

# The Brain Computes Using Time, and So Should Neural Networks

Jason K. Eshraghian  
Assistant Professor, ECE, UC Santa Cruz

3 April 2025



## How can I help you today?

**Come up with concepts**

for a retro-style arcade game

**Design a database schema**

for an online merch store

**Help me debug**

a linked list problem

**Brainstorm names**

for an orange cat we're adopting from the she...

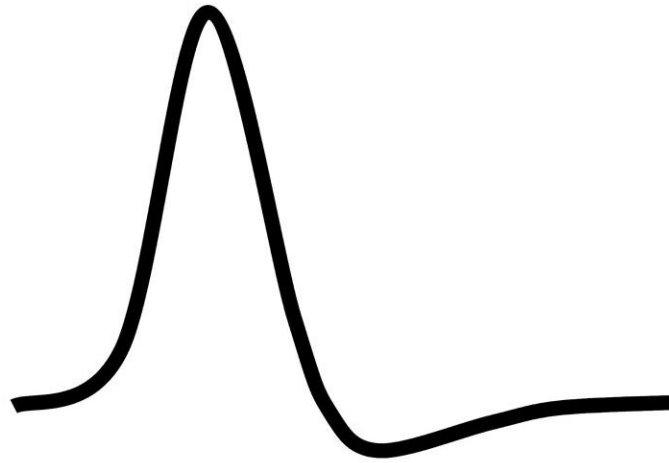


Message ChatGPT...



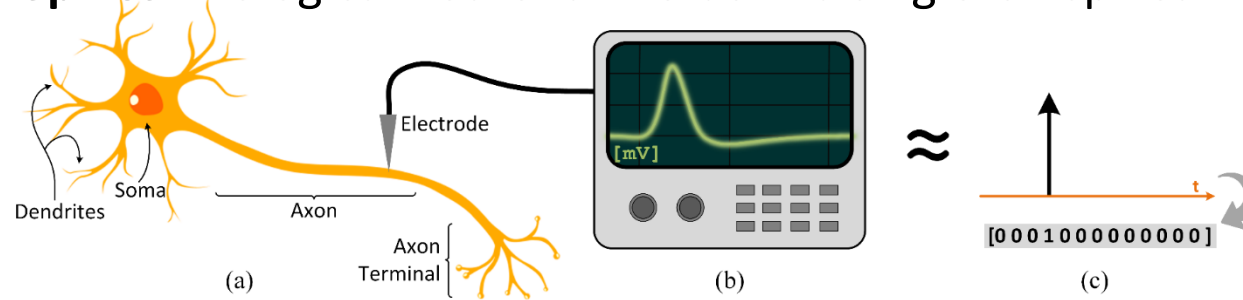
ChatGPT can make mistakes. Consider checking important information.

The fundamental unit of computation in the brain is a spike



# Toward Biological Networks: The Three S's

**Spikes:** Biological neurons interact via single-bit spikes



---

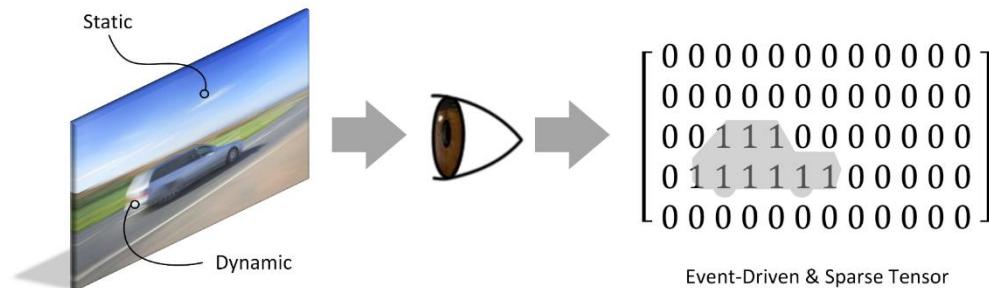
**Sparsity:** Biological neurons spend most of their time at rest, setting most activations to *zero* at any given time

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 5]

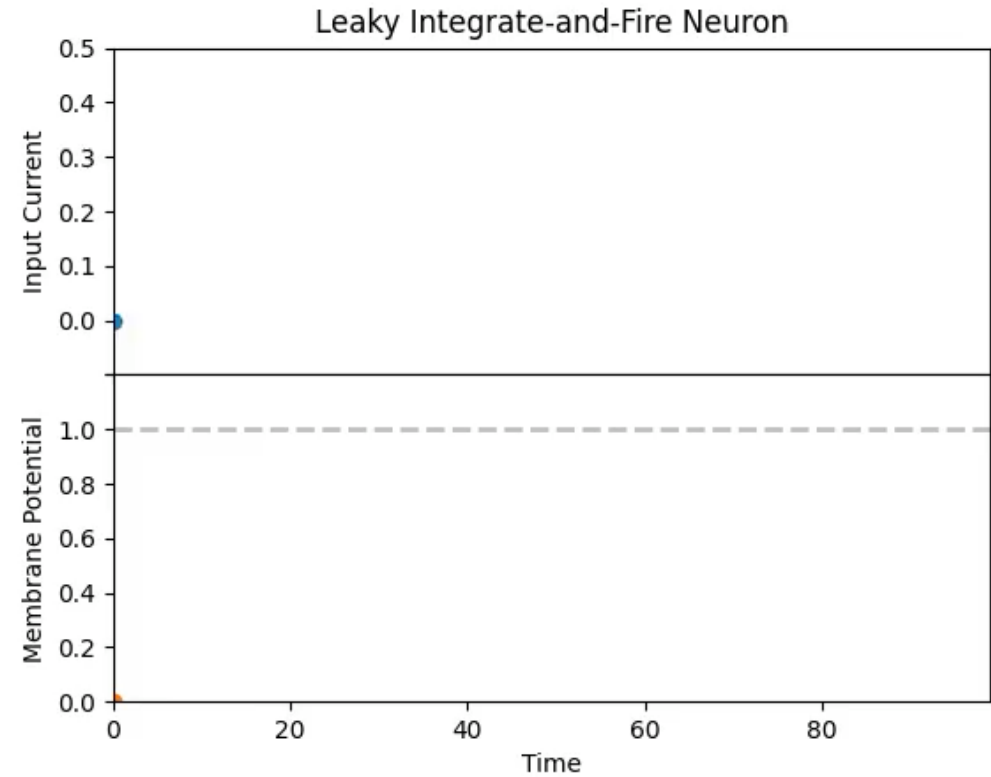
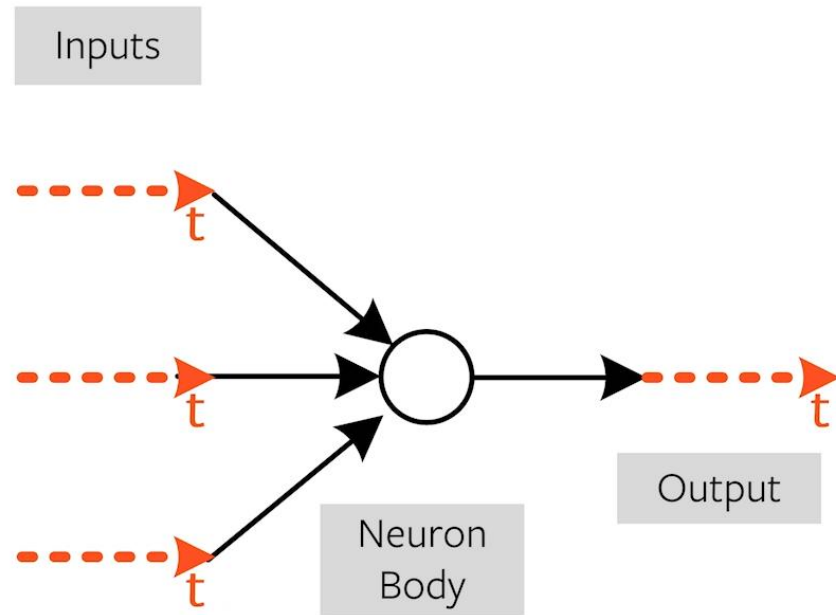
*“7 at position 10; 5 at position 20”*

---

**Static Suppression (aka Event-driven Processing):** The sensory periphery only processes information when there is *new* information to process



# The Leaky Integrate-and-Fire Neuron



Python package for gradient-based  
optimization of SNNs



Gradient-based Learning with Spiking Neural Networks

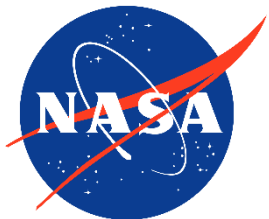


real-time online learning

seamless integration with PyTorch








>200,000 downloads




neuromorphic HW compatible



[github.com/jeshraghian/snntorch](https://github.com/jeshraghian/snntorch)



Tutorial	Title	Colab Link
<a href="#">Tutorial 1</a>	Spike Encoding with snnTorch	 <a href="#">Open in Colab</a>
<a href="#">Tutorial 2</a>	The Leaky Integrate and Fire Neuron	 <a href="#">Open in Colab</a>
<a href="#">Tutorial 3</a>	A Feedforward Spiking Neural Network	 <a href="#">Open in Colab</a>
<a href="#">Tutorial 4</a>	2nd Order Spiking Neuron Models (Optional)	 <a href="#">Open in Colab</a>
<a href="#">Tutorial 5</a>	Training Spiking Neural Networks with snnTorch	 <a href="#">Open in Colab</a>
<a href="#">Tutorial 6</a>	Surrogate Gradient Descent in a Convolutional SNN	 <a href="#">Open in Colab</a>
<a href="#">Tutorial 7</a>	Neuromorphic Datasets with Tonic + snnTorch	 <a href="#">Open in Colab</a>

Advanced Tutorials	Colab Link
<a href="#">Population Coding</a>	 <a href="#">Open in Colab</a>
<a href="#">Regression: Part I - Membrane Potential Learning with LIF Neurons</a>	 <a href="#">Open in Colab</a>
<a href="#">Regression: Part II - Regression-based Classification with Recurrent LIF Neurons</a>	 <a href="#">Open in Colab</a>
<a href="#">Accelerating snnTorch on IPUs</a>	—

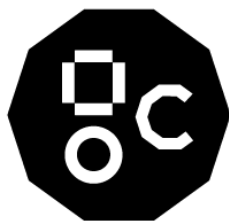
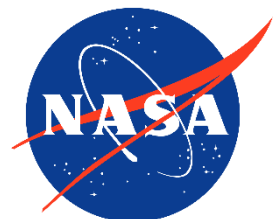
Python package for gradient-based optimization of SNNs

real-time online learning

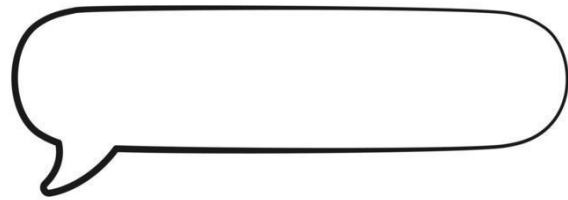
seamless integration with PyTorch

CUDA + IPU accelerated

neuromorphic HW compatible



[github.com/jeshraghian/snntorch](https://github.com/jeshraghian/snntorch)



+

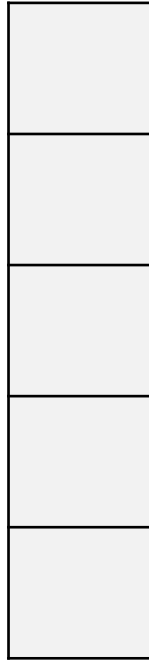


Language Modelling



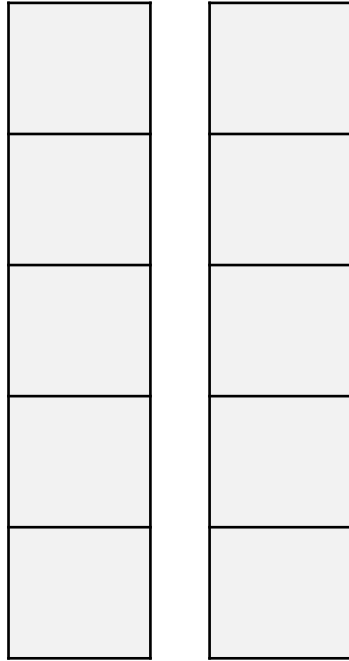
Shrek saw Taylor with binoculars





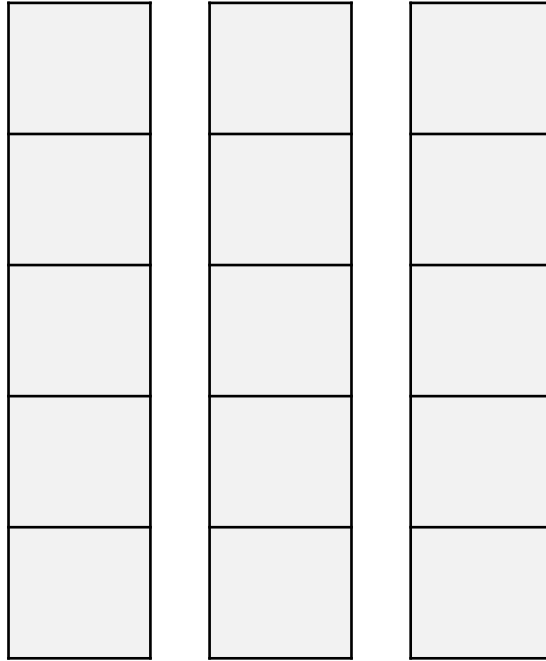
Shrek saw Taylor with binoculars





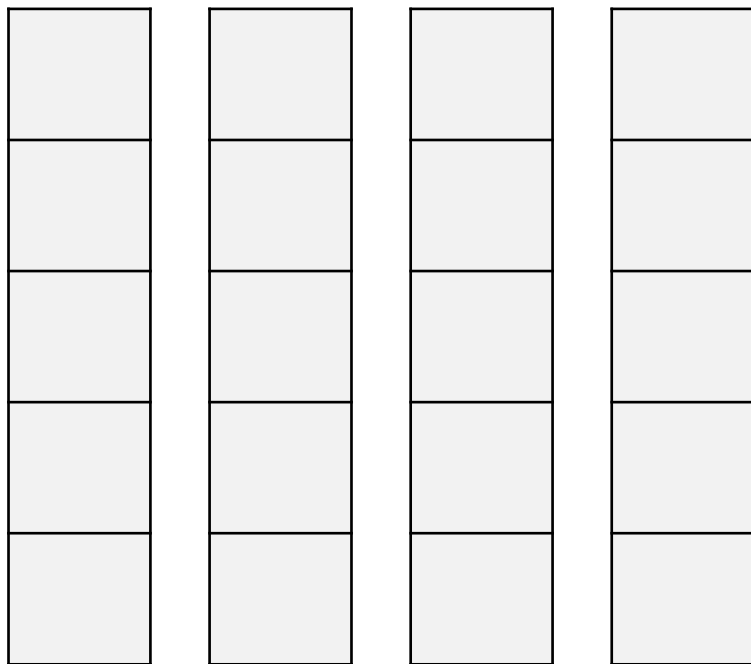
Shrek saw Taylor with binoculars





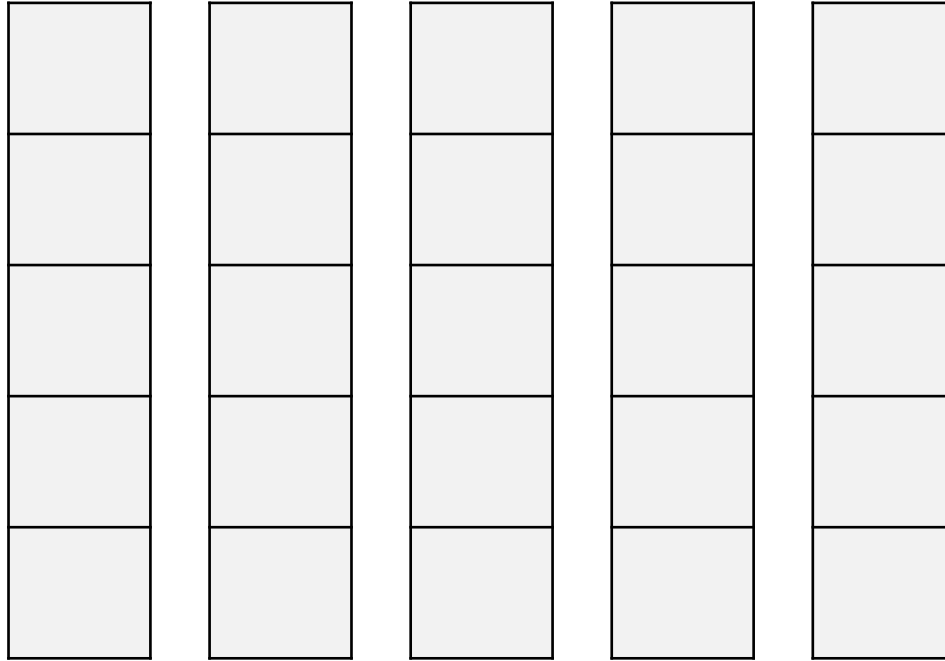
Shrek saw Taylor with binoculars





Shrek saw Taylor with binoculars

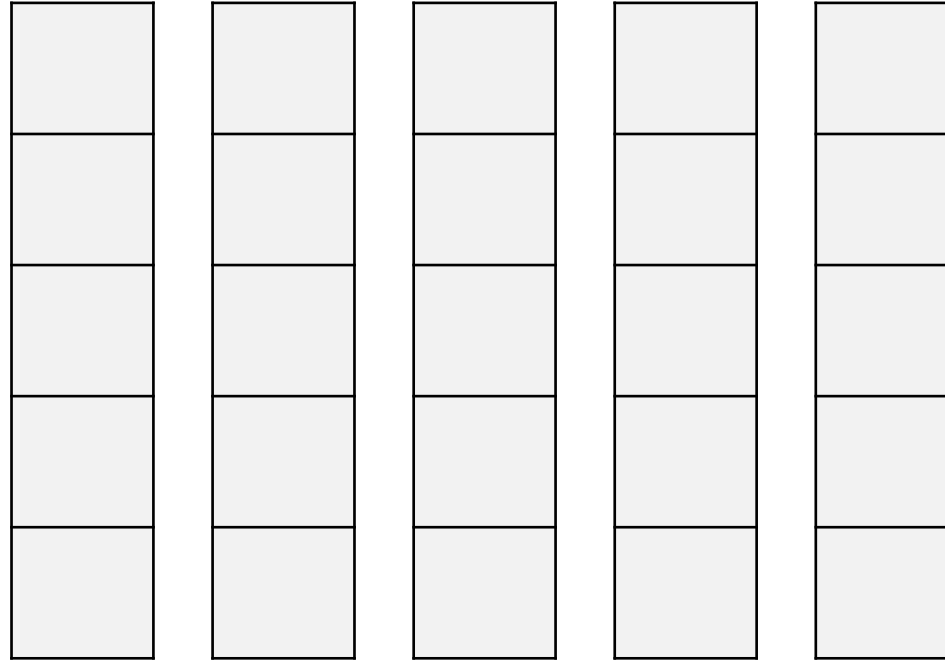




Shrek saw Taylor with binoculars



Shrek  
saw  
Taylor  
with  
binoculars

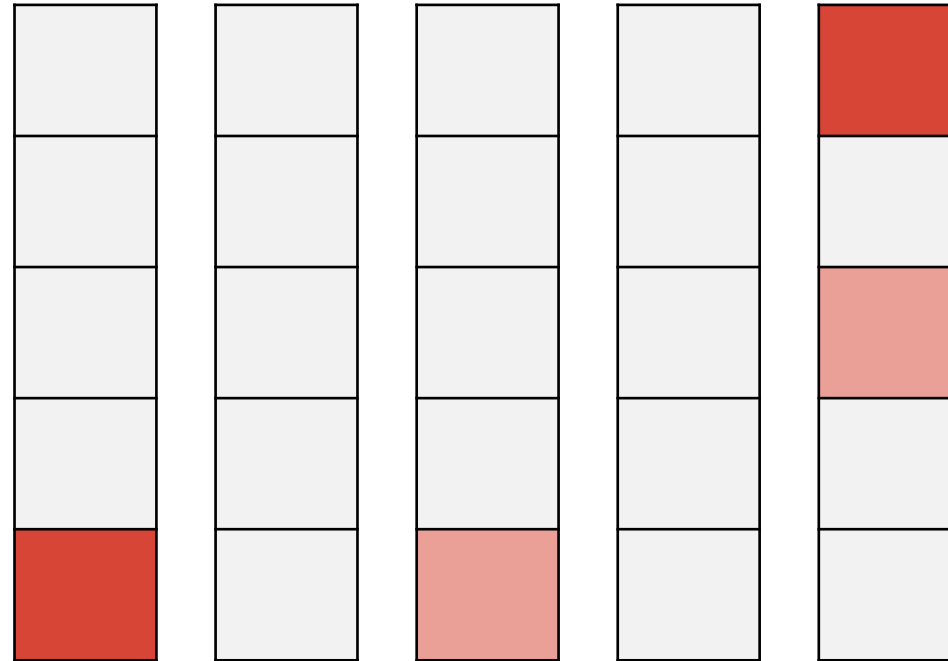


Shrek saw Taylor with binoculars



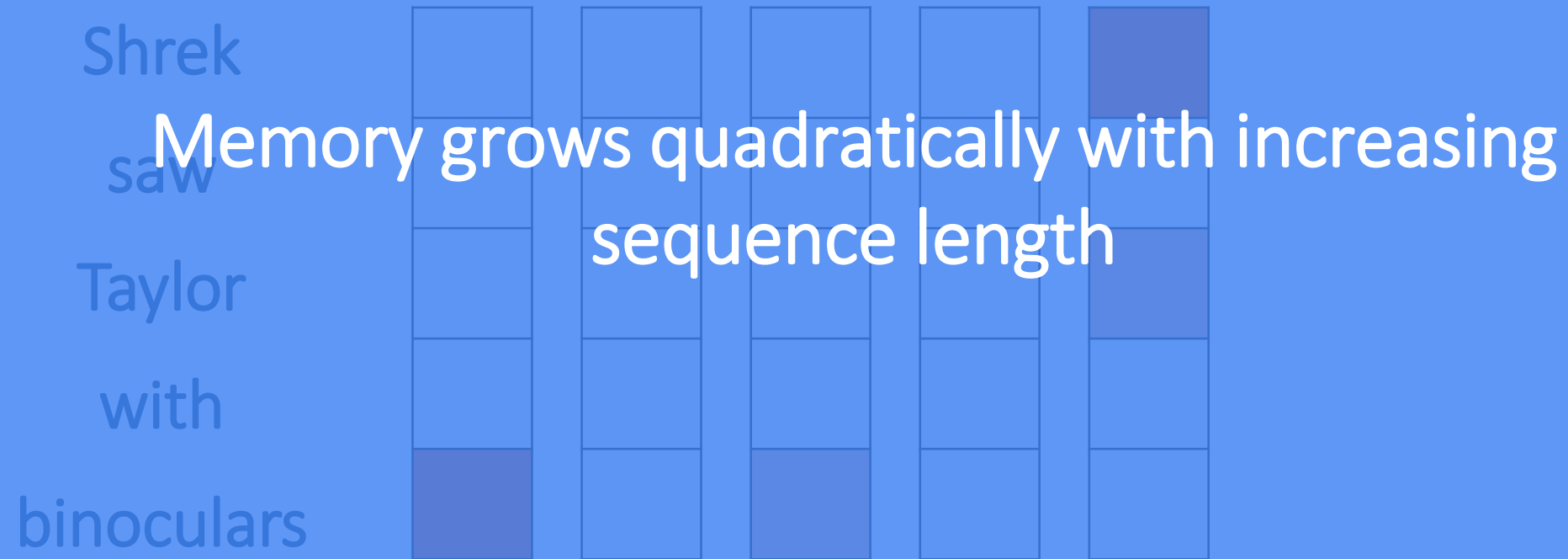


Shrek  
saw  
Taylor  
with  
binoculars

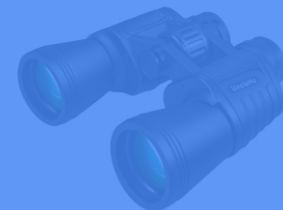
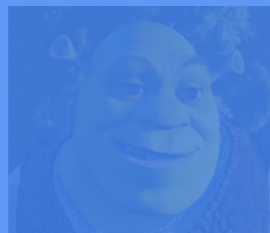


Shrek saw Taylor with binoculars





Shrek saw Taylor with binoculars



Shrek

saw

Taylor

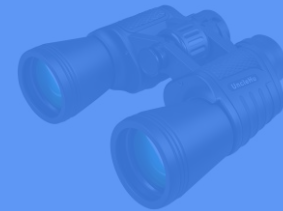
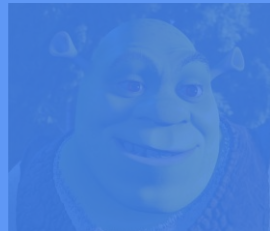
with

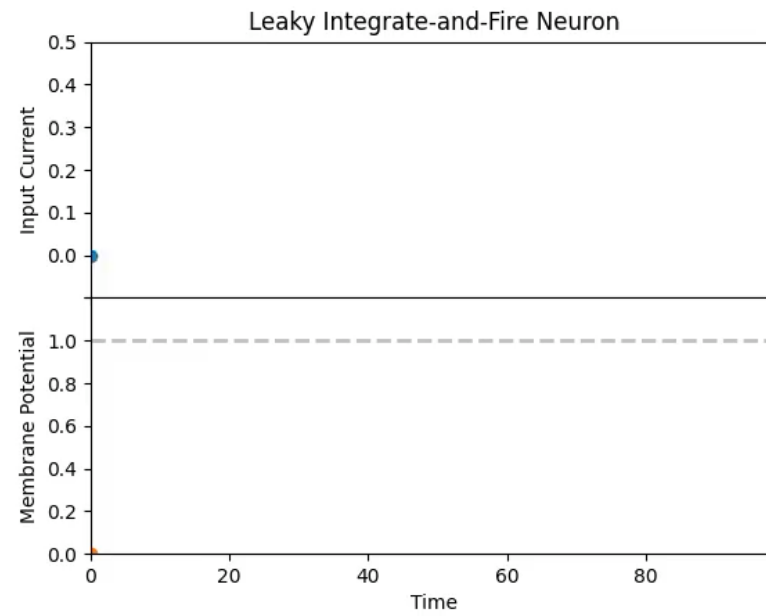
binoculars

Memory grows quadratically with increasing  
sequence length

Your brains are not undergoing  
neurogenesis with every word I say

Shrek saw Taylor with binoculars



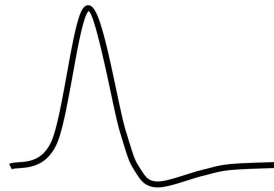


Shrek saw Taylor with binoculars

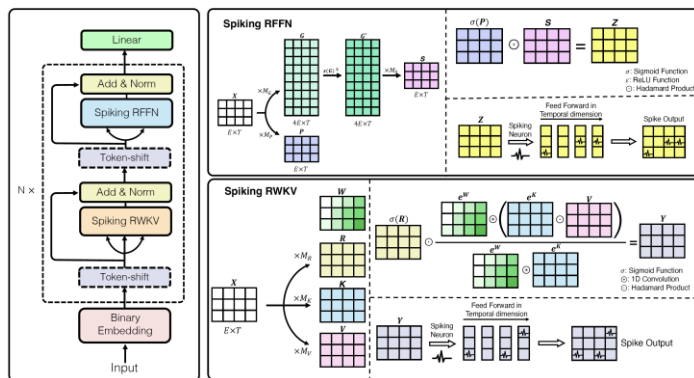


Attention: too much information

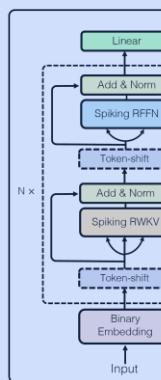
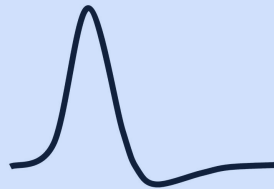
SNNs: not enough information



# SpikeGPT



The largest SNN trained via backprop & the first to perform language generation



The largest S  
first to perform language generation

---

Context → <start>

---

---- Generated Answer 1 ----  
"so what do you think of me? "  
I asked.  
he was standing behind me, still looking at the posters,  
and I suddenly knew what he was asking.  
" what do you think of me? "  
I looked up at him , the corners of his lips pressed together.  
"i don't know."  
he looked at me and i was staring at him, not even realizing what he was doing.  
all I could do was watch him,  
look at him, listen to his voice,  
and then I would fall back to the stage and look at him.

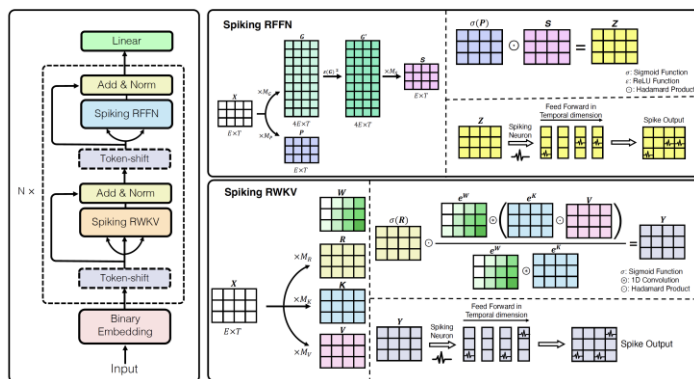
---

Figure 3: Example of text generated by SpikeGPT 120M. The model is trained on BookCorpus.



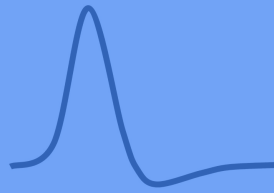


## SpikeGPT



The largest SNN trained via backprop & the first to perform language generation

30x less operations than a transformer of equal size (N=12 blocks)



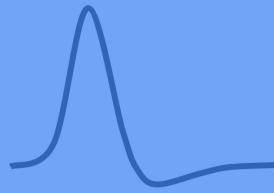
## SpikeGPT

Why isn't the world using it then?



The largest SNN trained via backprop & the first to perform language generation

30x less operations than a transformer of equal size (N=12 blocks)



SpikeGPT

Why isn't the world using it then?

Scaling hurts.

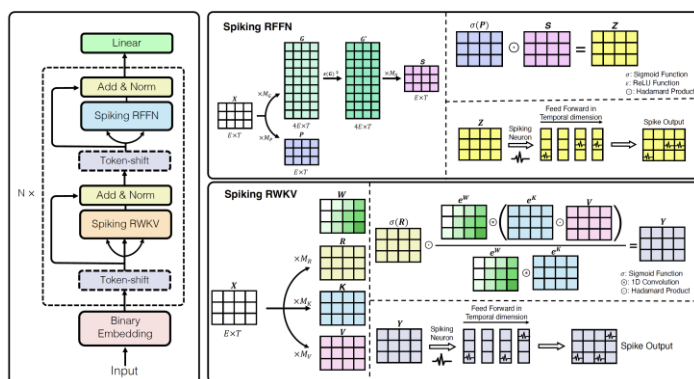


The largest SNN trained via backprop & the first to perform language generation

30x less operations than a transformer of equal size (N=12 blocks)

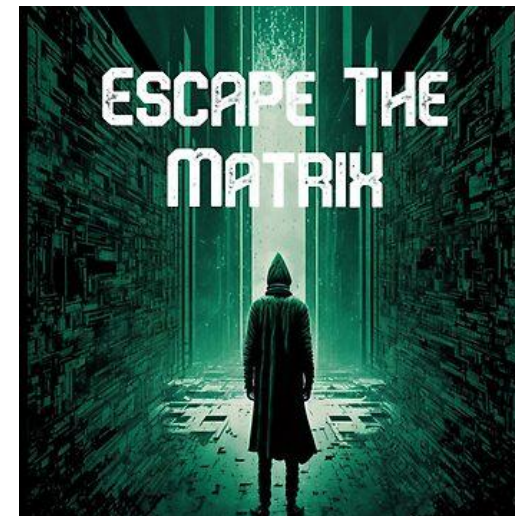


## SpikeGPT



The largest SNN trained via backprop & the first to perform language generation

30x less operations than a transformer of equal size (N=12 blocks)



## MatMul-free LM

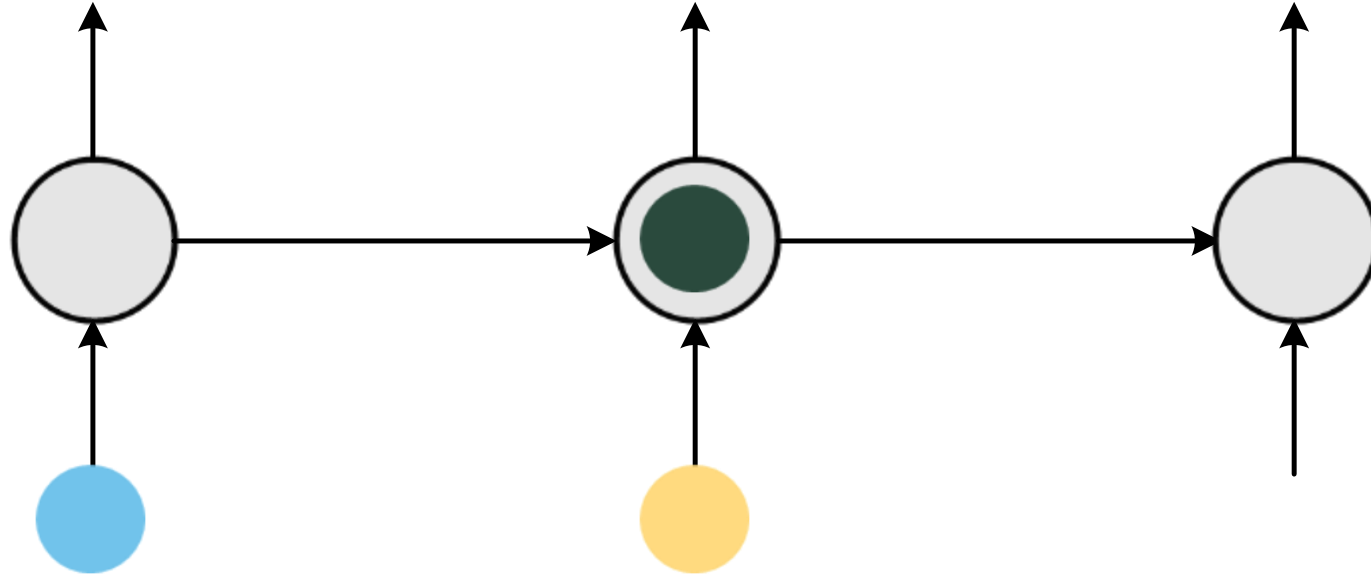
The first billion-parameter scale MatMul-free model to achieve language generation

Ternary parameters: 8-10x less memory

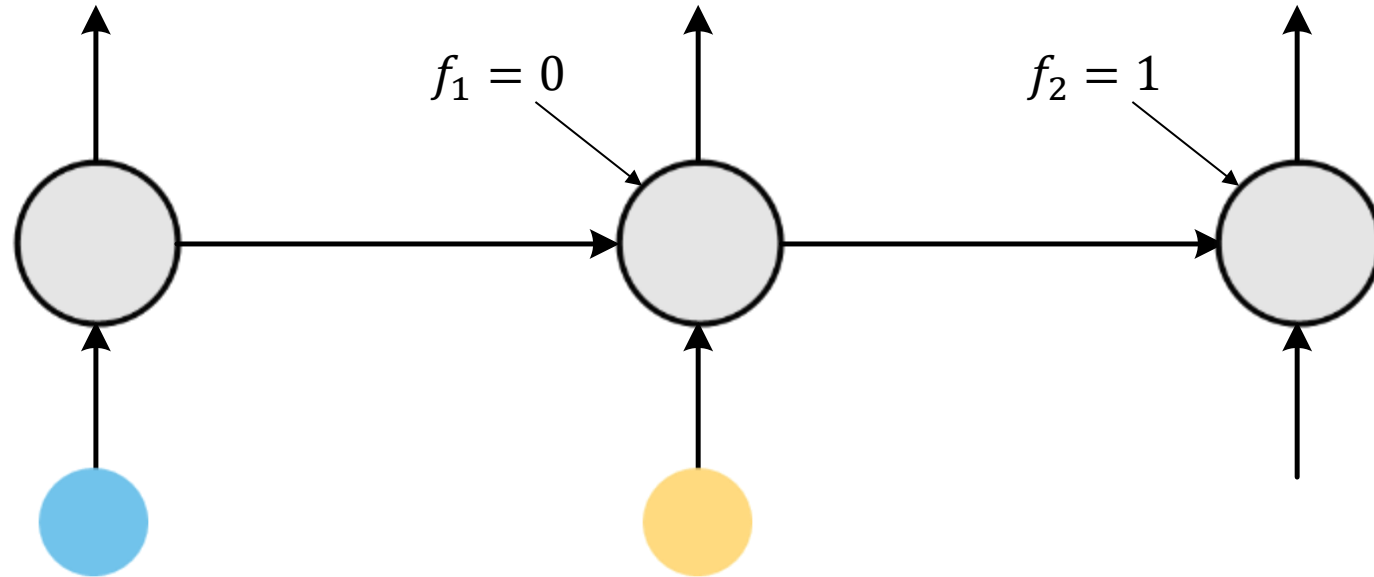
“Learning to forget”

13W at human readable throughput

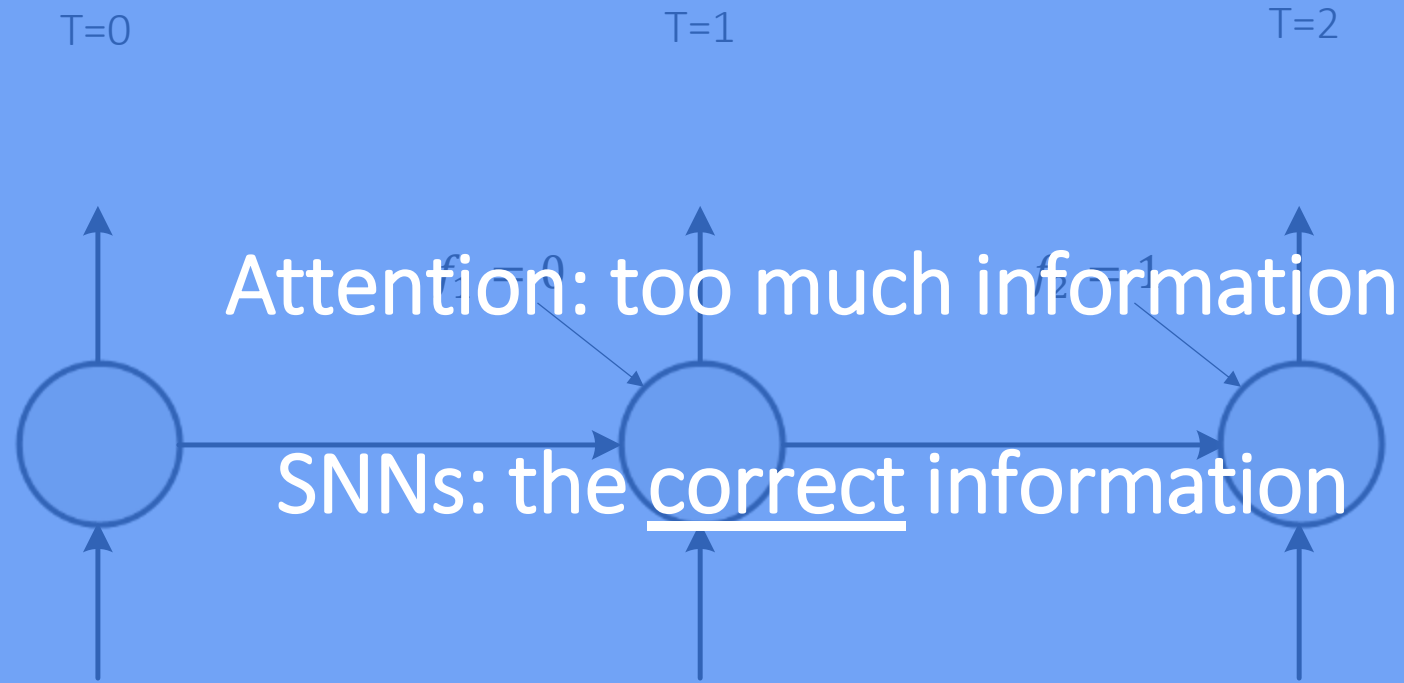
# Forget Gates in Linear RNNs



# Forget Gates in Linear RNNs



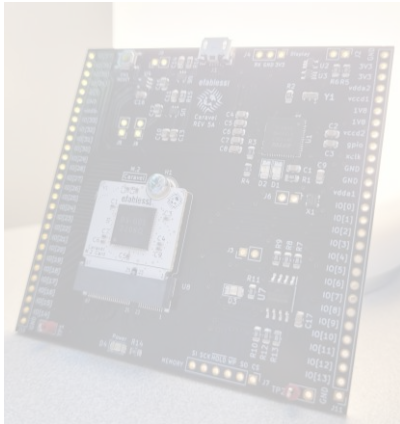
# Learning to Forget





# MatMul-free Language Modelling on Hardware

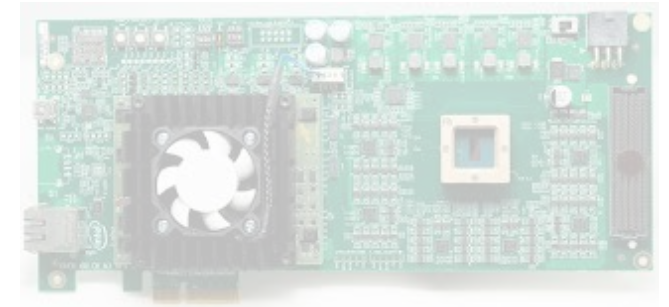
The first billion-parameter scale MatMul-free model to achieve language generation



Custom silicon chips  
(130nm / 22nm)



D5005 Stratix 10  
(FPGA)

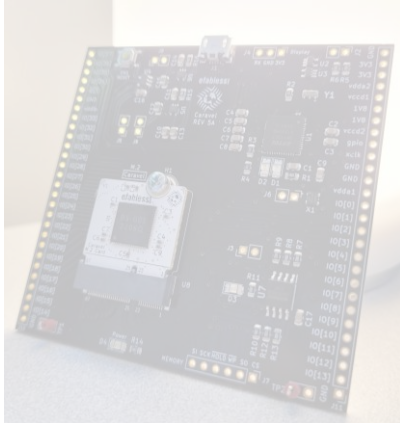


Neuromorphic Processor  
(Intel Loihi: 7nm)

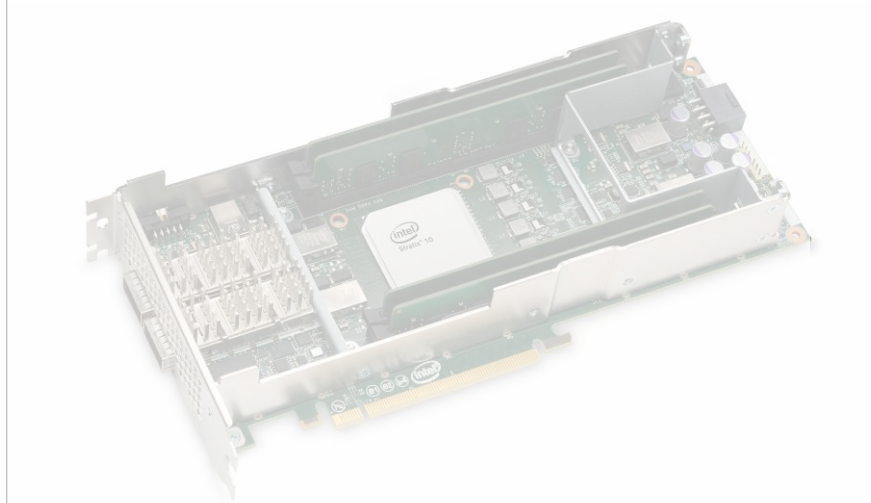
23.8 tokens p/sec @ 13 W  
Human readable throughput:  
~5-15 tokens p/sec @ 20 W

# MatMul-free Language Modelling on Hardware

The first billion-parameter scale MatMul-free model to achieve language generation



Custom silicon chips  
(130nm / 22nm)



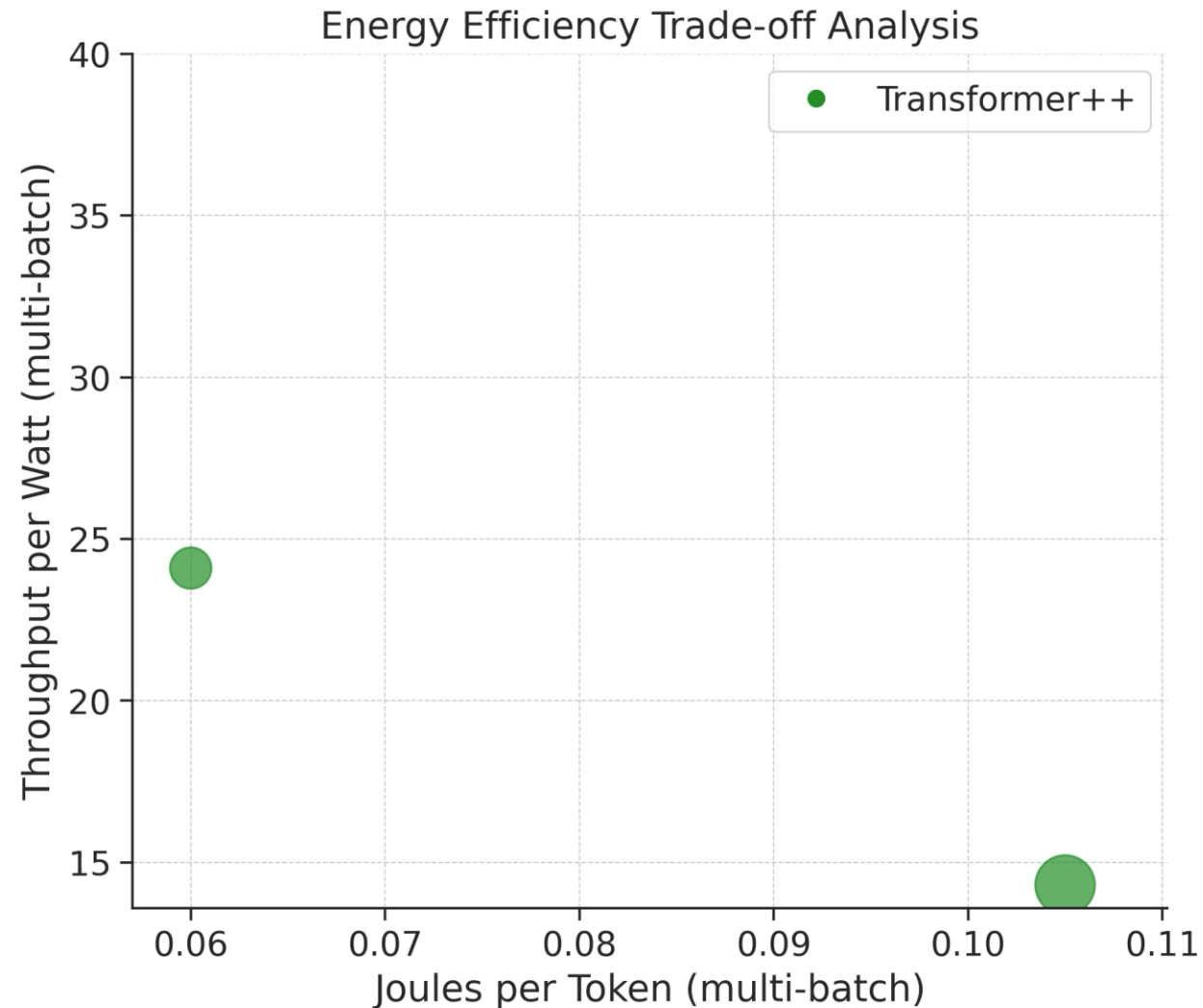
D5005 Stratix 10  
(FPGA)



Neuromorphic Processor  
(Intel Loihi: 7nm)

# MatMul-free Language Modelling on Hardware

The first billion-parameter scale MatMul-free model to achieve language generation

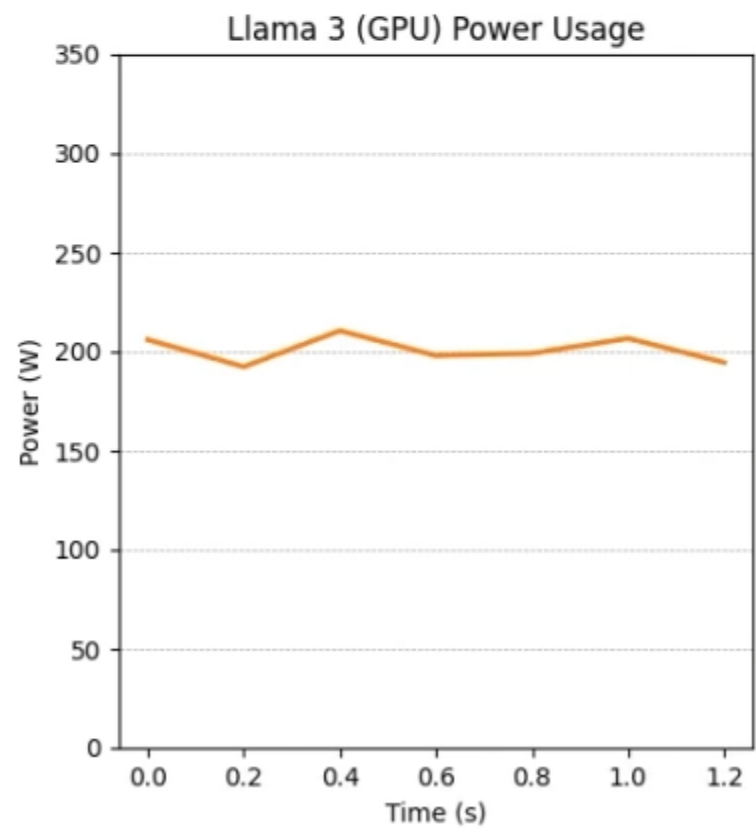


owen@DESKTOP-P9OQ0A1 MINGW64 /d/Dropbox/repos/conscium/llm-profiler

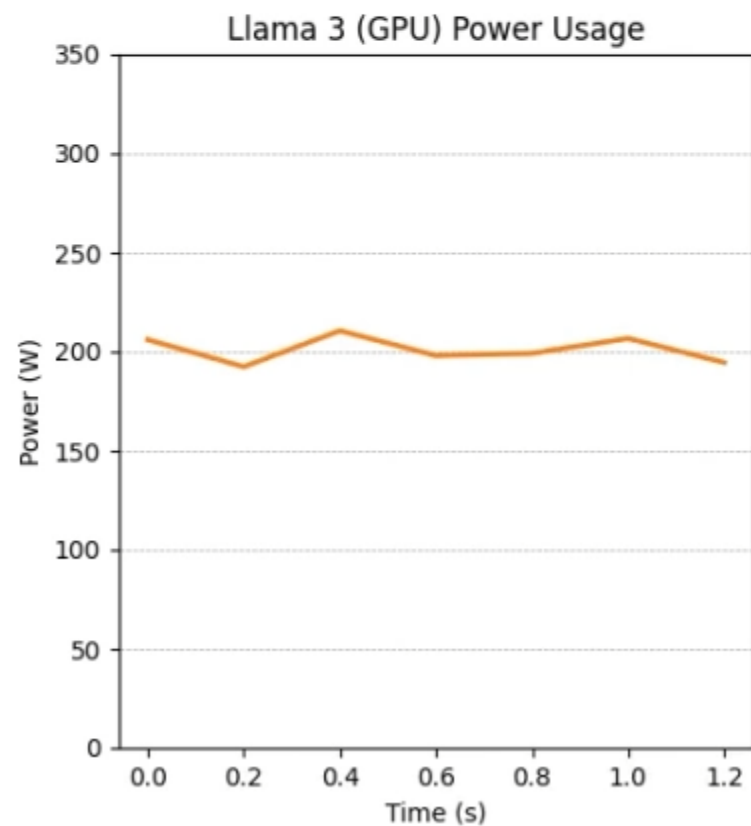
\$ █

```
ower@DESKTOP-P90Q0A1 MINGW64 /d/Dropbox/repos/conscium/llm-profiler
```

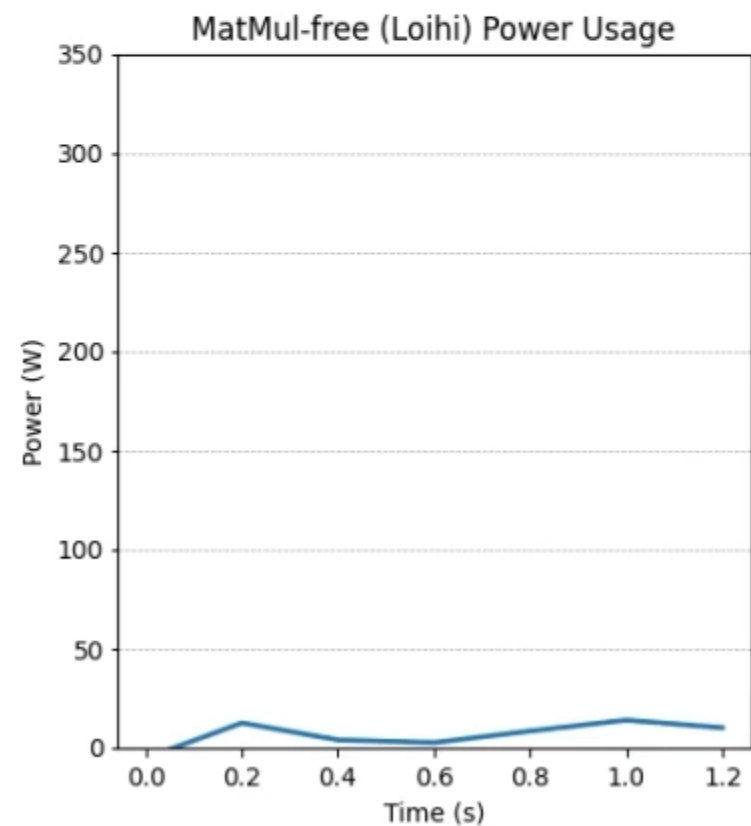
```
$ █
```



```
ower@DESKTOP-P90QOA1 MINGW64 /d/Dropbox/repos/conscium/llm-profiler
$ █
```



```
ower@DESKTOP-P90QOA1 MINGW64 /d/Dropbox/repos/conscium/llm-profiler
$ p █
```

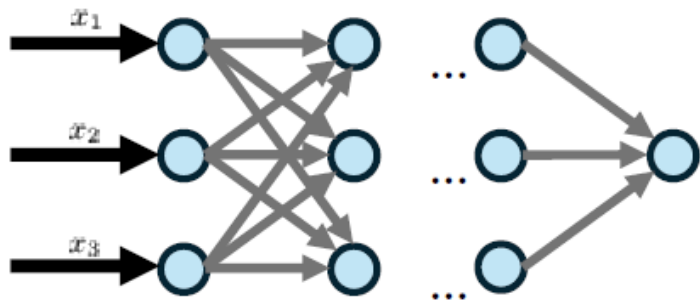


This barely scratches the surface.



## Low-Power Training Methods

### Neural Network



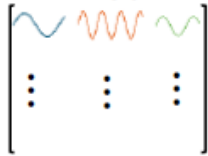
### Gradient “Cell Tower”



### Gradients

$$\begin{bmatrix} \frac{\partial L}{\partial w_{1,1}} & \frac{\partial L}{\partial w_{1,2}} & \frac{\partial L}{\partial w_{1,3}} \\ \frac{\partial L}{\partial w_{2,1}} & \frac{\partial L}{\partial w_{2,2}} & \frac{\partial L}{\partial w_{2,3}} \\ \frac{\partial L}{\partial w_{3,1}} & \frac{\partial L}{\partial w_{3,2}} & \frac{\partial L}{\partial w_{3,3}} \end{bmatrix}$$

### Encode



Filtering

Fine-Tuning

## Multi-Agent System

Architecture

Dataflow

Resource

Sparsity

Memory

etc.

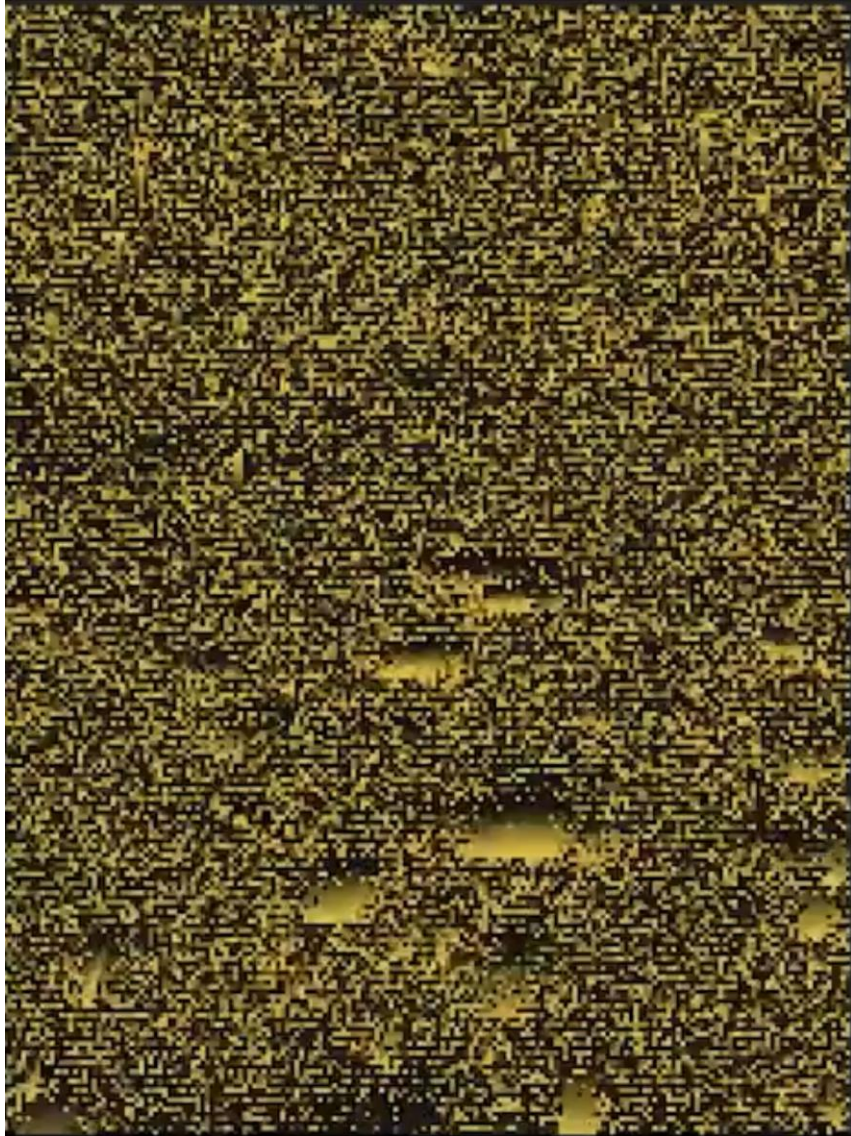
# Event-based Cameras in Space

00:00:00.374231

## RAM Camera Recording

- 27<sup>th</sup> February 2023
- Captured at 18:33:49 UTC
- 30 second duration

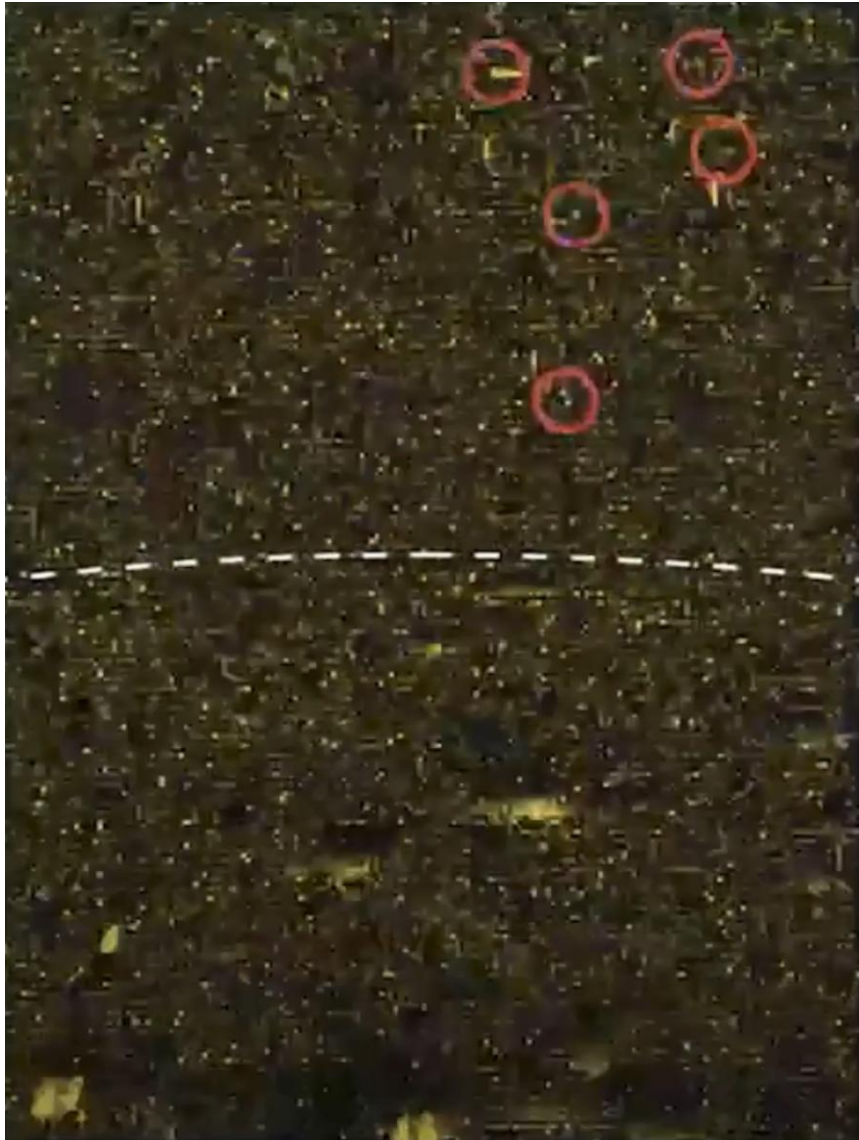
# Event-based Cameras in Space



## RAM Camera Recording

- High-speed playback can be critical to spot interesting objects
- Easy to understand the geometry of the scene
- Information that is inherently lost when frames are made from the data

# Event-based Cameras in Space



RSOs

Horizon



IRIDIUM 103  
4787 km

STARLINK-3093  
4310 km

STARLINK-4650  
4216 km

STARLINK-3079  
4493 km

STARLINK-2525  
4367 km

STARLINK-3097  
4817 km

Harbin

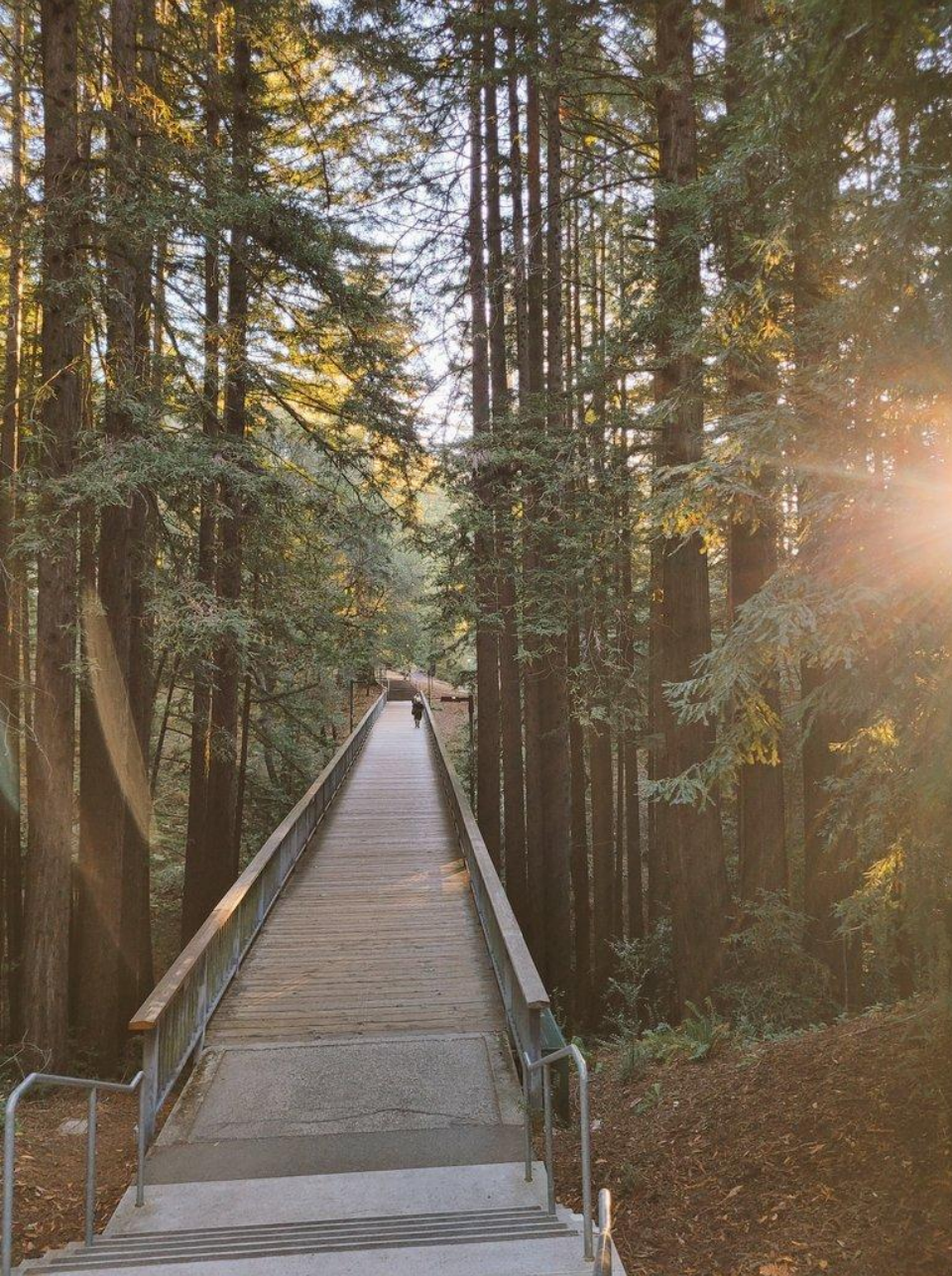
Songyuan

Tongliao

# UCSC Neuromorphic Computing Group







# The Brain Computes Using Time, and So Should Neural Networks

Jason K. Eshraghian  
Assistant Professor, ECE, UC Santa Cruz

3 April 2025

# References

## Our Work

**MatMul-Free:** R. Zhu, J. K. Eshraghian, et al., “Scalable MatMul-Free Language Modeling”, arXiv, 2024.

**SNNs 101:** J. K. Eshraghian et al., “Training Spiking Neural Networks Using Lessons from Deep Learning”, Proc. of the IEEE, 2023.

**SpikeGPT:** R. Zhu, Q. Zhao, G. Li, J. K. Eshraghian, “SpikeGPT: Generative Pre-Trained Language Modeling with Spiking Neural Networks”, TMLR, 2024.

**snnTorch:** J. K. Eshraghian, snnTorch, [github.com/jeshraghian/snnTorch](https://github.com/jeshraghian/snnTorch), 2020.

## Related Work

**HGRN2:** Z. Qin, S. Yang, et al., “HGRN2: Gated Linear RNNs with State Expansion”, arXiv, 2024.

**HGRN:** Z. Qin, S. Yang, Y. Zhong, “Hierarchically Gated Recurrent Neural Networks for Sequence Modeling”, NeurIPS, 2023.

**MetaFormer:** W. Yu et al., “Metaformer is actually what you need for vision”, CVPR 2022.

**Mamba:** A. Gu, T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces”, arXiv 2023.

**Legendre Memory Units:** A. Voelker, I. Kajić, C. Eliasmith, “LMUs: Continuous-time representation in recurrent neural networks”, NeurIPS 2019.