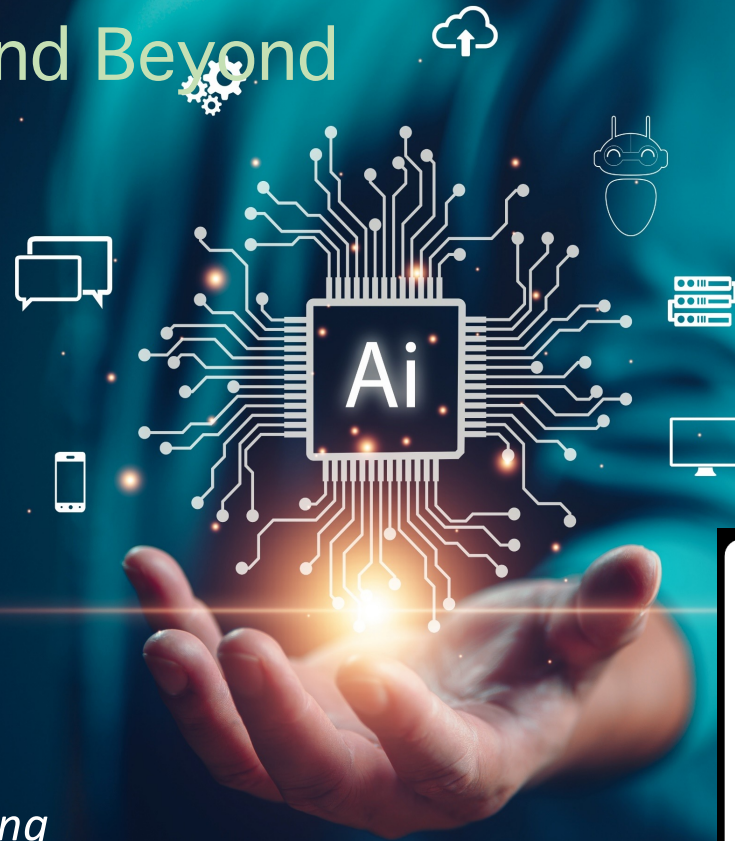


# AI/ML for Embodied Systems at the Edge

## Generative Models, LLMs and Beyond



*Andreas G. Andreou*

<http://andreoulab.net/>

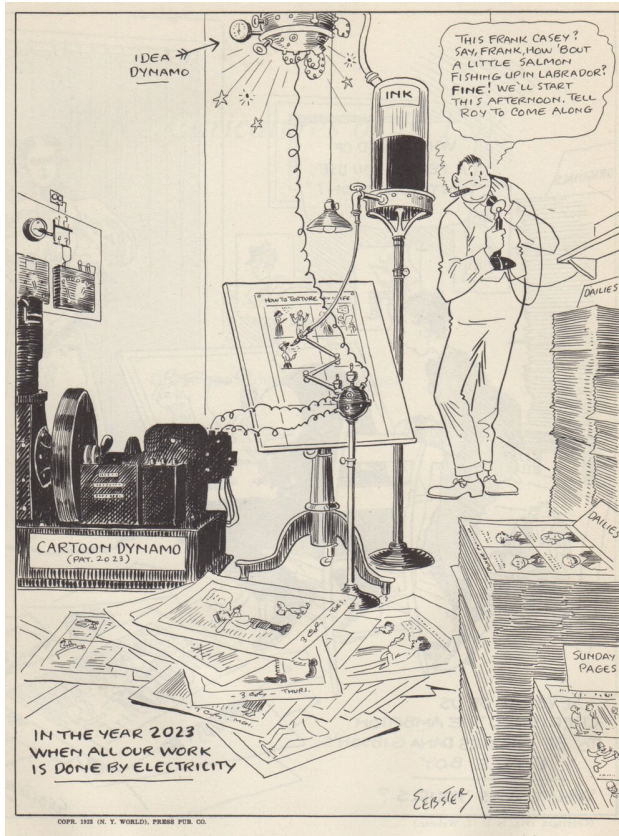
@andreoulab

*Electrical and Computer Engineering,  
Center for Language and Speech Processing  
Johns Hopkins University*

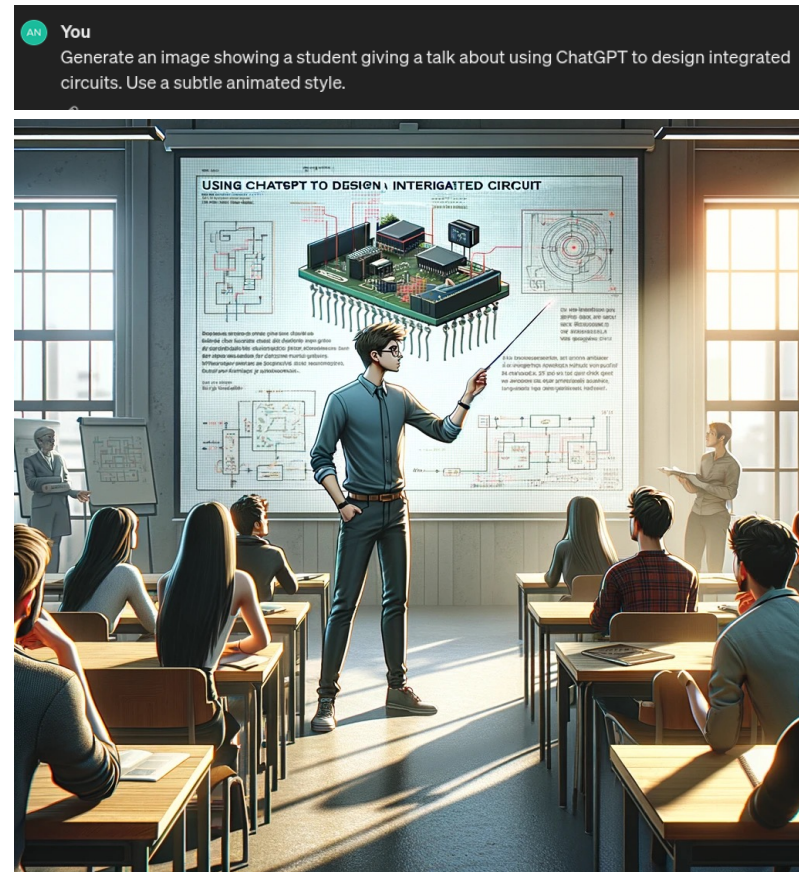


<https://andreoulab.net>

# 2022-2023: The Year of Generative-AI: Large Language Models and Stable Diffusion



A 1923 comic for *New York World* by cartoonist H. T. Webster (1885-1952)

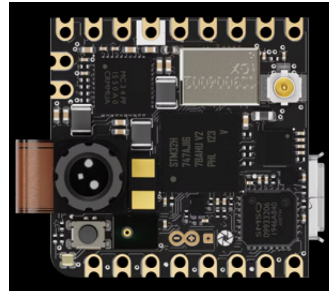


ChatGPT 4, March 2024

# Typical devices at the "Edge"



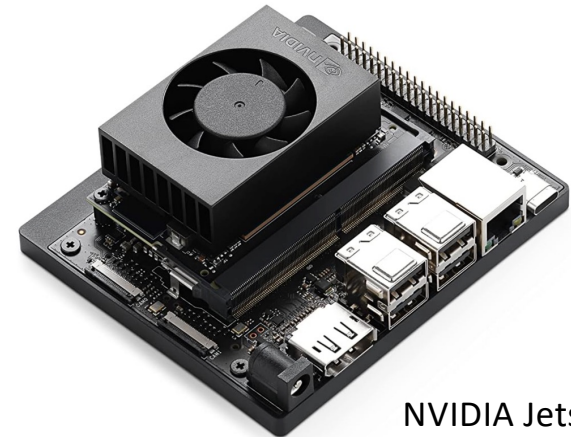
Arduino Nano 33 BLE Sense



Arduino Pro Niclea Vision



Google Edge TPU Dev Micro



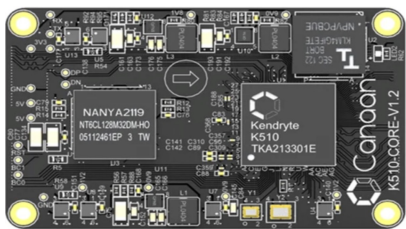
NVIDIA Jetson



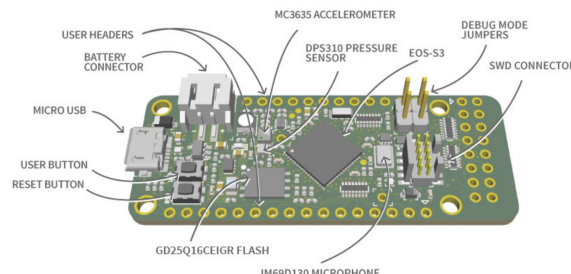
Kendryte K210



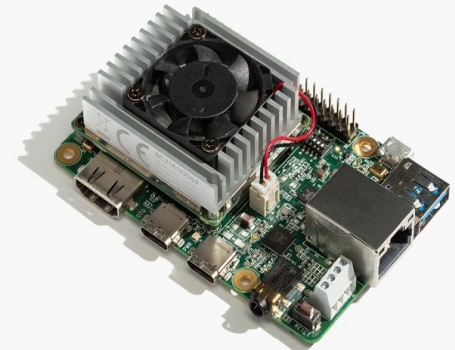
Rasp-Pico



Kendryte K510



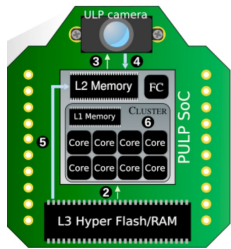
QuickFeather



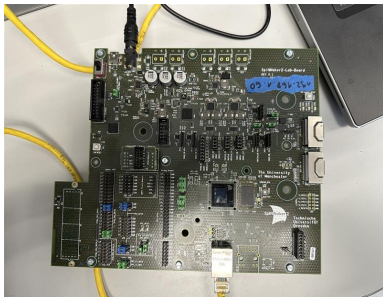
Google Edge TPU Dev

# but these as well ...

A. Garofalo, M. Rusci, F. Conti, D. Rossi, and L. Benini, "PULP-NN: accelerating quantized neural networks on parallel ultra-low-power RISC-V processors," *Phil Trans A* vol. 378, no. 2164, p. 20190155, Feb. 2020

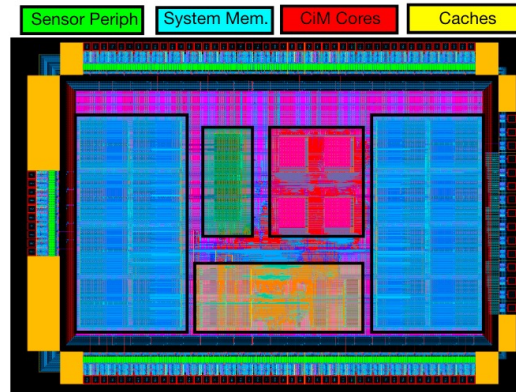


SpiNNaker2 172 M4 FP cores  
DNN accelerators



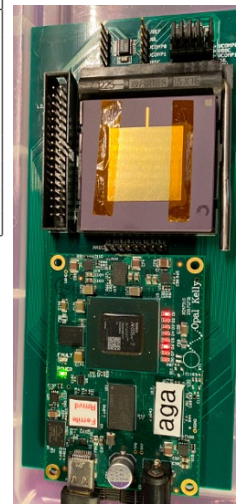
C. Mayr, S. Hoepfner, and S. Furber, "SpiNNaker 2: <http://arxiv.org/abs/1911.02385>

March 2024

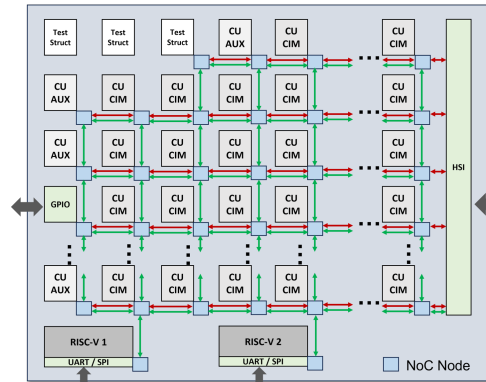


Metric	Value
Clock Speed	100 MHz
Die Dimensions	7mm x 4mm
Max Event Readout	17Meps
CiM Cell Density	1.55M/mm <sup>2</sup>
1-bit MAC Throughput	2.1 * 10 <sup>11</sup> Op/s
1-bit efficiency	45fJ/Op
8-bit MAC Throughput	3.3 * 10 <sup>9</sup> Op/s
8-bit efficiency	1.47pJ/Op
Power Consumption	30 mW

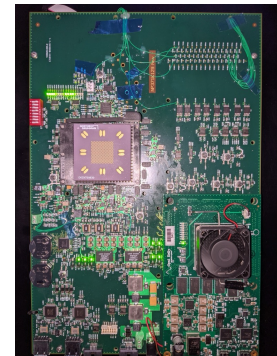
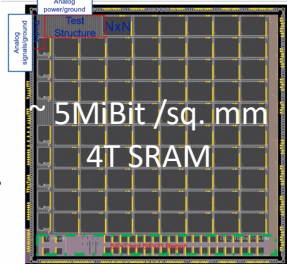
D. R. Mendat, A.G. Andreou *et al.*, "A RISC-V Neuromorphic Micro-Controller Unit (vMCU) with Event-Based Physical Interface and Computational Memory for Low-Latency Machine Perception and Intelligence at the Edge," in *ISCAS-2023*.



TinyML Workshop on Generative AI and Foundation Models at the Edge



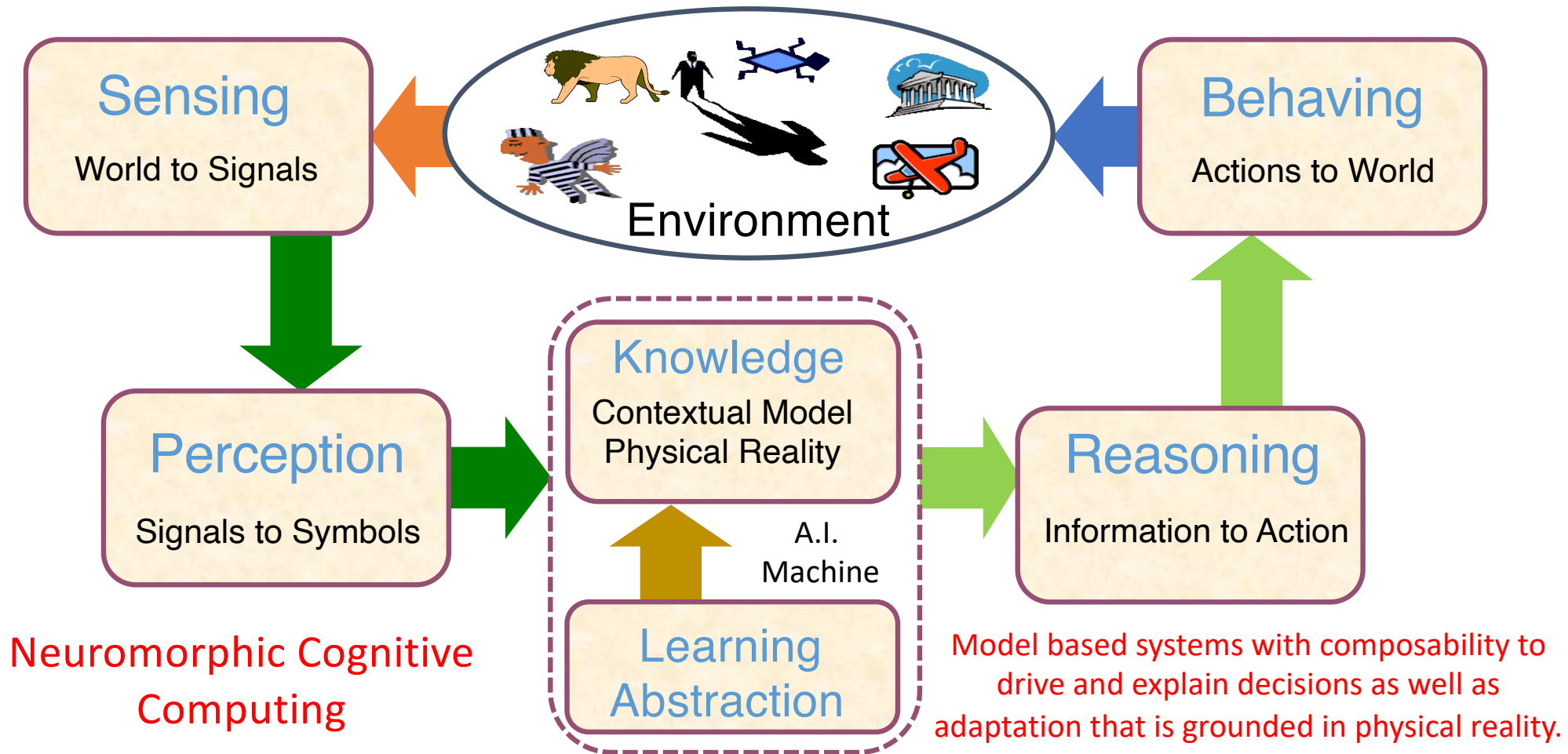
Specification	Value
# CUs with CIM Arrays	77
# CIM Cells per CIM Array	262K
# Total 1-bit CIM Cells on Chip	20.1M
CiM Cell Density	4.66 M/mm <sup>2</sup>
# Auxiliary CUs	9
# RISC-V Processors	SiFive S7 and SiFive S21
I/O Interfaces	HSI (DDR), GPIO, UART (x2), 4 bit-SPI (x2)
HSI Data Rate	25.6Gbps (64-bit DDR @ 200MHz)
GPIO Data Rate	0.64Gbps (32-bit @ 100MHz)
CU Internal Clock Speed	100MHz
1-bit MAC Throughput	6.7TOPS
Energy per 1-bit MAC Op	28.6 fJ/Op (8-bit output)
8-bit MAC Throughput	104.6GOPS
Energy per 8-bit MAC Op	1.83 pJ/Op



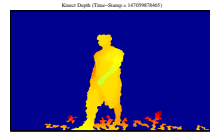
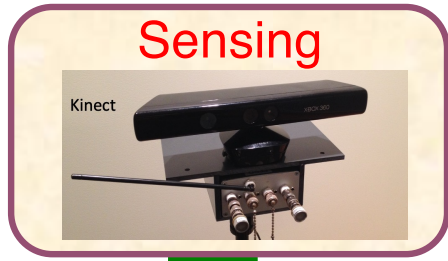
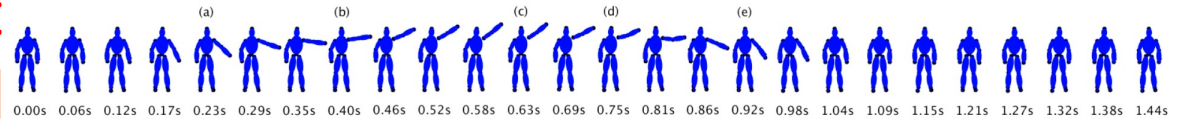
J. L. Molin and A. G. Andreou *et al.*, "Edge Super Compute (ESC) 2.0 A Low Power System-on-Chip AI Accelerator for the Edge," in *Proceedings GOMACTech-2024*, 20 March 2024.

3

# AI/ML in embodied/embedded systems at the EDGE



# Action/gesture/gait recognition at the EDGE without a camera- like a bat



High Dimensional Data (RGBD)

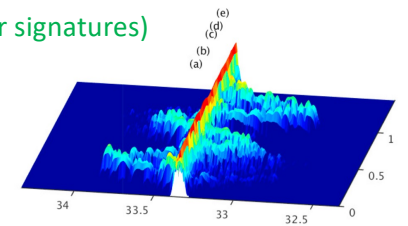
Low Dimensional Data (uDoppler signatures)

Skeletal Pose Sequences

Ultrasound Time-Series

Rotation Representation

Spectrogram Representation



**Learning:** High dimensional data (Kinect cloud) + Low dimensional data (micro-Doppler signatures - time series)

Train

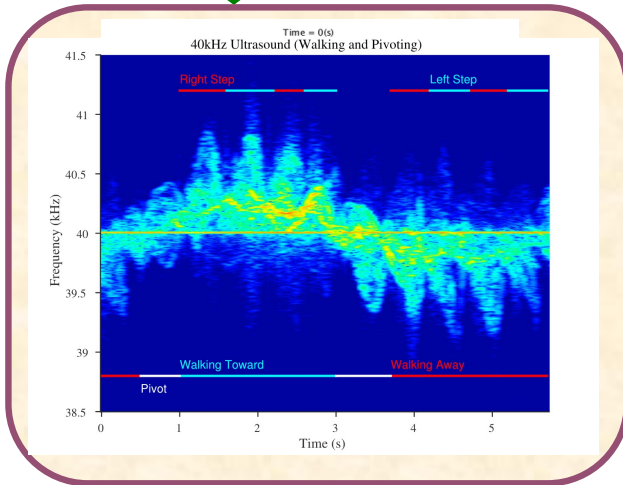
Train/Test

Action Recognizer

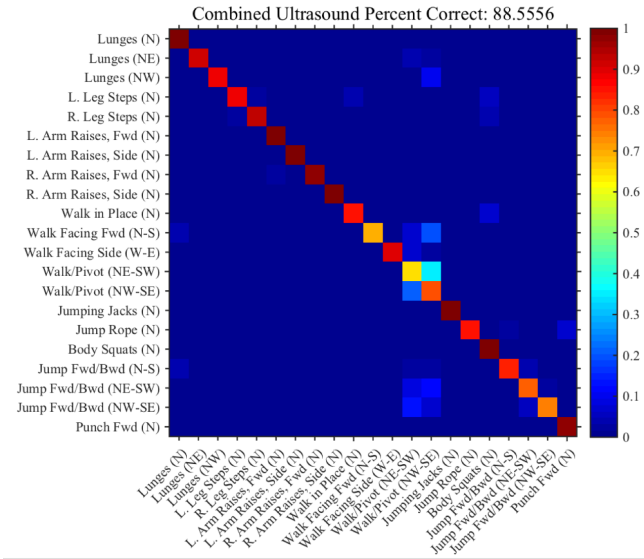
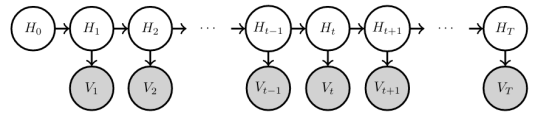
**Inference:** Low dimensional data i.e. u-Doppler signatures - time series

Skeletal Pose Estimate

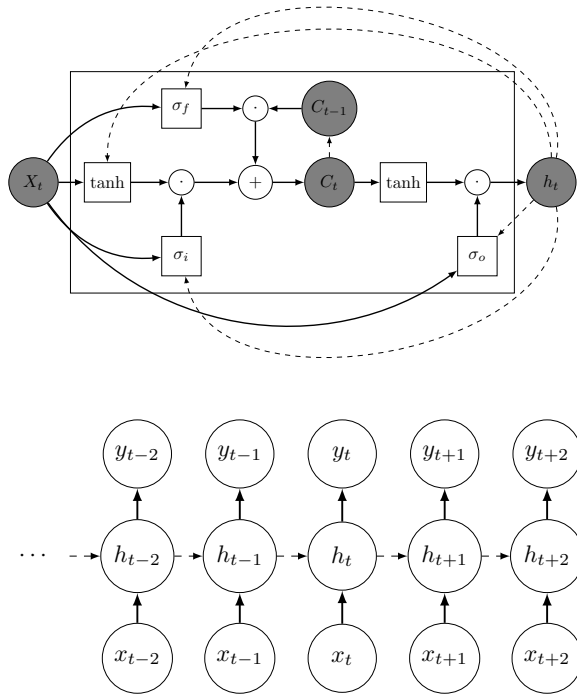
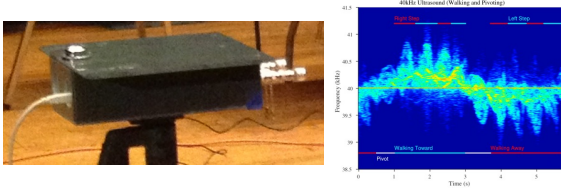
Action Label



**Models**



# Run time: LSTM Model

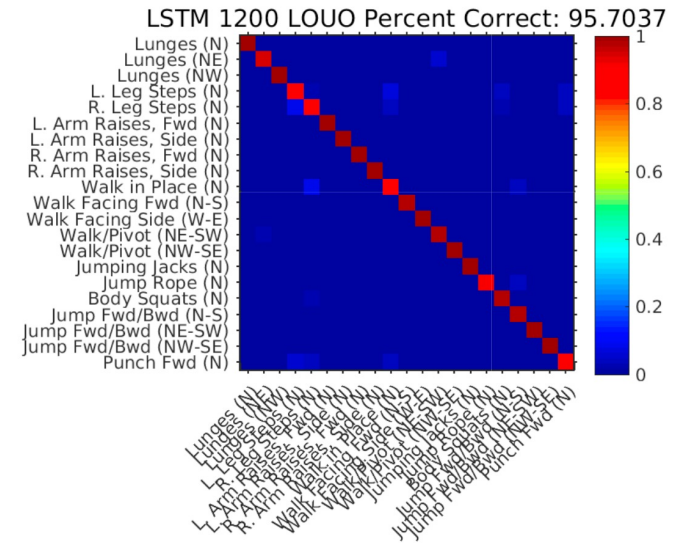
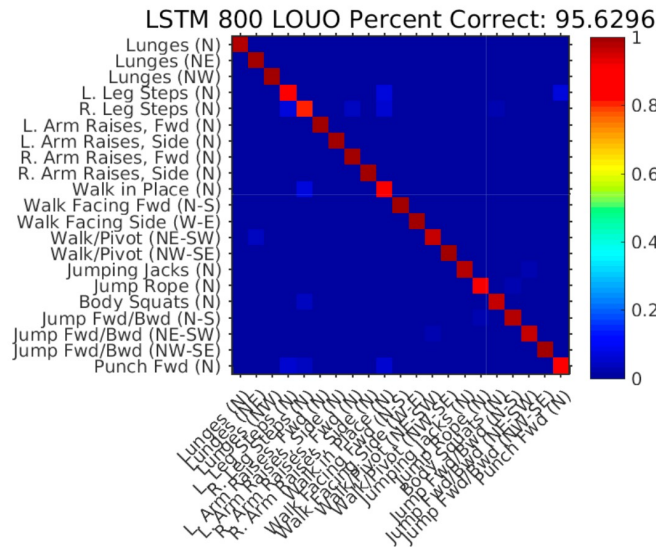


Default Training Parameters	
Number of epochs	200
Seed	1337
Optimizer	Adam
Dropout	0.5
Max Sequence Length	404
Batch Size	100

Model	5 Fold	LOUO
POE Baseline	87.8889	67.963
Stacked HMM	93.6296	89.0
LSTM 800	98.1852	95.6296
LSTM 1200	97.5926	95.7037

POE: Product Of Experts

LOUO: Leave One User Out



## Learning new actions and new end users at the EDGE

### Why?

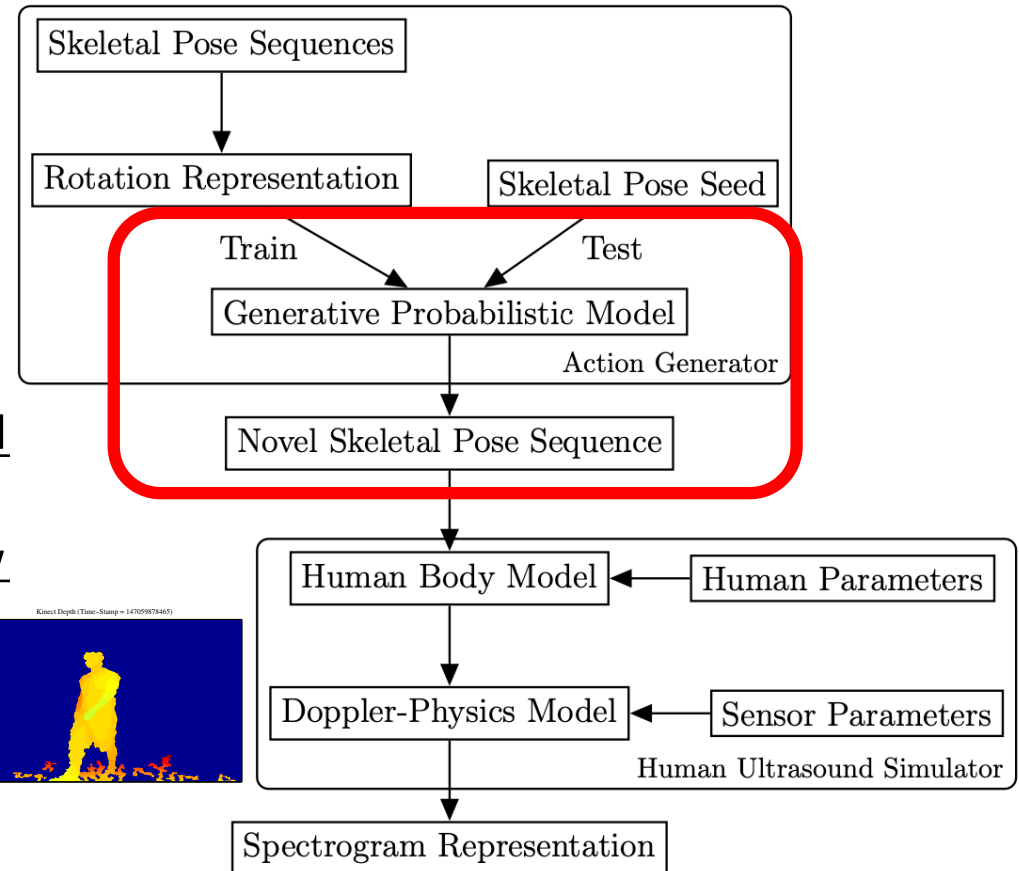
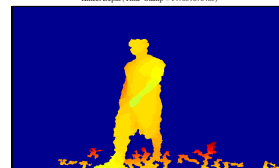
Do learning at the EDGE to:

1. Enroll new end users and actions
2. Alleviate data-center energy expenses. i.e. end user pays for electricity (the problem with Alexa business model; Amazon pays!)

### Technical Challenge:

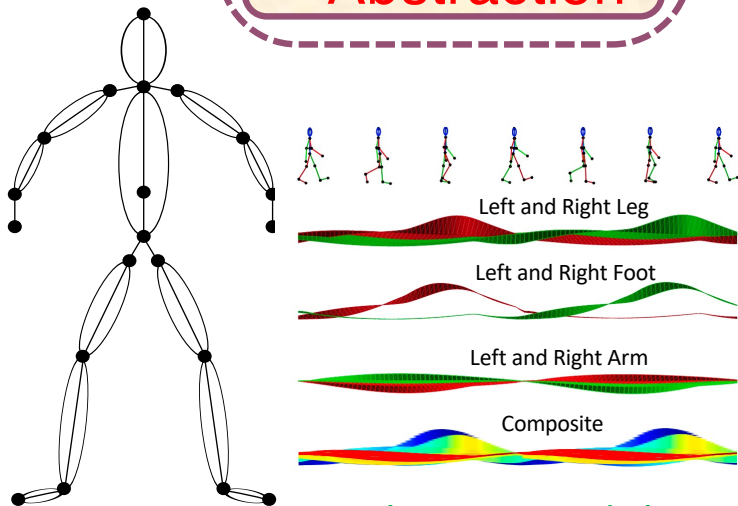
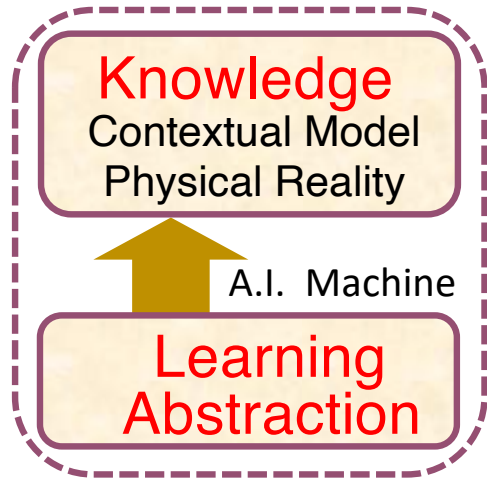
When deployed at the EDGE, high dimensional data (RGBD) are not available to facilitate learning of novel actions/gait/gestures or new users.

1. We need a model base approach that is grounded to physical reality.
2. Need a generative model to create novel poses and introduce stochasticity.

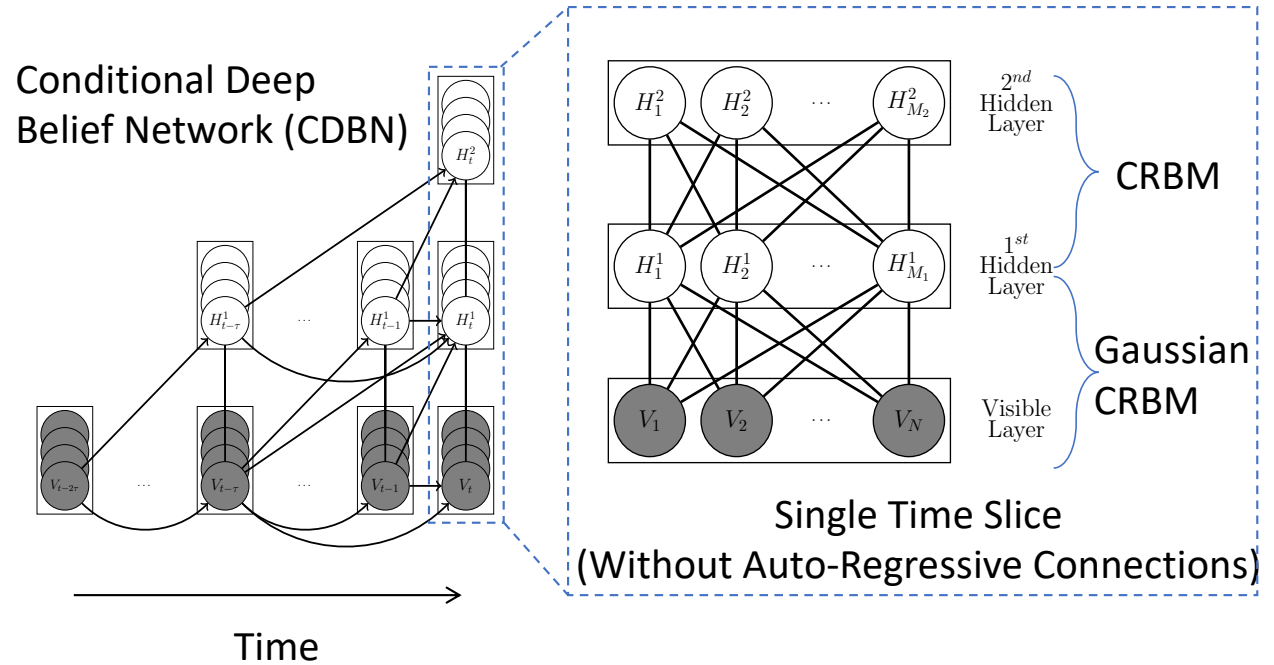




# Contextual Modeling: Physics and Generative AI

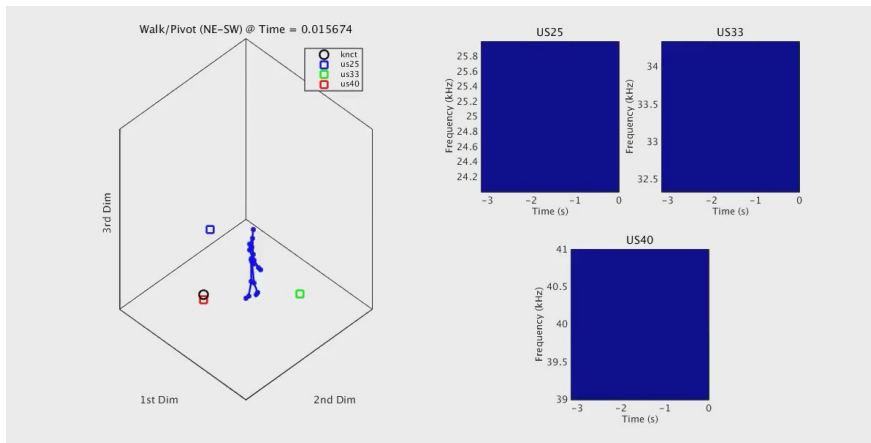
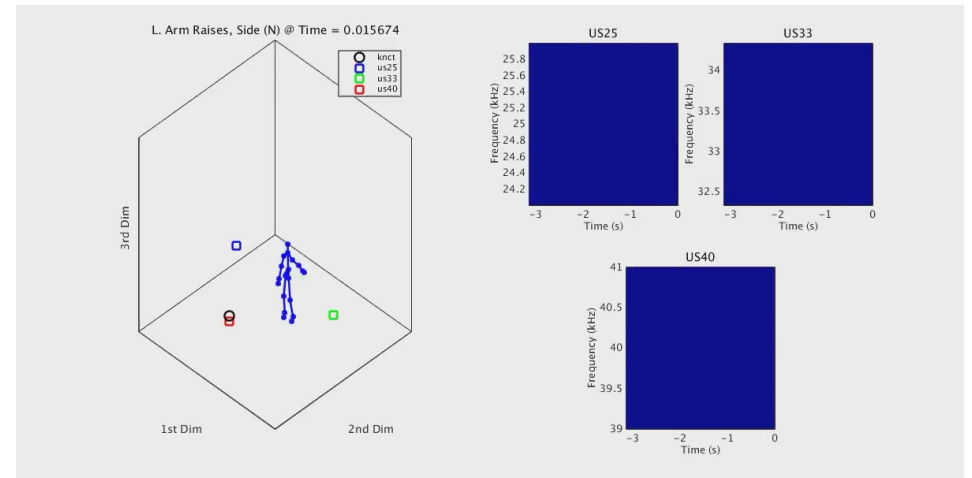
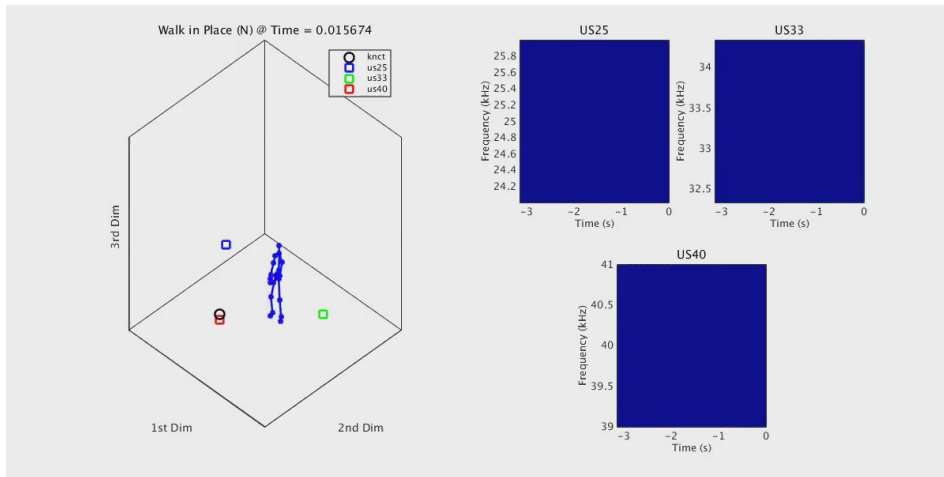


Physics Model



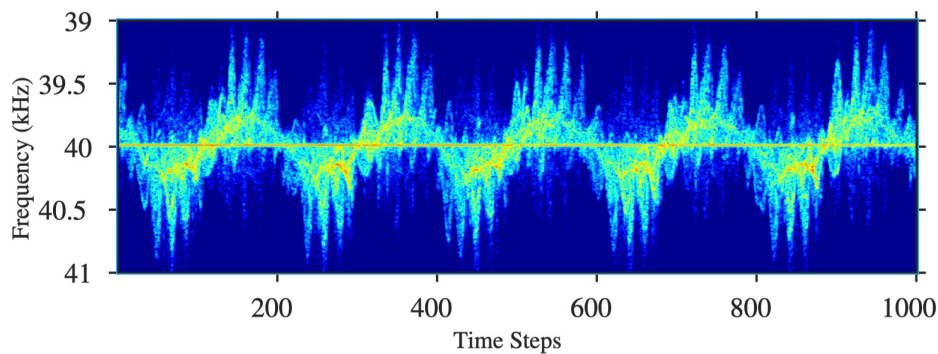
Generative AI model

# Hallucinating skeletal poses and micro-Doppler signatures

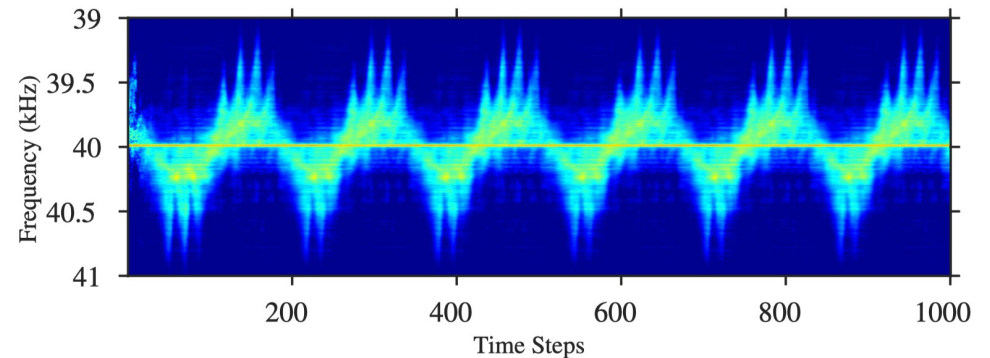


Use these generated micro-Doppler signatures with local zero/few shot learning to adapt the models.

# Hallucinating can be troublesome!



Real micro-Doppler data



Hallucinating micro-Doppler data  
(sometimes it is missing or has extra limbs!)

Z. Zhang, P. O. Pouliquen, A. M. Waxman, and A. G. Andreou, "Acoustic micro-Doppler radar for human gait imaging," *The Journal of the Acoustical Society of America*, vol. 121, no. 3, pp. EL110-3, Mar. 2007.

S. Dura-Bernal, G. Garreau, J. Georgiou, A. G. Andreou, S. L. Denham, and T. Wennekers, "Multimodal integration of micro-Doppler sonar and auditory signals for behavior classification with convolutional networks," *International Journal of Neural Systems*, vol. 23, no. 5, 2013.

K. A. Sanni, G. Garreau, J. L. Molin, and A. G. Andreou, "FPGA implementation of a Deep Belief Network architecture for character recognition using stochastic computation," *49th Annual Conference on Information Sciences and Systems IEEE (CISS-2015)*

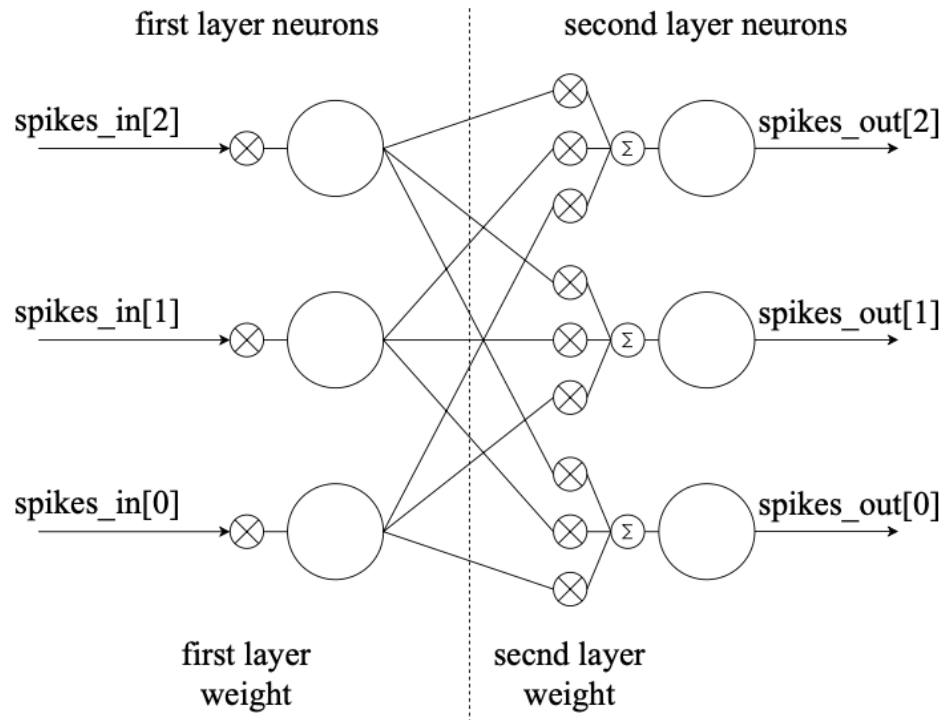
J. Craley, T. S. Murray, D. R. Mendat, and A. G. Andreou, "Action recognition using micro-Doppler signatures and a recurrent neural network," in *Proceedings of the 51st Annual Conference on Information Sciences and Systems IEEE (CISS-2017)*.

T. S. Murray, D. R. Mendat, K. A. Sanni, P. O. Pouliquen, and A. G. Andreou, "Bio-inspired Human Action Recognition With A micro-Doppler Sonar System," *IEEE Access*, vol. PP, no. 99, 2017.

T. S. Murray, Human action recognition from active acoustics: physics modeling for representation learning and inference using generative probabilistic graphical models, Ph.D. Dissertation 2015.

# LLMs for creating new generation of AI chips

# Writing chips using natural language and LLMs: Design a Spiking Neural Network chip



## Topic Search Results from Github

Search Term	Language	Number of Results
"integrate and fire"	Verilog	3
"integrate and fire"	Any	750
"snn"	Verilog	26
"snn"	Any	2.4k
"spiking neuron"	Verilog	6
"spiking neuron"	Any	478
"spiking"	Verilog	29
"spiking"	Any	8.7k
"alu"	Verilog	1.1k
"accelerator"	Verilog	339
"cnn"	Verilog	198
"spi"	Verilog	530

M. A. Tomlinson, J. Li, and A. G. Andreou, "Designing Silicon Brains using LLMs: Leveraging ChatGPT for Automated Description of a Spiking Neuron Array," in *Proceedings 2024 Argentine Conference on Electronics, IEEE, 2024*, pp. 154–159.

# Neuromorphic chip design using chatGPT

AN Write a verilog module for a leaky integrate and fire neuron

Sure, I can provide a basic example of a Leaky Integrate-and-Fire (LIF) neuron model implemented in Verilog. The LIF neuron is the simplest type of spiking neuron model. It accumulates its inputs until it reaches a threshold, at which point it emits a spike and resets.

```
module LIF_neuron(
  input wire clk,
  input wire rst,
  input wire [7:0] input_spike, //input spike train
  output reg output_spike //output spike train
);
```

This simple LIF neuron module in Verilog accumulates input spikes, 'leaks' at each clock cycle, and produces an output spike when the membrane potential reaches a certain threshold. Note that this is a highly simplified model and may not represent all the complexities of a biological neuron.

Also, remember to test your design in a simulation environment before implementing it in a real system. Use stimuli that represent real-world conditions to ensure your design performs as expected. Furthermore, do a formal verification if possible, to mathematically prove that your design meets its specifications.

```
end
else begin
  output_spike <= 0;
end
end
end
endmodule
```

AN Now that we have a working neuron, can you create a new module instantiates a network of neurons with 2 layers in a fully connected fashion?

Certainly! Let's define a simple 2-layered neural network with neurons fully connected between the layers. For simplicity, let's assume:

1. Layer 1 has  $N$  neurons.
2. Layer 2 has  $M$  neurons.

```
module two_layer_neural_network (
  input clk, // clock input
  input reset, // asynchronous reset
  input [7:0] input_current[N], // Current input for each neuron in layer 1
  output spike_out[M] // Spike outputs for each neuron in layer 2
);

// Parameters
parameter N = 4; // Number of neurons in Layer 1
parameter M = 3; // Number of neurons in Layer 2
parameter SPIKE_CONTRIBUTION = 8'd10; // Current contribution to Layer 2 for every Layer 1 spike

// Signals
wire spike_layer1[N];
reg [7:0] layer2_current[M];

// Instantiate neurons in Layer 1
generate
```

AN

I dont think verilog allows for 2d arrays in the port list, can you fix this?

You're right. Verilog does not support 2D arrays for port lists. I apologize for the oversight. We can fix this by using 1D arrays (vectors) to represent the input currents and spike outputs.

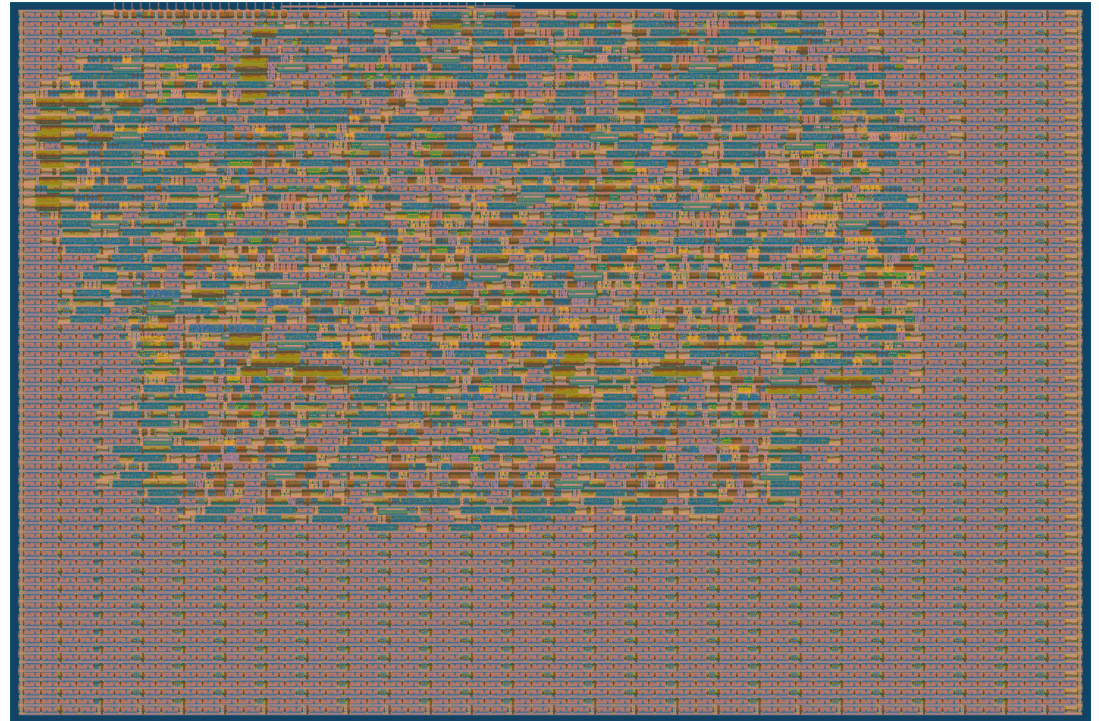
Let's modify the design to use concatenated vectors for input currents and spikes:

# Conclusions



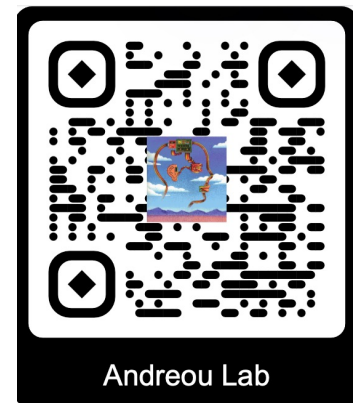
- Fully LLM written Spiking Neuron Array with SPI interface.
- Generated Verilog verified with functional simulation
- Design was submitted on Tiny Tapeout 5 using an open source RTL-> GDS flow
- 33% Density in a 320um x 200um tile
- Chips expected back on July 30 for testing
- All code, scripts, and ChatGPT transcripts are available on Github.

- One of the first fully AI described ASICs
- Not an easy process, but a fully Natural Language -> Verilog flow is possible
- For now, it requires extensive verification
- Future work – Given a top module can LLMs automate functional verification & stimulus generation?



Layout for LLM Spiking Neuron Array

# Variations of our team picture generated by Dall-E



<https://andreoulab.net>



# Telluride Neuromorphic Engineering Workshop



The core of Telluride is broken into separate **Topic Areas**. Each Topic Area is guided by a group of experts who will provide tutorials, lectures, and hands-on project ideas. Participants should expect to be exposed to all of the Topic Areas, but will generally focus on one or two to work on during the event; see [past workshops](#) for examples. There are also a series of (mostly) neuroscience talks during 2nd week; see below.

[NIC24](#)

Neuromorphic integrated circuits

[AUD24](#)

Understanding the auditory brain with neural networks

[L&T24](#)

Language and thought

[SPA24](#)

Neuromorphic systems for space applications



[CNS24](#)

Computational Neuroscience Talks

30th year anniversary:  
Join for 3 weeks of fun @ 9000 feet