Toward Robust Spiking Neural Networks

Anthony Baietto[1] (baietto.2@osu.edu), Christopher Stewart[1], Trevor J. Bihl[2]

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

I. Introduction

- Spiking neural networks (SNNs) transmit output signals only after their input signals exceed the activation threshold
- Processors designed for SNNs consume much less power[1] than processors designed for artificial neural networks (ANNs) making SNNs a promising architecture for energy-constrained datacenters and Internet of Things (IoT) devices
- SNN training difficult because backpropagation cannot infer the changing subset of transmitting neurons and the duration of their transmissions
- State-of-the-art SNN platforms provide platform-specific mechanistic models to characterize neuron activations; however, these models are often heavily tied to the specific spike distribution used for training
- **In this poster, we present a platform-agnostic approach that automatically learns neuron activations from observations using established approximations, combined with a generative adversarial network (GAN) to augment the training dataset with broader spike distribution data**

II. Design

- SNN operations rely on content of the input data as well as its distribution in time e.g. where an ANN need only to classify the subject within a static image, a SNN would need to additionally process the distribution of pixels across the presentation time of that image
- Spiking classification is more complex due to additional time dimension introducing further distinctions between samples and ultimately affecting model robustness
- The generative adversarial network (GAN) architecture melds exceptionally well with the considerations SNNs require
- Generated samples can be used to enrich datasets to provide a broader spectrum of spike distributions leading to a more robust spiking model

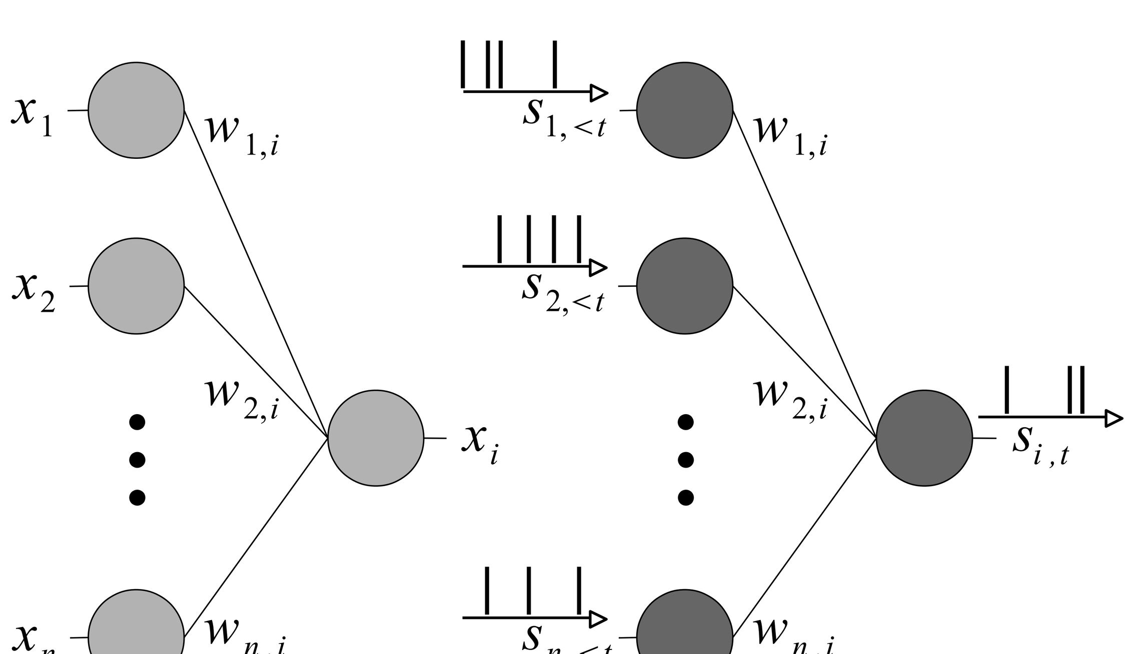
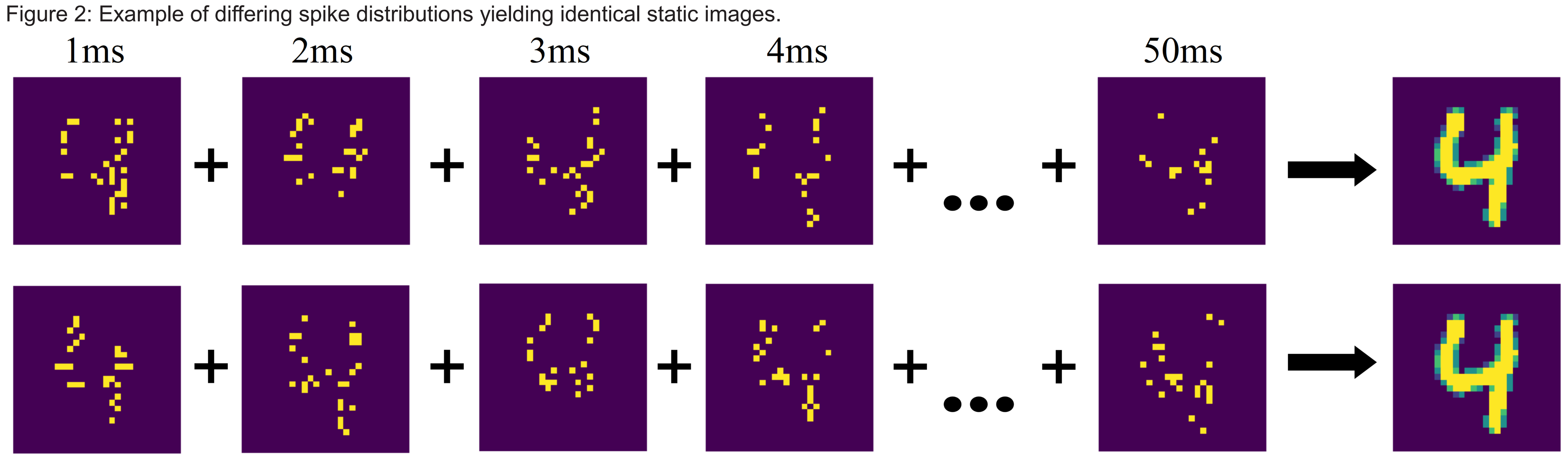


Figure 1: Comparison of neuron operations: an ANN which operates on real inputs $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ and outputs real number $\mathbf{x}_i$ as opposed to an SNN that acts on discrete spike trains $\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n$ and outputs spike train $\mathbf{s}_i$ where each spike train is the sequence of neuron firing times.

Figure 2: Example of differing spike distributions yielding identical static images.



- Samples of varying spike distributions can be easily generated that still satisfy the higher-level constraints of the problem e.g. images of a subject can be generated with varying distributions of spike events throughout the image presentation duration
- Our methodology is as follows:
  1) Utilizing the SLAYER[2] framework, we seeded a spiking GAN with the weights of a converged spiking classifier
  2) The generator of the trained GAN is then used to augment the training dataset for further training of the classifier
- Samples used to augment the training dataset were tailored based on the models performance using 3 schemes:
  a) Samples from each class were added equally
  b) Samples from the 3 worst performing classes added in an adhoc fashion
  c) Number of samples correlated to the relative performance of each class

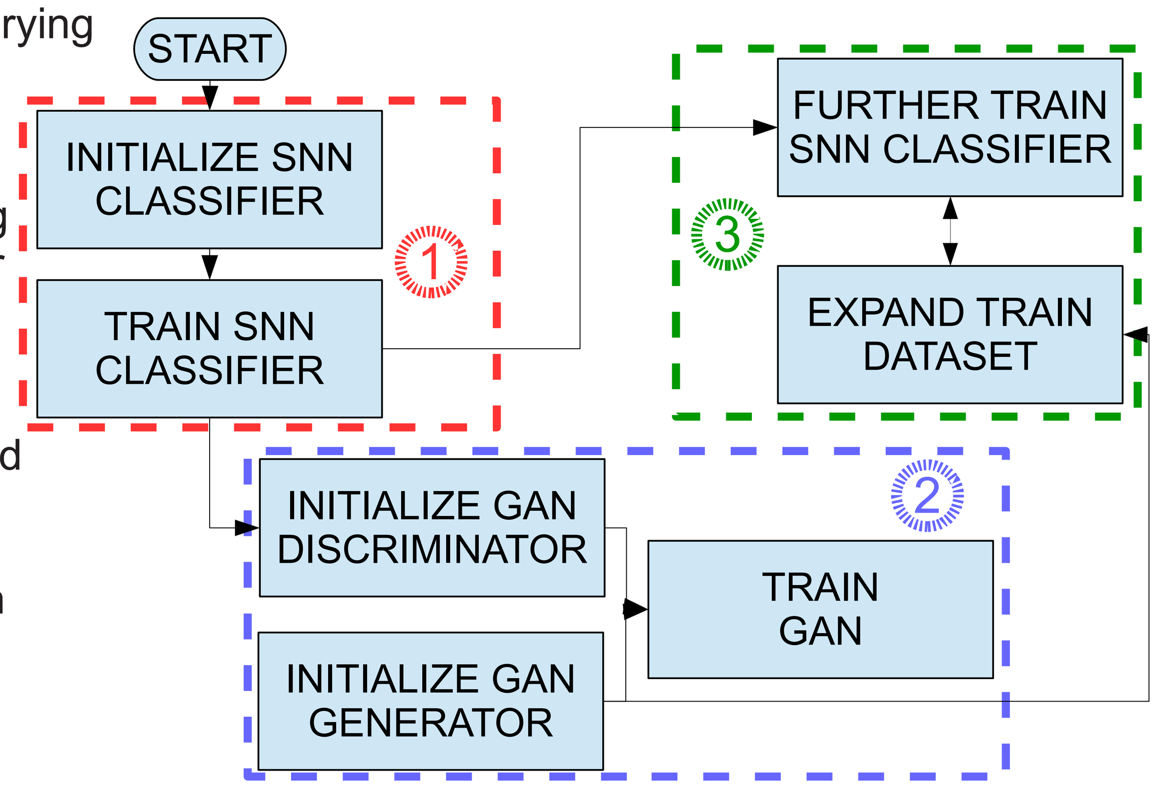


Figure 3: Our process: (1) An SNN classifier is trained to convergence (2) The SNN classifier weights seed the discriminator of the GAN to be trained (3) The trained generator is used to augment the training dataset for further SNN classifier training

III. Results

- Our first experiment focused on a model's ability to adapt to changing spike distributions during training
- We explored each model's training response when swapping to fewer and more spikes equating to half and double compared to the training distribution

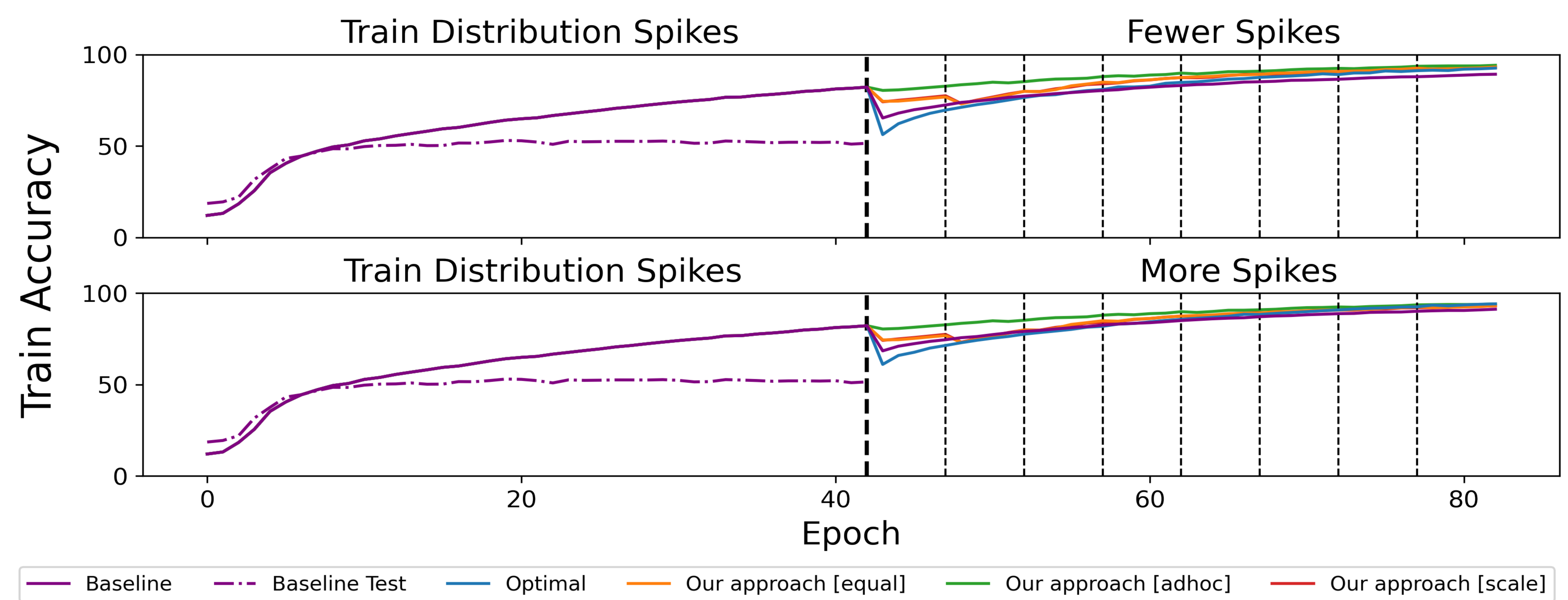


Figure 4: Train accuracies of baseline vs dataset-augmented models. Dashed vertical lines indicate where the training set was augmented with generated data.

Table 1: Test accuracy vs spike distribution with std. error across 6 folds.

Model	Testing Spike Distribution: Fewer Spikes	Train Dist. Spikes	More Spikes
Baseline	37.76 ± 0.34	52.67 ± 0.31	42.73 ± 0.52
Our approach [equal]	39.09 ± 0.25	**54.57 ± 0.27**	44.07 ± 0.52
Our approach [adhoc]	**39.33 ± 0.28**	54.25 ± 0.30	**44.67 ± 0.52**
Our approach [scale]	38.52 ± 0.39	54.05 ± 0.32	43.51 ± 0.29

- Our models quickly responded to the changing spike distribution and achieved a higher training accuracy
- Our second experiment focused on improving model robustness against samples different from the training spike distribution
- Intuitively, all models performed worse as the samples drifted further from the training spike distribution
- Our models outperformed the baseline classifier by an average of 1.80% in addition to having an average 1.02% lesser reduction in accuracy moving away from the training spike distribution
- Training dataset augmentation improved SNN model robustness without hindering training performance or sacrificing precision on the original spike distribution

IV. Conclusions

- Conventional SNN training methods do not ensure adequate generalization across spike distributions
- Training data richness is an important caveat for SNN training considering the additional variation of samples across the time dimension
- **Our preliminary results show detectable improvements in model performance when exposed to dissimilar samples during training**

V. References

[1] A. S. Kucik and G. Meoni, “Investigating spiking neural networks for energy-efficient on-board ai applications. a case study in land cover and land use classification,” in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 2020–2030.

[2] S. B. Shrestha and G. Orchard, “Slayer: Spike layer error reassignment in time,” 2018. [Online]. Available: https://arxiv.org/abs/1810.08646

VI. Acknowledgments

The work herein is the work of the authors and does not represent any position of the Air Force Research Laboratory, US Air Force, Department of Defense, or US Government. This work was approved for public release. This research was funded by SLAKE contract FA8650-19-C-1692. This work was supported in part by high-performance computer time and resources from the DoD High Performance Computing Modernization Program.

The Ohio State University

AFRL THE AIR FORCE RESEARCH LABORATORY