\}$	$\underset{c}{max}\sum_i \left\lvert a_i^{(c)} \right\rvert$
CIFAR-10	ViT-B/16	1.92	2.08	0.19 mJ	90.53%	**95.63%**
CIFAR-100	ViT-B/16	1.92	2.08	0.19 mJ	81.63%	**81.80%**
ImageNet-1K	ViT-B/16	1.92	2.08	0.19 mJ	71.51%	**80.24%**

[a]Tera FLOPs, [b]Giga FLOPs

TABLE II: Hyperparameters

Symbols	Description	Value
-	Patch size	16*16
L	Self-attention layers	12
M	Dictionary size	50K
$\hat{M}$	Average number of spiking neurons	200
N	Self-attention representation length	768
λ	Threshold	2
τ	Leakage term	100
K	Time steps	100

"GFLOPs" represents the Giga FLOPs for inference operations. Training FLOPs accounts for all training data points, whereas inference FLOPs is estimated for a single test input.

In addition to enhancing workload efficiency, ViT-LCA leverages recent advancements in in-memory computing through memristive crossbar arrays. Memristive crossbars can perform multiplication operations based on Kirchhoff's Current Law and Ohm's Law (I = V · G, where I is the current, V is the input voltage, and G is the conductance of each memristor), thereby enhancing energy efficiency and minimizing the area footprint. Using Resistive Random Access Memory (RRAM) crossbar arrays, Yao et al. [31] reported an energy efficiency of 11 Tera OPs per Watt for floating point multiply-and-accumulate (MAC) operations. This is equivalent to each floating point operation consuming at most approximately 9.09×10^{-14} joules of energy.

Fig. 2 illustrates an original hardware mapping of ViT-LCA, where the ϕ_i values are represented as the conductance of memristors in each column, used to perform neuron excitatory input multiplications as shown in Eq. 3. Inputs are preprocessed in the input peripheral circuits, while neuron dynamics and thresholding—defined in Eq.2 and Eq.6—are implemented in the output peripheral circuits located in the neural cores. The calculation of the Gramian matrix, which serves as the training head in this algorithm, can be performed offline, with the results stored in a lookup table (LUT) in the on-chip memory of a host processor that interacts with the neural cores via a digital interface.

VI. Discussion

This study serves as a proof of concept, demonstrating the potential for a uniform algorithm applicable across various transformer architectures and datasets. However, it currently exhibits limitations in accuracy compared to state-of-the-art transformer models, and further improvements could be realized by exploring alternative transformer architectures for self-attention representation extraction. We utilized the ViT-B/16 transformer model developed by Dosovitskiy et al. [19], which has reported accuracies of 98.13%, 87.13%, and 77.91% on CIFAR-10, CIFAR-100, and ImageNet-1K, respectively. In comparison, our results (TableI) were slightly lower on CIFAR-10 and CIFAR-100, but we achieved a higher accuracy on ImageNet. We believe this discrepancy arises from their use of a higher resolution of 384 x 384 pixels and fine-tuning on CIFAR-10 and CIFAR-100, whereas our model was pre-trained solely on ImageNet-1K without fine-tuning on the CIFAR-10 and CIFAR-100 datasets, which we opted for to avoid additional computational overhead.

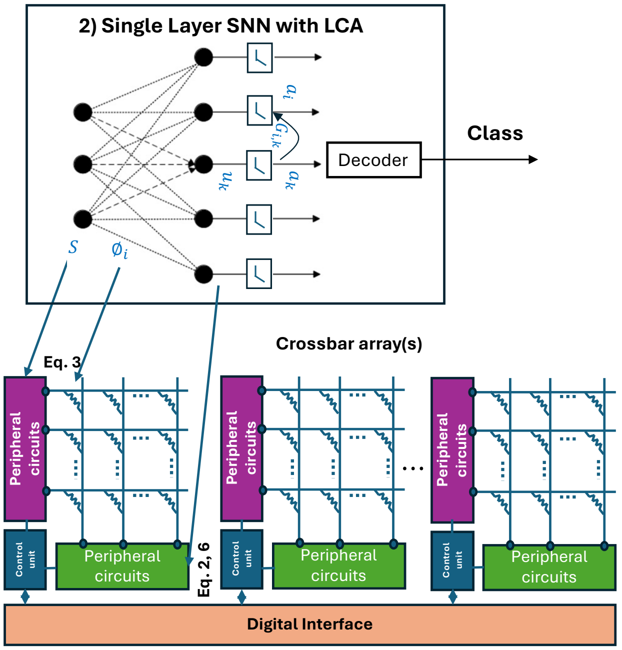

Fig. 2: An original hardware implementation of ViT-LCA.

In comparisons with spiking transformers [25], [27], [28], our model demonstrated superior performance across all three datasets, achieving higher accuracy than Spikformer while consuming only 0.19 mJ of energy. Spikformer recorded accuracies of 95.51% on CIFAR-10, 78.21% on CIFAR-100, and 74.81% on ImageNet-1K, with an estimated inference energy of 21.48 mJ [28]. Additionally, when compared to the Masked Spiking Transformer [27] based on Swin transformer [32], our model achieved higher accuracy on ImageNet, but showed lower accuracy on CIFAR-10 and CIFAR-100, where the Masked Spiking Transformer recorded accuracies of 97.06%, 86.73%, and 77.88%, respectively, with time steps of 128. SpikingResformer [28] achieved Top-1 accuracy of 79.40% on ImageNet-1K, with an energy consumption of 14.76 mJ. With the highest accuracy results and lowest energy consumption on ImageNet, along with direct deployment on neuromorphic systems, ViT-LCA demonstrates significant potential for future implementation on neuromorphic platforms.

VII. Conclusion

In this work, we explored the integration of LCA with Vision Transformers (ViT) for the first time and evaluated its performance on classification tasks using the CIFAR-10, CIFAR-100, and ImageNet-1K datasets. We leveraged self-attention representations extracted through ViT within an LCA encoder-decoder framework, eliminating the need to train a dictionary as required in the original LCA implementation. Additionally, we demonstrated how this approach can be effectively mapped to memristor crossbar arrays to leverage efficient in-memory computing on neuromorphic systems. This study underscores the potential for further research, including the investigation of additional transformer architectures to achieve even higher accuracy results.

References

[1] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, and others, "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017.

[2] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, and others, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[3] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[4] S. Höppner, Y. Yan, A. Dixius, S. Scholze, J. Partzsch, M. Stolba, F. Kelber, B. Vogginger, F. Neumärker, G. Ellguth, and others, "The SpiNNaker 2 processing element architecture for hybrid digital neuromorphic computing," *arXiv preprint arXiv:2103.08392*, 2021.

[5] R. Khaddam-Aljameh, M. Stanisavljevic, J. F. Mas, G. Karunaratne, M. Brändli, F. Liu, A. Singh, S. M. Müller, U. Egger, A. Petropoulos, and others, "HERMES-core—A 1.59-TOPS/mm² PCM on 14-nm CMOS in-memory compute core using 300-ps/LSB linearized CCO-based ADCs," *IEEE J. Solid-State Circuits*, vol. 57, no. 4, pp. 1027–1038, 2022.

[6] M. Le Gallo, R. Khaddam-Aljameh, M. Stanisavljevic, A. Vasilopoulos, B. Kersting, M. Dazzi, G. Karunaratne, M. Brändli, A. Singh, S. M. Mueller, and others, "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," *Nature Electronics*, vol. 6, no. 9, pp. 680–693, 2023.

[7] J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He, and others, "Towards artificial general intelligence with hybrid Tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.

[8] Sebastian, Abu, Le Gallo, Manuel, Khaddam-Aljameh, Riduan, and Eleftheriou, Evangelos. "Memory devices and applications for in-memory computing." *Nature Nanotechnology* 15, no. 7 (2020): 529–544.

[9] M. Rao, H. Tang, J. Wu, W. Song, M. Zhang, W. Yin, Y. Zhuo, F. Kiani, B. Chen, X. Jiang, and others, "Thousands of conductance levels in memristors integrated on CMOS," *Nature*, vol. 615, no. 7954, pp. 823–829, 2023.

[10] Aguirre, Fernando, Sebastian, Abu, Le Gallo, Manuel, Song, Wenhao, Wang, Tong, Yang, J. Joshua, Lu, Wei, Chang, Meng-Fan, Ielmini, Daniele, Yang, Yuchao, et al. *Hardware implementation of memristor-based artificial neural networks. Nature Communications*, vol. 15, no. 1, p. 1974, 2024.

[11] C. Rozell, D. Johnson, R. Baraniuk, and B. Olshausen, "Locally Competitive Algorithms for Sparse Approximation," in *Proc. 2007 IEEE Int. Conf. Image Processing*, vol. 4, pp. IV-169–IV-172, 2007.

[12] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits," *Neural Computation*, vol. 20, no. 10, pp. 2526–2563, 2008.

[13] Fair, Kaitlin L., Mendat, Daniel R., Andreou, Andreas G., Rozell, Christopher J., Romberg, Justin, and Anderson, David V. (2019). Sparse coding using the locally competitive algorithm on the TrueNorth neurosynaptic system. *Frontiers in Neuroscience*, 13, 754.

[14] Sheridan, Patrick M., Cai, Fuxi, Du, Chao, Ma, Wen, Zhang, Zhengya, and Lu, Wei D. (2017). Sparse coding with memristor networks. *Nature Nanotechnology*, 12(8), 784–789.

[15] S. M. Takaghaj and J. Sampson, "D-SELD: Dataset-Scalable Exemplar LCA-Decoder," *Neuromorphic Computing and Engineering*, 2024.

[16] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[17] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[22] Alex Krizhevsky and Geoffrey Hinton, *CIFAR-10 (Canadian Institute For Advanced Research)*, 2009. Available at: https://www.cs.toronto.edu/~kriz/cifar.html.

[23] A. Krizhevsky, "CIFAR-10 and CIFAR-100 datasets," [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html. [Accessed: July, 2024].

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[25] Haokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural networks meet transformers. In *Proceedings of the International Conference on Learning Representations*, pages 1–17, 2023.

[26] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In *Advances in Neural Information Processing Systems*, pages 1–20, 2023.

[27] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. Masked spiking transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1761–1771, 2023.

[28] Xinyu Shi, Zecheng Hao, and Zhaofei Yu. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2024.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and others, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] T. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and others, "Torchvision: Datasets, transforms and models for computer vision," Available: https://pytorch.org/vision/stable/index.html. [Accessed: July, 2024].

[31] Yao, Peng, Wu, Huaqiang, Gao, Bin, Tang, Jianshi, Zhang, Qingtian, Zhang, Wenqiang, Yang, J. Joshua, and Qian, He. "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.