Generative Data for Neuromorphic Computing

Anthony Baietto[1], Trevor J. Bihl[2]

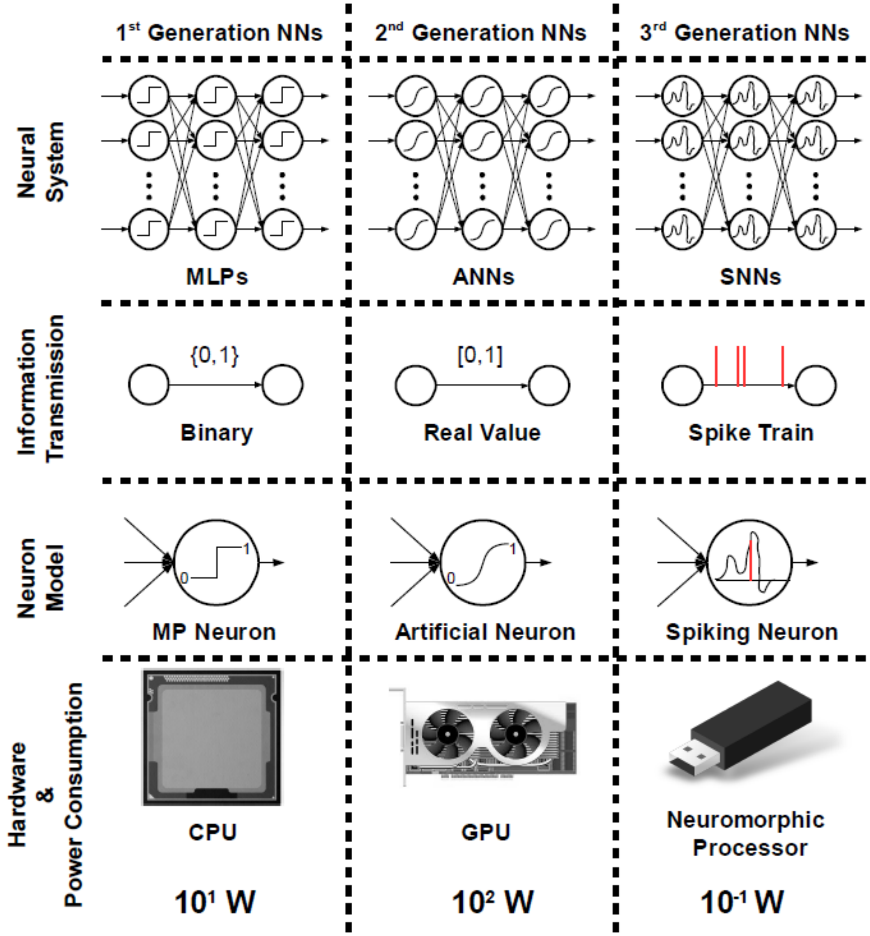
Department of Computer Science and Engineering, Ohio State University[1]
Sensors Directorate, U.S. Air Force Research Laboratory[2]

Outline

- Spiking Neural Networks (SNNs) and Neuromorphic Sensors
- Generating Synthetic Samples
- Results
- Conclusions

Spiking Neural Networks (SNNs)

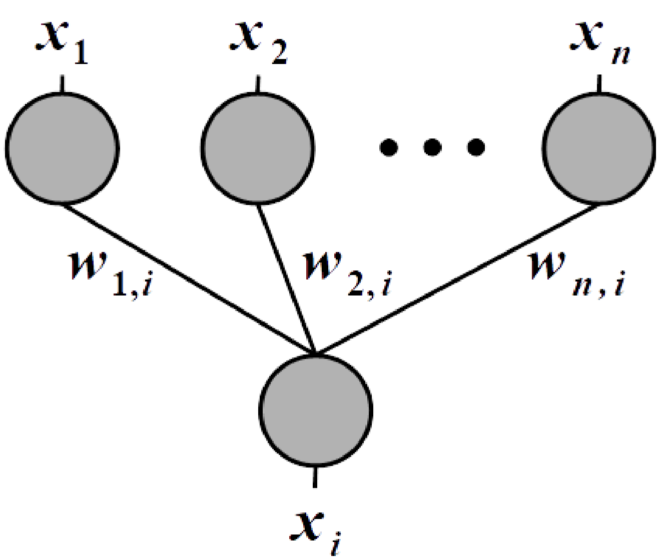
- Biologically inspired 3rd generation neural networks
- Neurons communicate via discrete pulses over time
- SNN processing consumes less power when realized on neuromorphic processors



ANNs vs SNNs

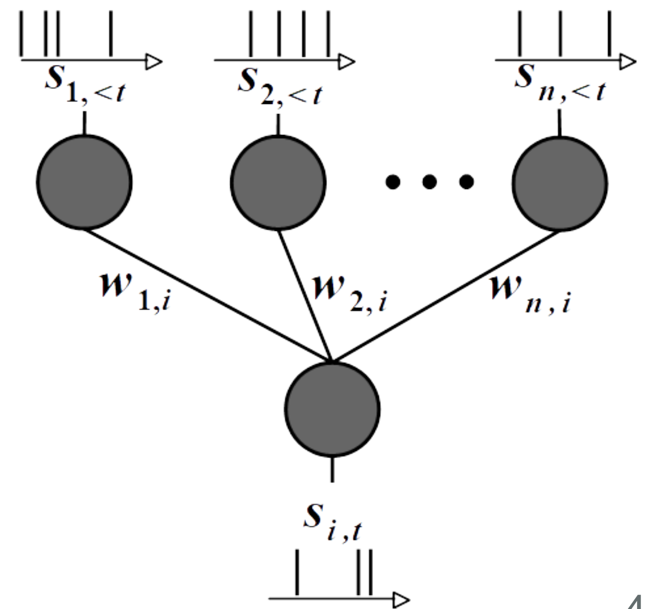
ANNs

- Operate on continuous values $x_1, x_2, \ldots, x_n$
- Information propagates instantaneously



SNNs

- Operate on discrete spike trains $S_1, S_2, \ldots, S_n$
- Must be run over a period of time



ANNs vs SNNs

$$E = L(y, \hat{y})$$

$$net_j = \sum_i w_{ij} x_i + b$$

$$o_j = \varphi(net_j)$$

$$\delta_j = \begin{cases} \frac{\partial L(y, o_j)}{\partial o_j} \frac{d\varphi(net_j)}{dnet_j} & j \text{ output} \\ \left(\sum_k w_{jk}\delta_k\right) \frac{d\varphi(net_j)}{dnet_j} & j \text{ hidden} \end{cases}$$

$$\Delta w_{ij} = -\eta o_j \delta_j$$

Input $s_i(t) = \sum_f \delta\left(t - t_i^{(f)}\right)$

$$a_i(t) = (\epsilon * s_i)(t) \quad \epsilon(t) = \frac{t}{\tau_s} exp\left(1 - \frac{t}{\tau_s}\right)\Theta(t)$$

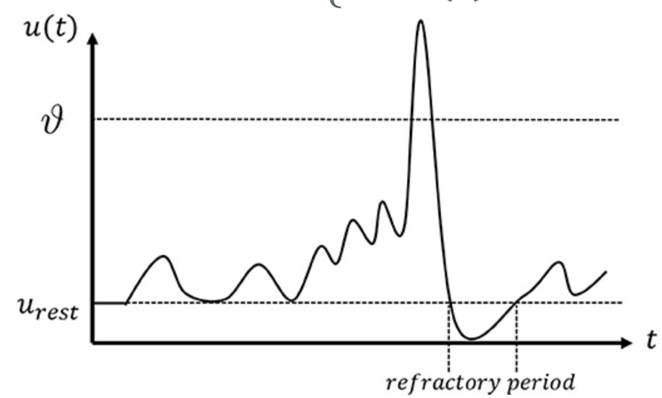
$$v_i(t) = (v * s)(t) \quad \mathrm{v}(t) = -2\vartheta exp\left(1 - \frac{t}{\tau_r}\right)\Theta(t)$$

$$u(t) = \sum_i w_i a_i(t) + v_i(t)$$

$$f_s(u) : u \to s$$

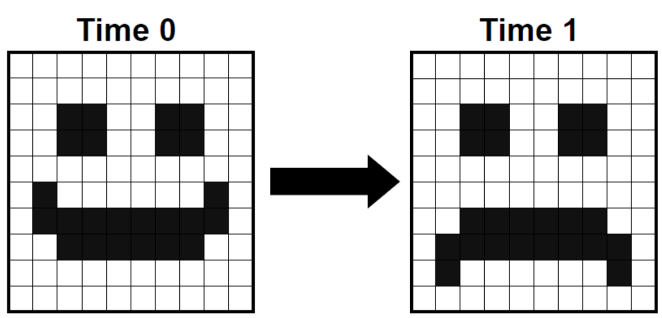
$$s(t) \coloneqq s(t) + \delta\left(t - t^{(f+1)}\right)$$

$$t^{(f+1)} = min\{t : u(t) = \vartheta, t > t^{(f)}\}$$



Neuromorphic Sensors

- Event based rather than frame based (less communication $\rightarrow$ lower energy consumption)
- $[\boldsymbol{x}$ coordinate, $\boldsymbol{y}$ coordinate, $\boldsymbol{t}$ timestep, $\boldsymbol{p}$ $+ / -$ change in luminance$]$

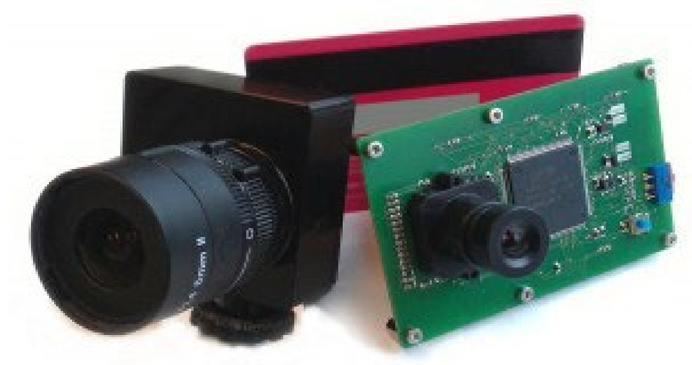


Frame-based Sensor (all 100 pixels communicated):

(1, 10, #000000)	(2, 10, #000000)	•••	(9, 10, #000000)	(10, 10, #000000)
(1, 9, #000000)	(2, 9, #000000)	•••	(9, 9, #000000)	(10, 9, #000000)
⋮	⋮	⋱	⋮	⋮
(1, 2, #000000)	(2, 2, #FFFFFF)	•••	(9, 2, #FFFFFF)	(10, 2, #000000)
(1, 1, #000000)	(2, 1, #000000)	•••	(9, 1, #000000)	(10, 1, #000000)

Neuromorphic Sensor (8 events communicated):

(2, 4, -, 1) (2, 2, +, 1)
(2, 5, -, 1) (2, 3, +, 1)
(9, 4, -, 1) (9, 2, +, 1)
(9, 5, -, 1) (9, 3, +, 1)



https://inilabs.com/products/

Issues

- Scarcity of neuromorphic datasets hinders testing and development of neuromorphic computing models
- Inadequate standardization and collaboration between neuromorphic computing community and traditional computer science complicates dataset proliferation

Outline

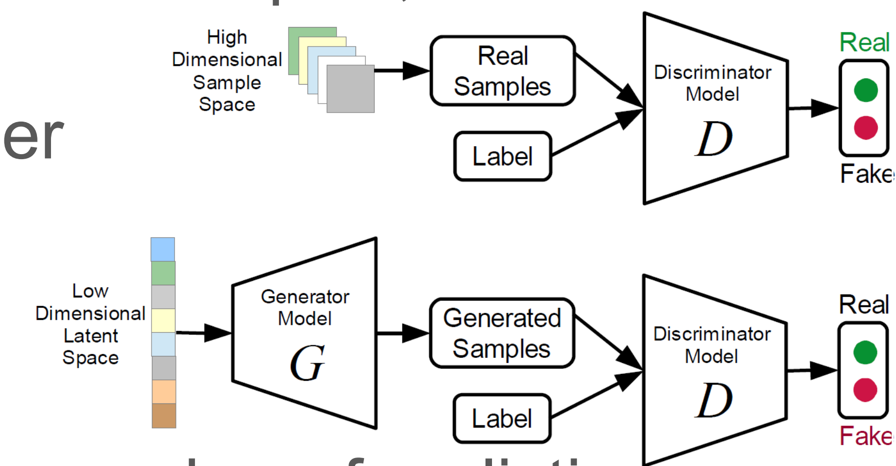
- Spiking Neural Networks (SNNs) and Neuromorphic Sensors
- Generating Synthetic Samples
- Results
- Conclusions

Our approach

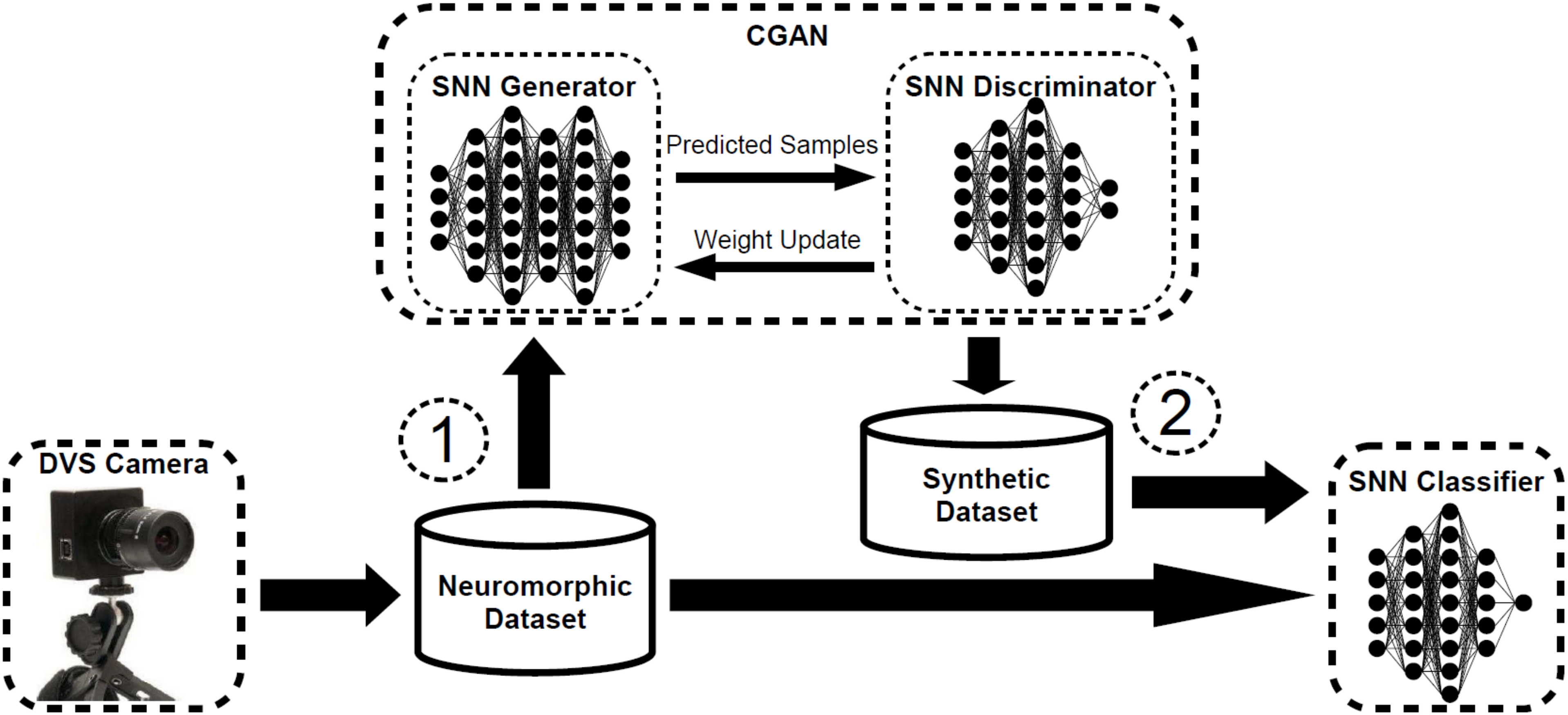
- Leverage generative models to expand existing neuromorphic datasets with synthetic samples
- Ensure quality sample generation via comparisons to real samples across multiple metrics
- Show SNN training improvements on augmented training datasets

Conditional Generative Adversarial Networks (CGANs)

- Adversarial learning paradigm in which a generator model G synthesizes artificial samples, and a discriminator model *D* classifies samples as either real or generated
- Once converged, the generator model can be used to create an arbitrary number of realistic samples



The Ohio State University

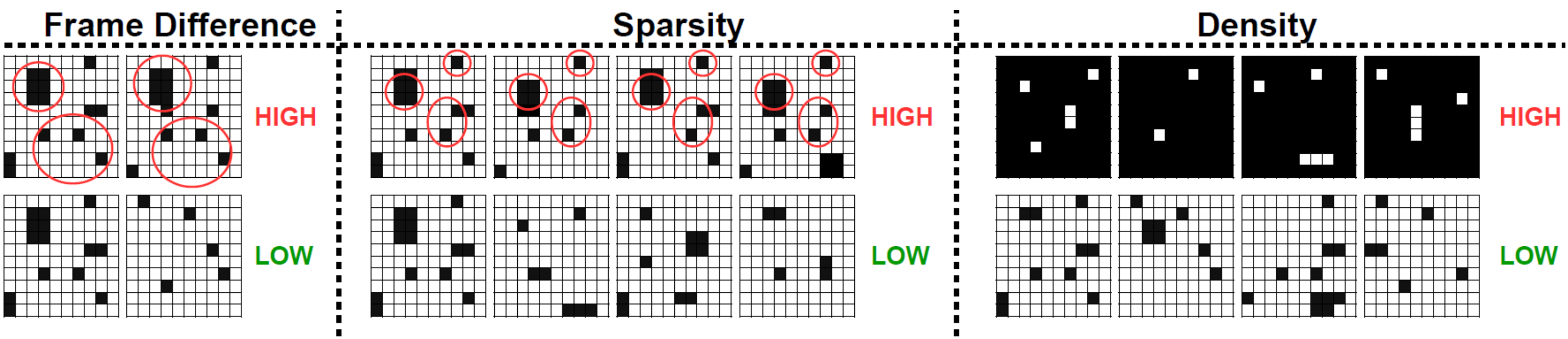


(1) Initial neuromorphic dataset is used to train CGAN

(2) Synthetic dataset is used in tandem for SNN training

Generative Sample Metrics

- **Frame Difference** – Cosine Similarity between adjacent timesteps
- **Sparsity** – Average number of eventer per pixel
- **Density** – Average number of events per frame

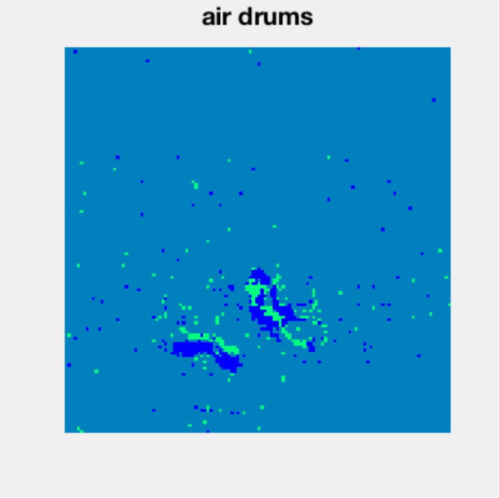


Outline

- Spiking Neural Networks (SNNs) and Neuromorphic Sensors
- Generating Synthetic Samples
- Results
- Conclusions

Setup

- IBM DVSGesture dataset
  - Neuromorphic dataset of 29 subjects performing hand motions in front of a DVS camera
- Experiments performed using ¼, ½, and entire dataset to simulate affect of insufficient access to training data



A. Amir *et al.*, "A Low Power, Fully Event-Based Gesture Recognition System," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 7388-7397, doi: 10.1109/CVPR.2017.781.

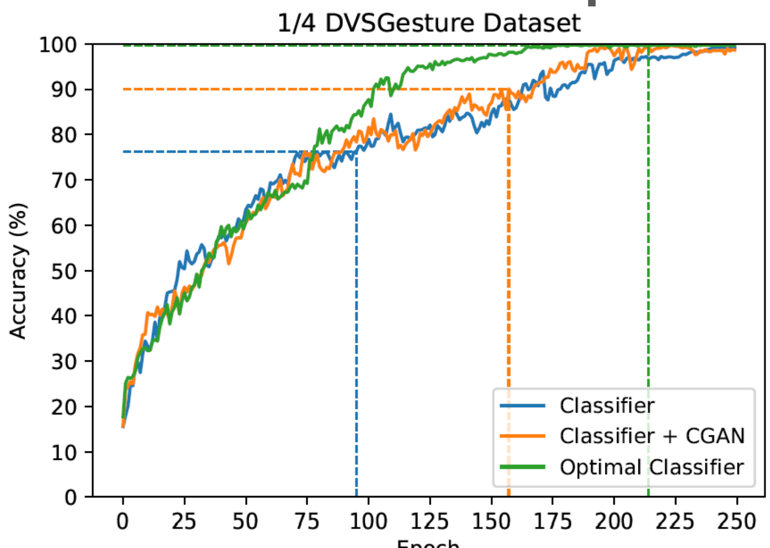
Generative Samples Metrics

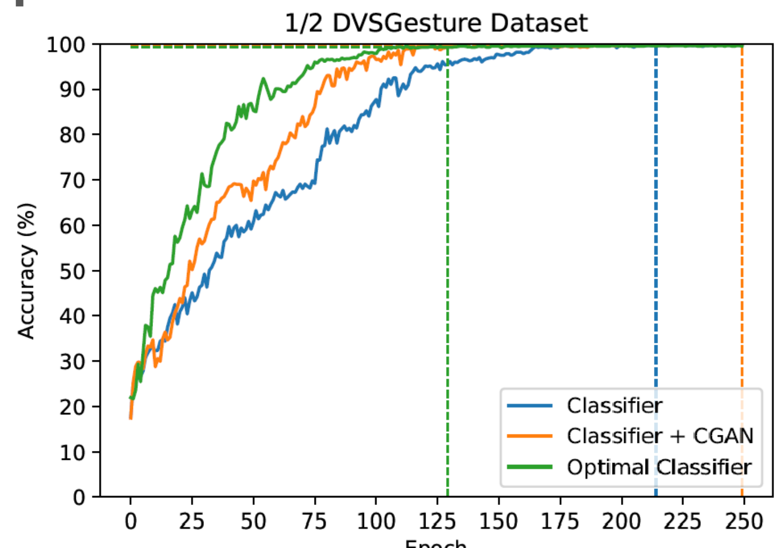
- Generative samples have similar distribution of spikes to real samples

Dataset	Frame Difference	Sparsity	Density
DVSGesture	0.07105	0.00152	0.00152
CGAN 1/4 DVSGesture	0.04033	0.00424	0.00424
CGAN 1/2 DVSGesture	0.00020	0.00030	0.00030
CGAN Entire DVSGesture	0.00010	0.00030	0.00030

Generative Samples Training Impact

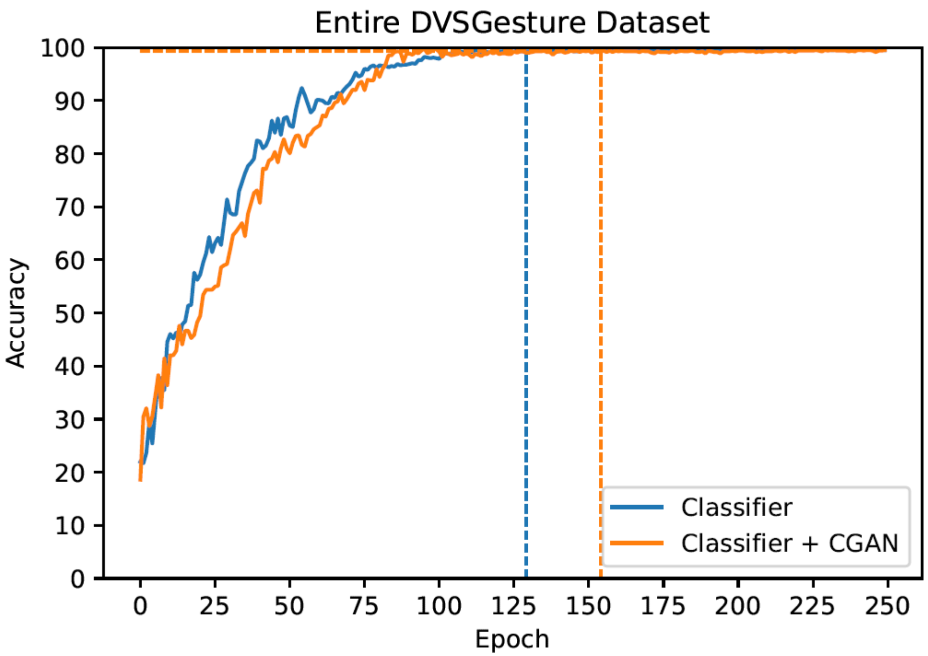
- Access to generative samples improves training performance
- CGAN sample generation quality improves with more sample exposure





Generative Samples Training Impact

- As # of generated samples exceeds # of CGAN training samples, impact on training performance decreases



Outline

- Spiking Neural Networks (SNNs) and Neuromorphic Sensors
- Generating Synthetic Samples
- Results
- Conclusions

Conclusions

- SNNs and neuromorphic sensors are highly energy efficient
- Generative data can be used to augment scarce spiking datasets
- SNN training benefits from access to generative samples