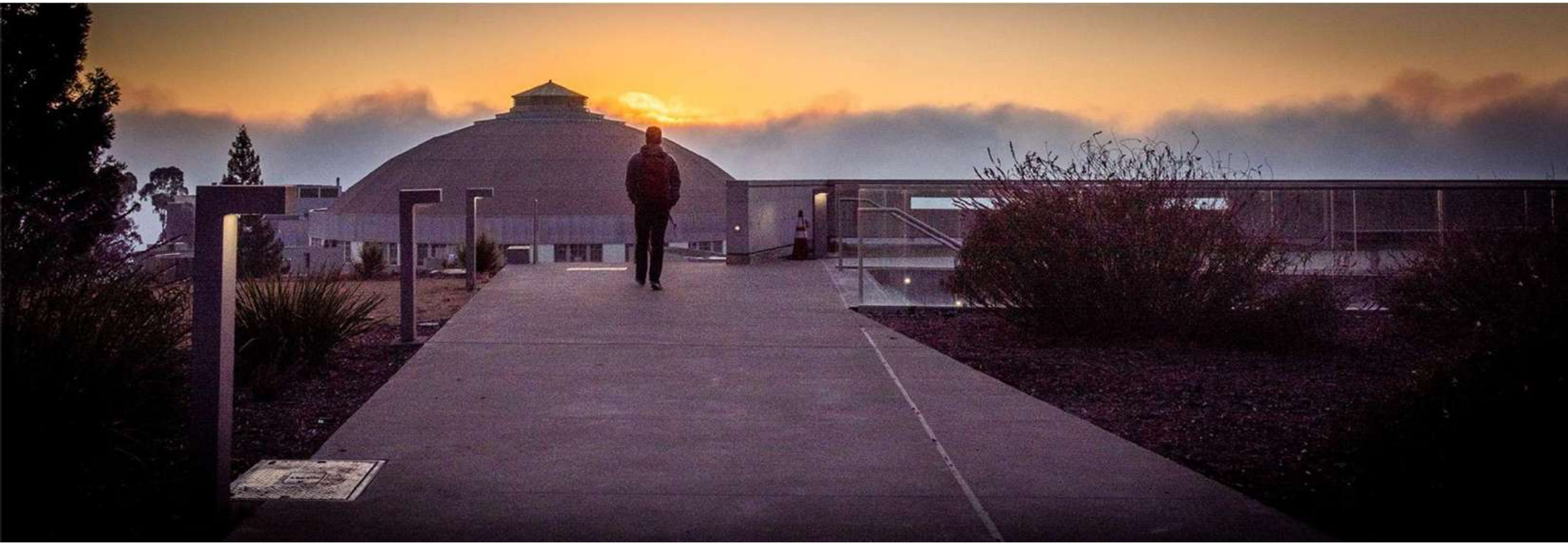
BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

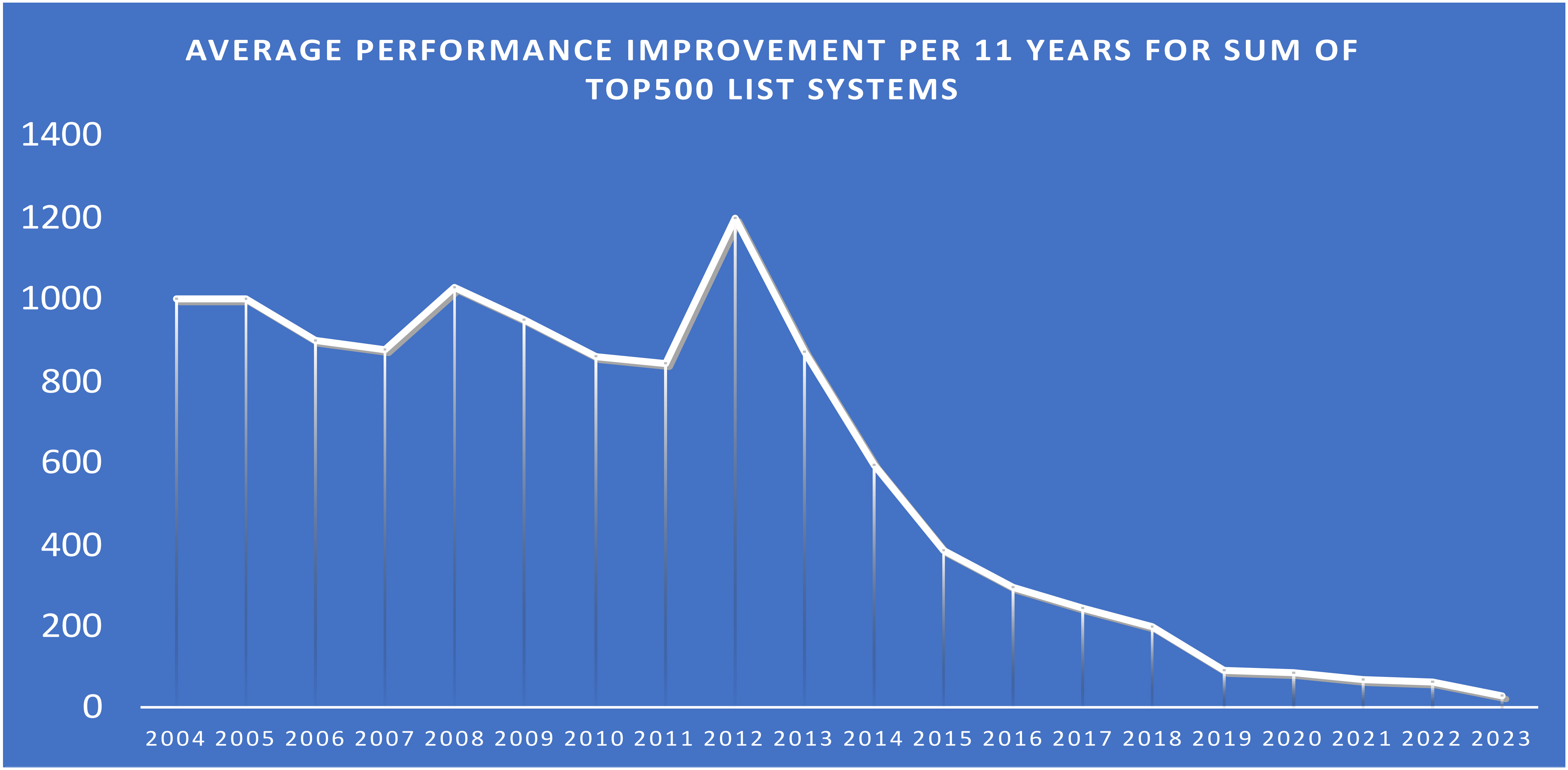
Reliable Novel Compute Methods for Unreliable Environments

Presenter: George Michelogiannakis
Applied Math and Computational Research Division (AMCR), LBNL
mihelog@lbl.gov



HPC's Future If We Do Not Change Course

"Business as usual" is not sustainable



Credit: John Shalf, LBNL

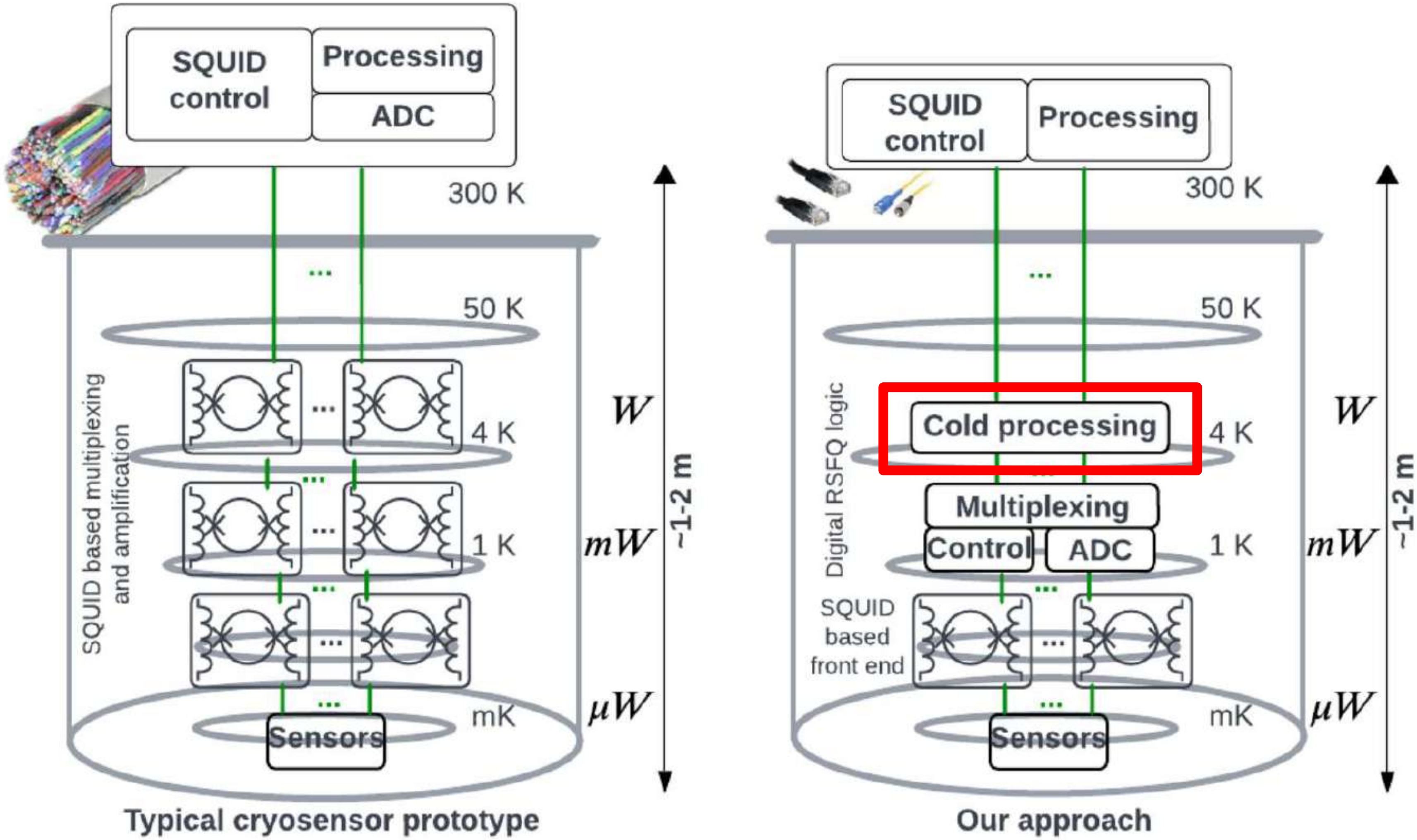
Today’s Topics

- Superconducting digital computing
- Reliable computation on unreliable hardware and environments
  - (at room temperature)

Superconducting Digital Computing

Move Compute Closer to Cryogenic Sensors/Qubits

- Expensive and noise-prone cables to move data to room temperature
- Lower temperature environments are more noisy

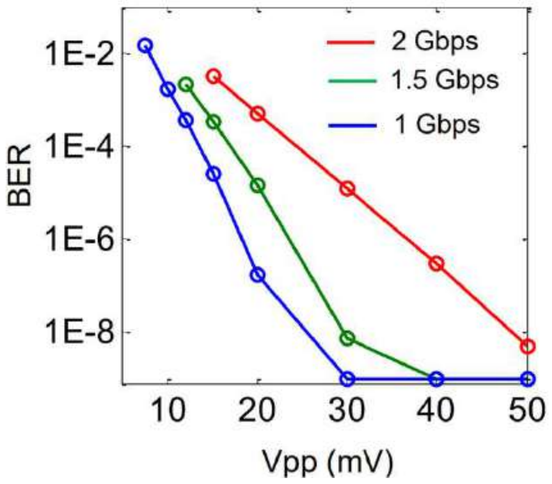


Typical cryosensor prototype

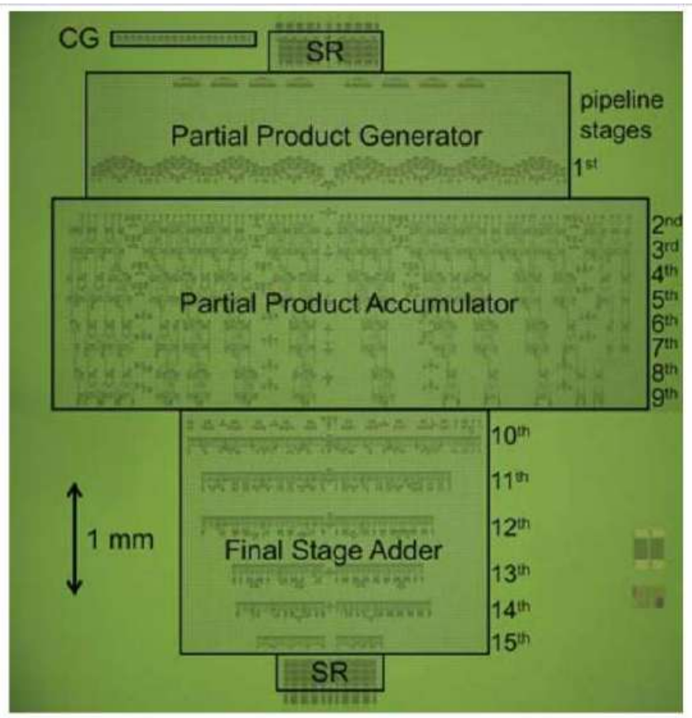
Our approach

Challenges

- Device density
  - Makes area a primary constraint
  - And memory capacity
- Cables to room temperature
- Reliability under harsh environments
  - Or with cooling and device variations



P Pintus et al., “Ultralow voltage, high-speed, and energy-efficient cryogenic electro-optic modulator”, Optica Vol. 9, Issue 10, pp. 1176-1182 (2022)

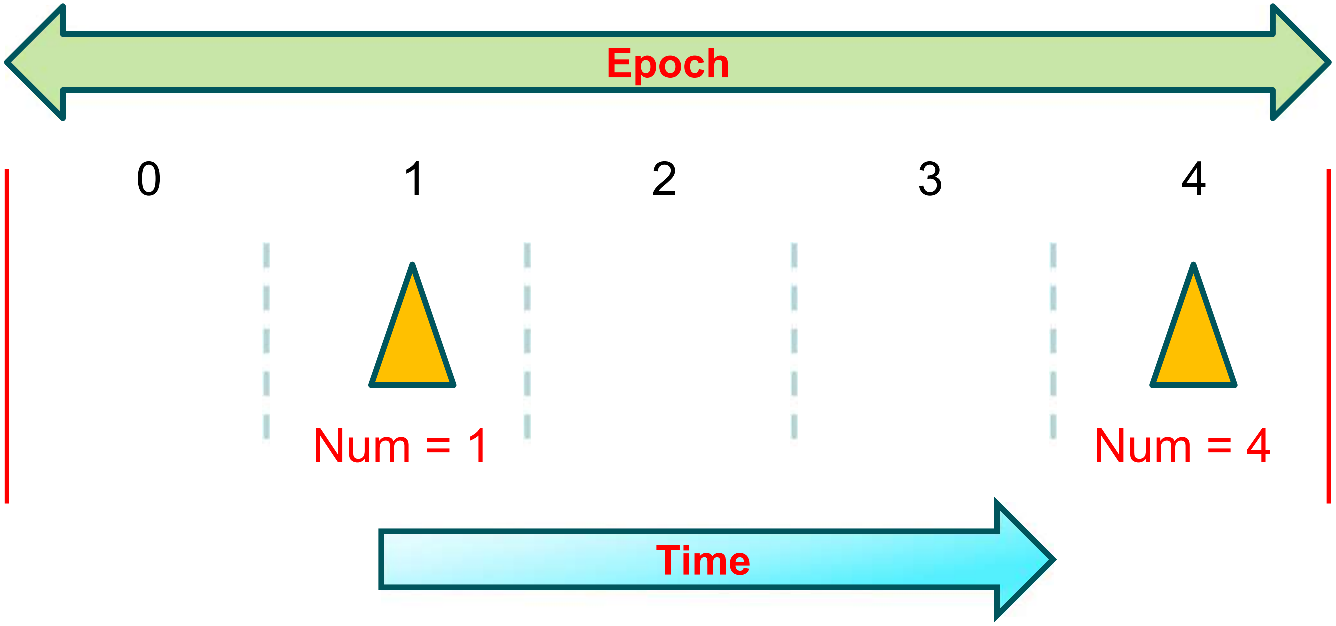


I Nagaoka et al., “A 48GHz 5.6mW Gate-Level-Pipelined Multiplier Using Single-Flux Quantum Logic”, ISSCC 2019

Data Encoding in Race Logic

An epoch contains N time slots. A pulse in time slot “I” encodes the value “I”

- Epochs repeat
- Epoch duration = TimeSlotDuration × NumTimeSlots
- Each pulse represents an equivalent 2^N binary number (N = NumTimeSlots)
- Can efficiently represent non power of two number ranges



G Tzimpragos et al., “A computational temporal logic for superconducting accelerators”, ASPLOS 2020

Instead: Unipolar and Bipolar Race Logic

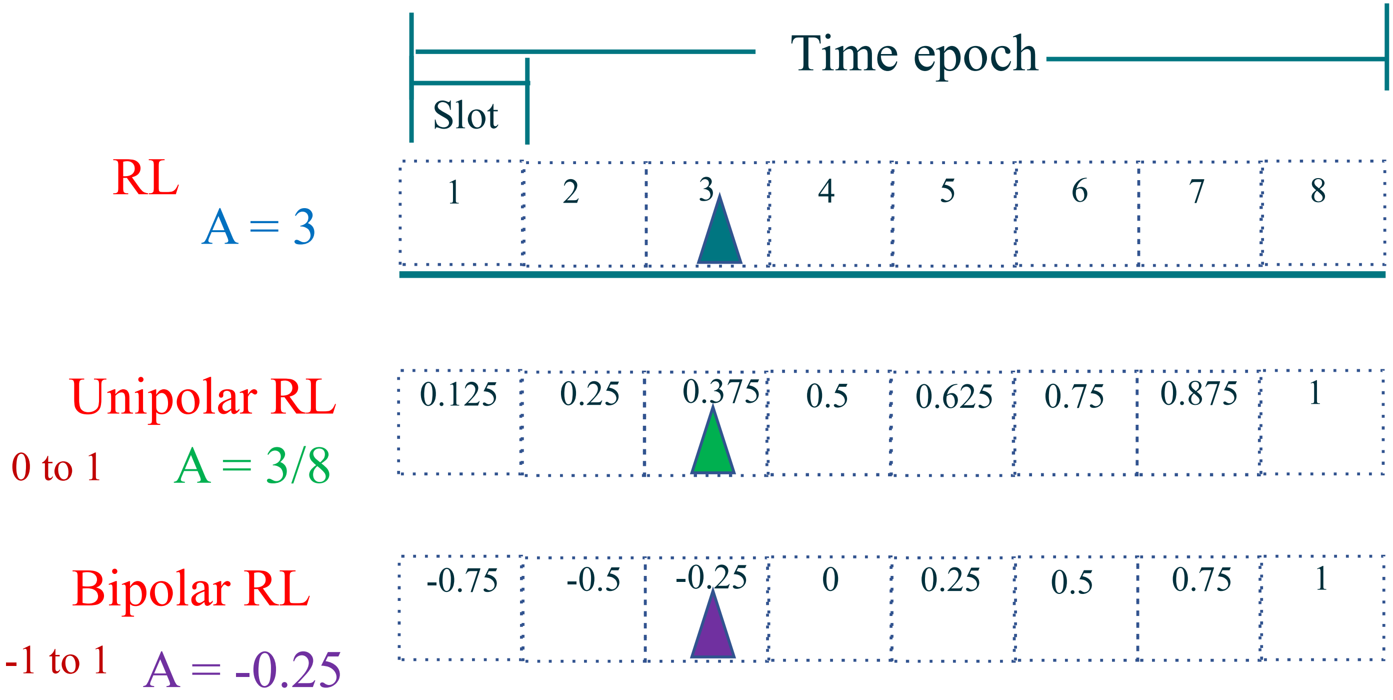
Changing the range of representation to [0,1] (unipolar)

or [-1,1] (bipolar)

To obtain bipolar representation

$$N_{max} = 8$$

$$A_b = 2A_u - 1$$



RL

A = 3

Unipolar RL

0 to 1 A = 3/8

Bipolar RL

-1 to 1 A = -0.25

1	2	3	4	5	6	7	8
0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
-0.75	-0.5	-0.25	0	0.25	0.5	0.75	1

P Gonzalez-Guerrero et al., “Temporal and SFQ pulse-streams encoding for area-efficient superconducting accelerators”, ASPLOS 2022

Pulse Train Operands

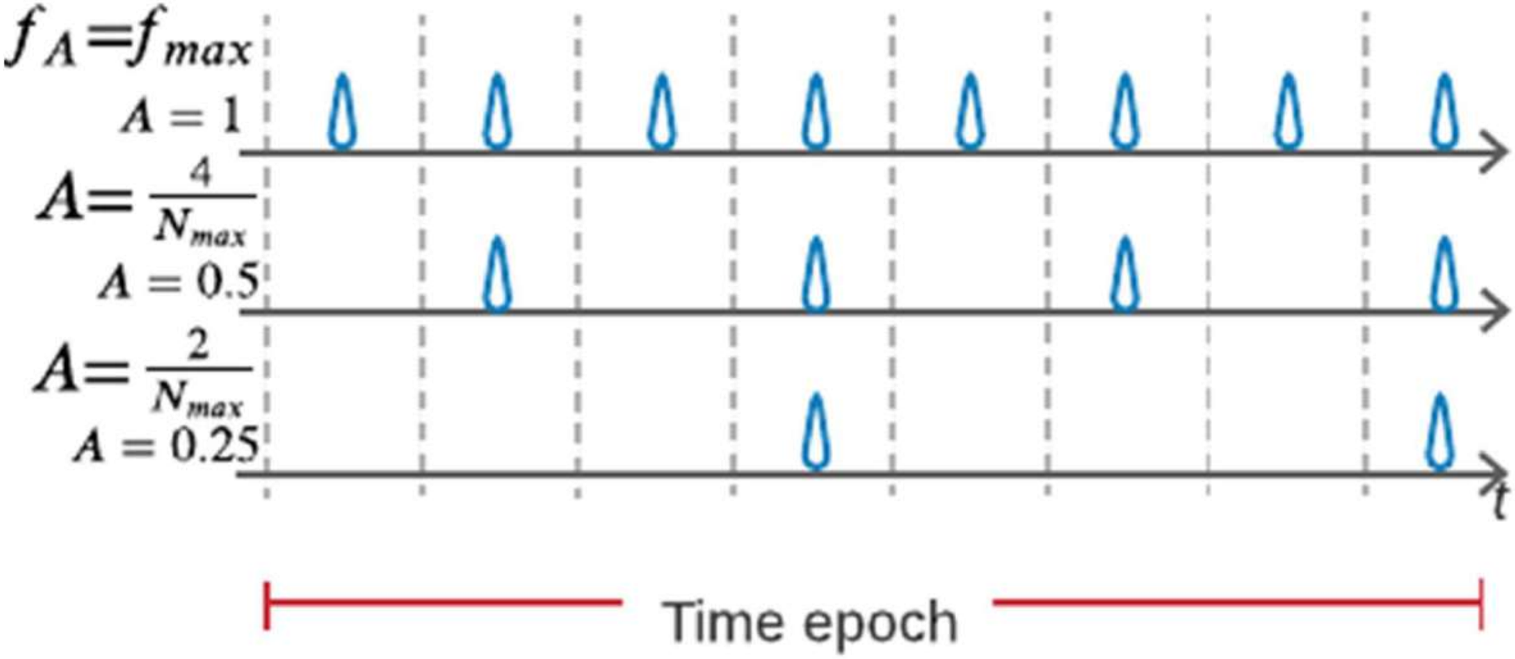
Maps a value to the number of pulses. “1” is for the maximum number of pulses

$$f_{max} \qquad N_{max} = 8$$

$$A = n/N_{max}$$

To obtain bipolar representation (not shown)

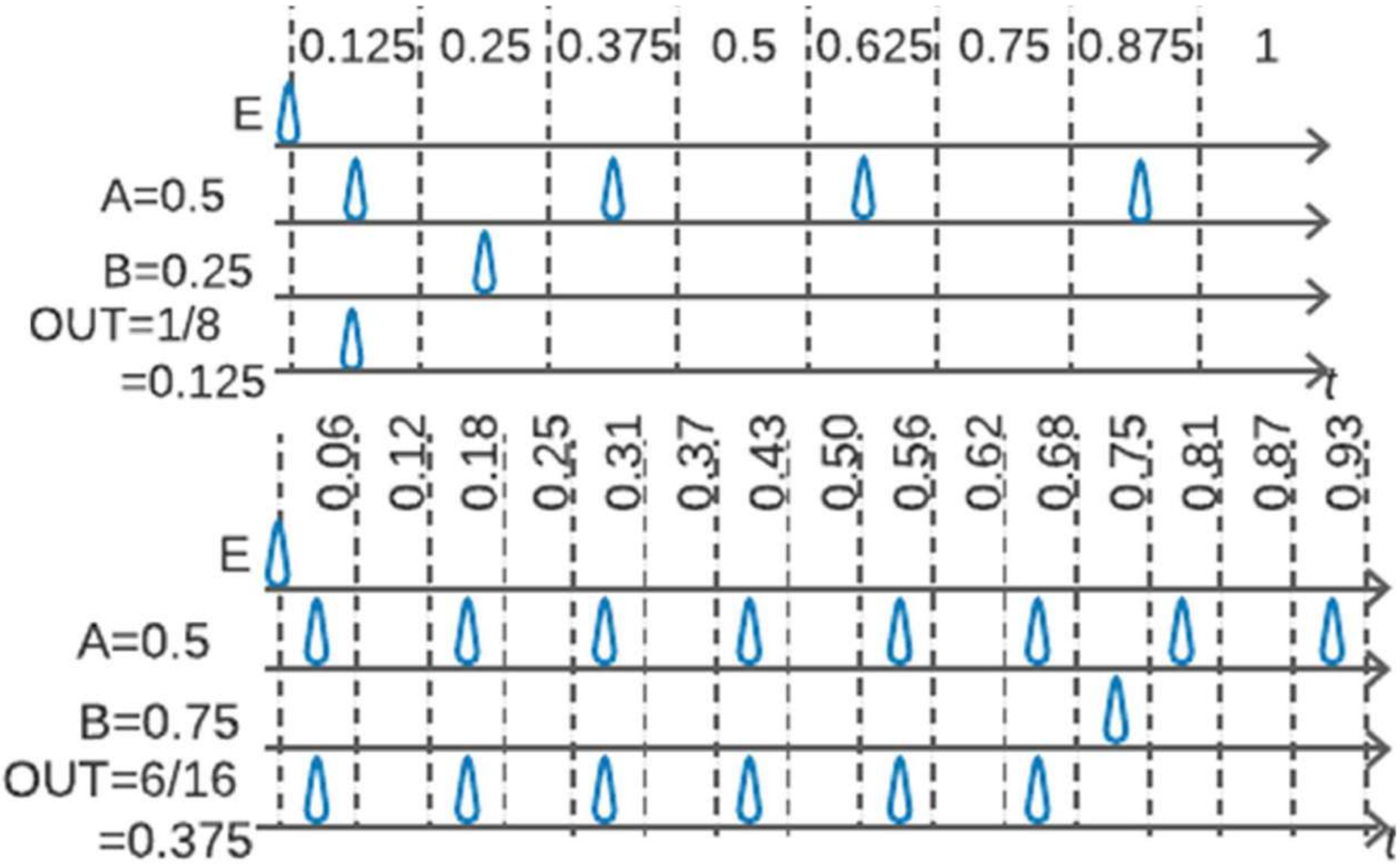
$$A_b = 2A_u - 1$$



P Gonzalez-Guerrero et al., “Temporal and SFQ pulse-streams encoding for area-efficient superconducting accelerators”, ASPLOS 2022

U-SFQ: Race Logic and Pulse Stream Operands

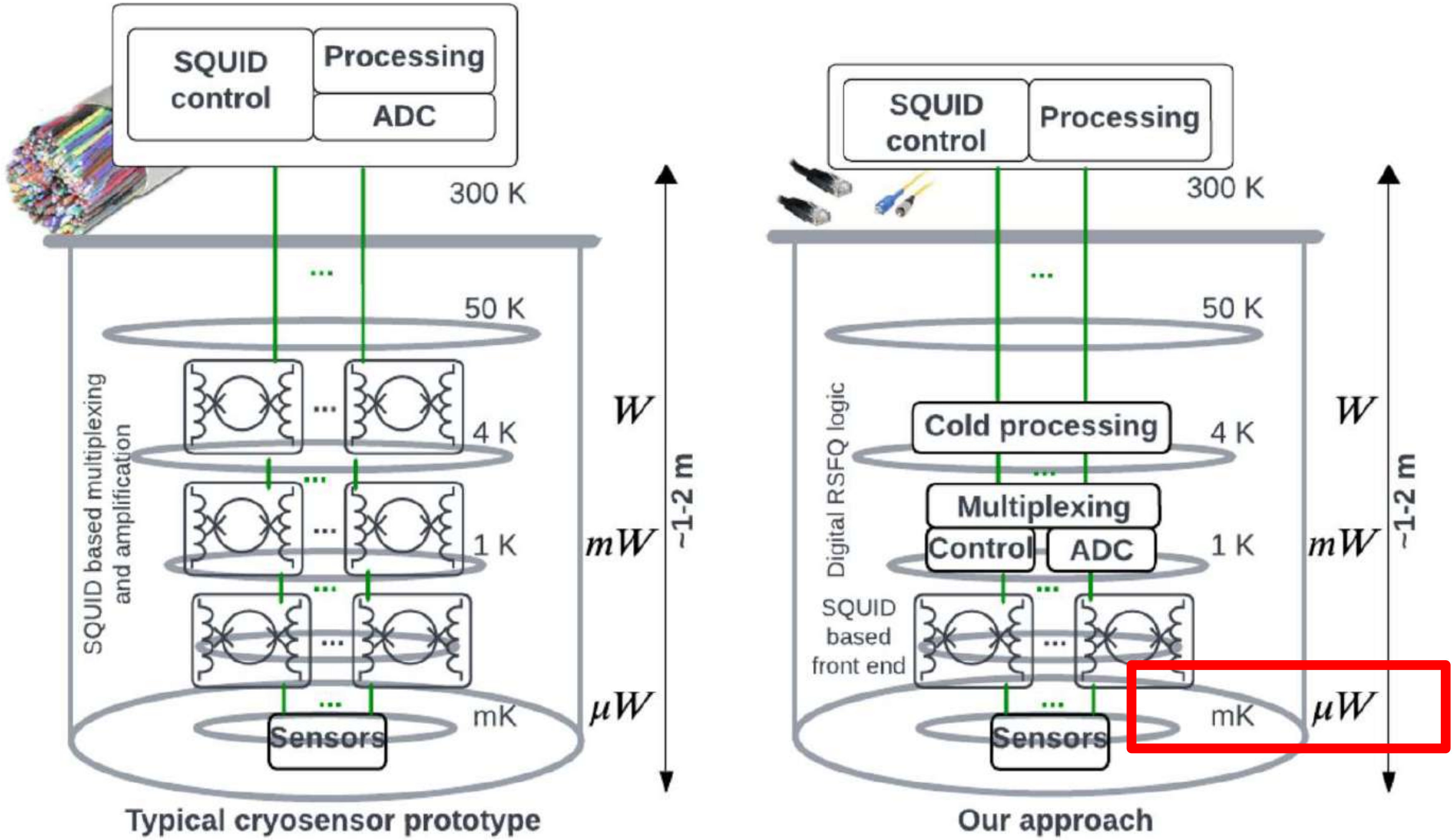
This shows a multiplication. The output is a pulse train



P Gonzalez-Guerrero et al., “Temporal and SFQ pulse-streams encoding for area-efficient superconducting accelerators”, ASPLOS 2022

What is Keeping us Away From mK?

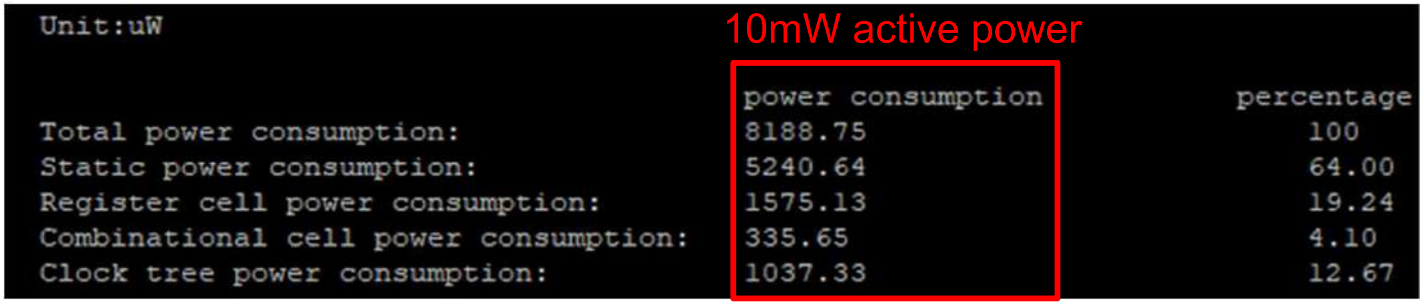
Such as 300 mK



https://github.com/andreamurillomtz/decimation_filter.git

Power Limits

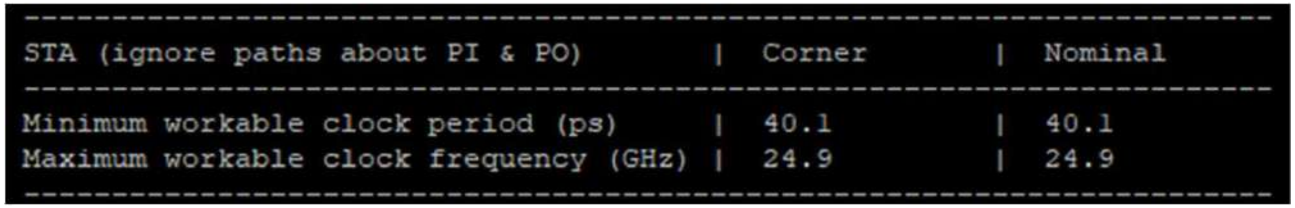
Simple 16-bit adder. Results with qPalace. 50% activity factor in every input



10mW active power

```
Unit:uW

                                           power consumption            percentage
Total power consumption:                   8188.75                         100
Static power consumption:                  5240.64                         64.00
Register cell power consumption:           1575.13                         19.24
Combinational cell power consumption:      335.65                          4.10
Clock tree power consumption:              1037.33                         12.67
```



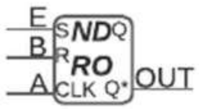
```
-----------------------------------------------------------------------
STA (ignore paths about PI & PO)         |  Corner          |  Nominal
-----------------------------------------------------------------------
Minimum workable clock period (ps)       |  40.1            |  40.1
Maximum workable clock frequency (GHz)   |  24.9            |  24.9
-----------------------------------------------------------------------
```

~1000x gap with µW target

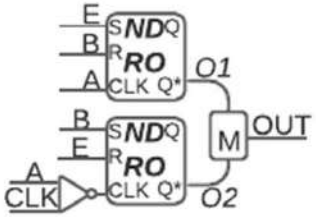
How to reduce power?

- ERSFQ/eSFQ (static power)
- Fewer gates (smaller circuit)
- Lower activity factor
- Lower clock frequency

U-SFQ's Power Requirements

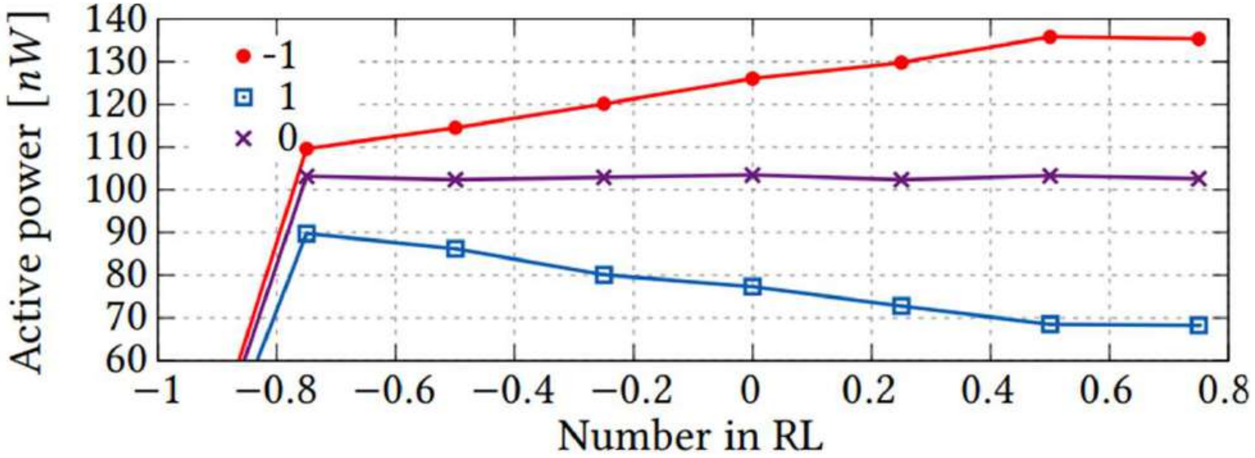


Unipolar SFQ multiplier



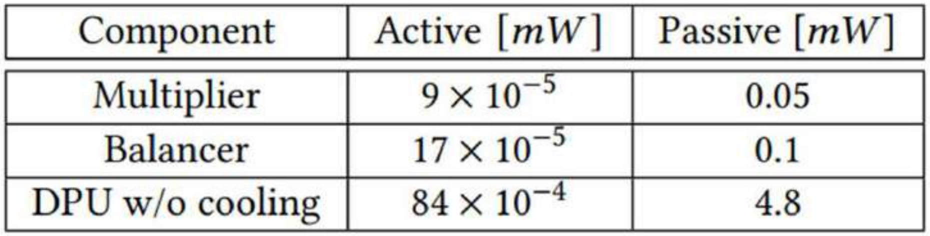
Bipolar SFQ multiplier

Power depends on numerical value of inputs. Higher numbers -> more pulses



Active power consumption for the bipolar multiplier, using three different pulse streams frequencies representing the numbers −1, 1, and 0. We vary the RL input from -1 to 1.

32-input 8-bit U-SFQ dot-product unit



Component	Active [mW]	Passive [mW]
Multiplier	9×10^{-5}	0.05
Balancer	17×10^{-5}	0.1
DPU w/o cooling	84×10^{-4}	4.8

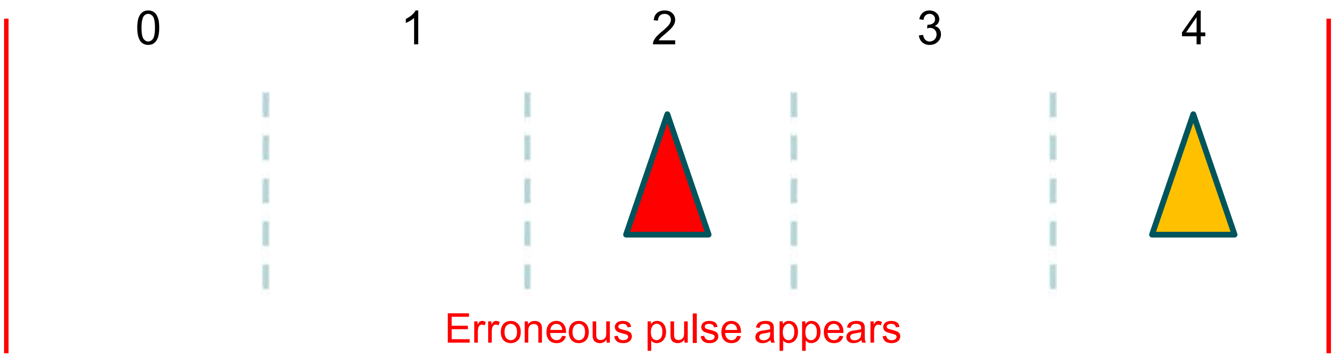
RSFQ. RL and pulse train inputs set to half the maximum value

P Gonzalez-Guerrero et al., “Temporal and SFQ pulse-streams encoding for area-efficient superconducting accelerators”, ASPLOS 2022

Noise / Variability

From external factors, cooling imperfections, and device variability

- SFQ devices are usually biased around 70% of their switching voltage
- If noise levels are high, fluctuations in voltage can cause JJs to produce erroneous pulses
  - SFQ devices due to their low switching energy and particularly susceptible
- Noisy grounds and power supplies can cause issues with device performance
- These are already challenges in 4K, worse at 300mK
  - SFQ devices need to be tuned
- **Can compute models accept device errors but bound the numerical error?**
- **We need a solid noise model and resulting simulation models**

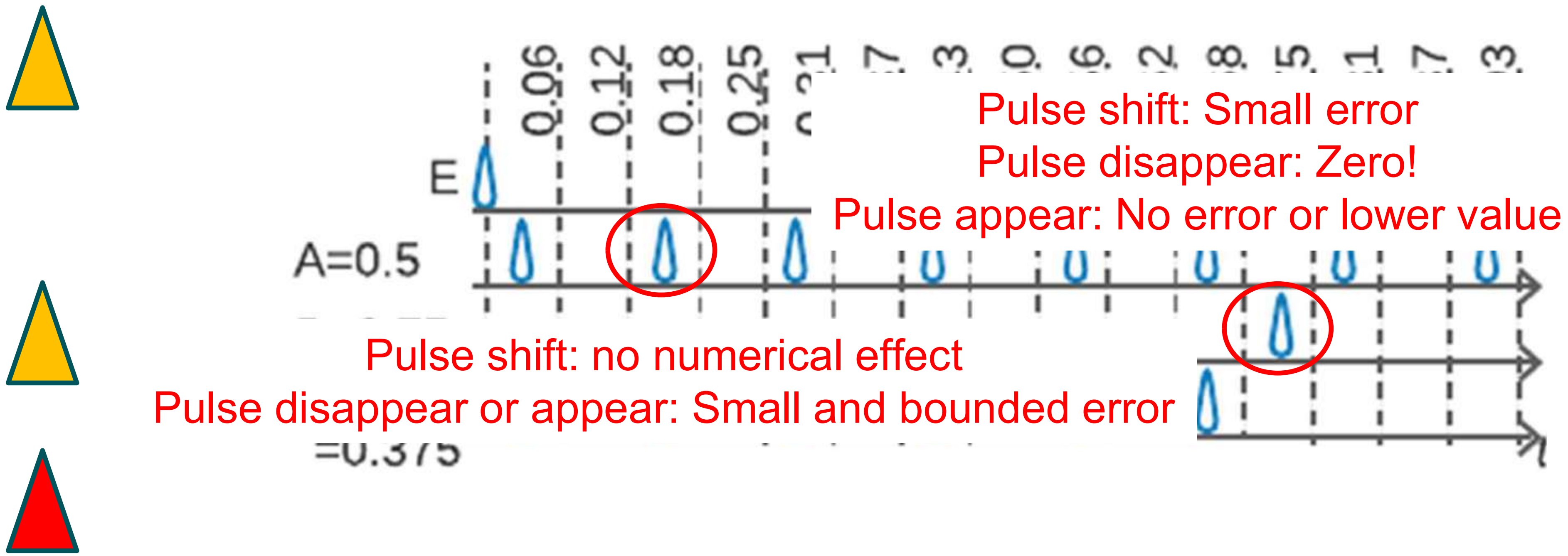


How Do Compute Models Affect Error?

Predictability of errors matters too, not just its numerical impact

- For binary, pulse trains, and race logic:
  - How does an erroneous pulse appearing affect the represented value?
  - How about a pulse disappearing?
  - How about a pulse shifting?

Binary



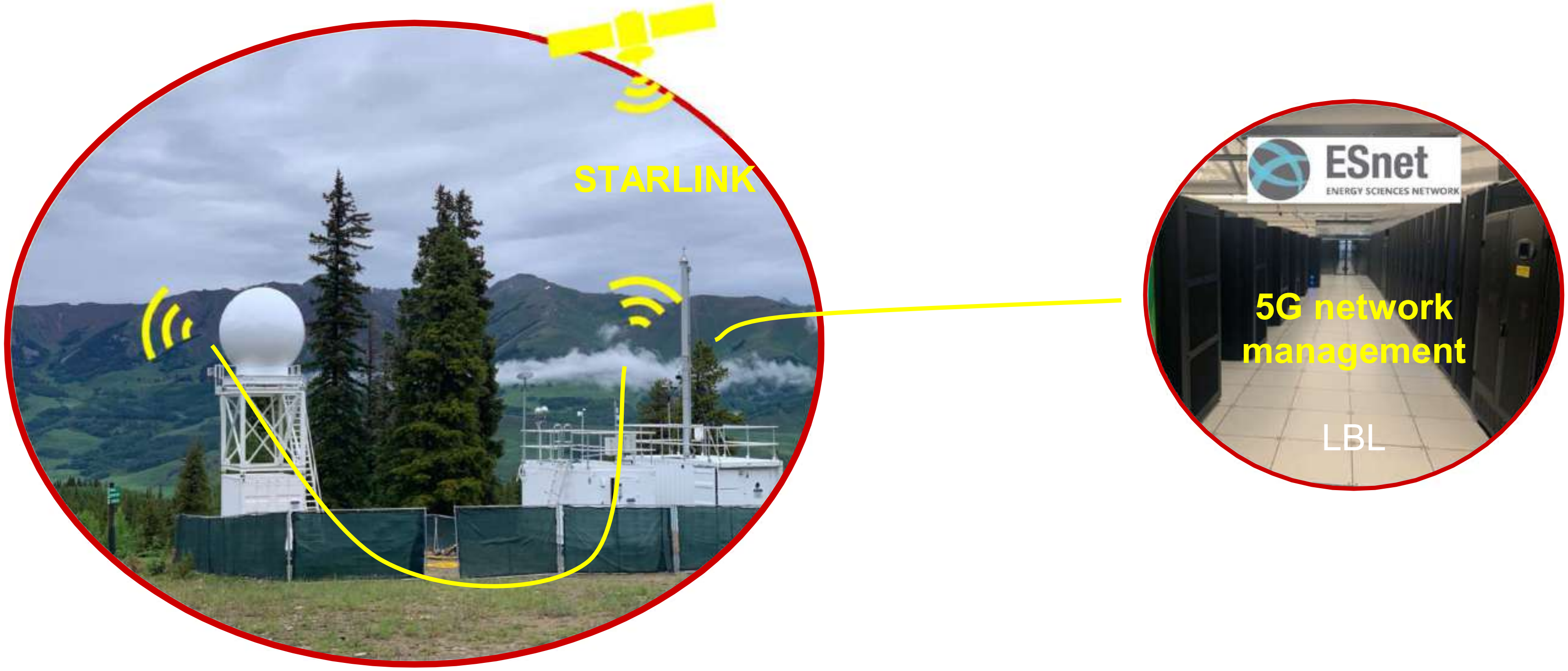
Pulse shift: Small error
Pulse disappear: Zero!
Pulse appear: No error or lower value

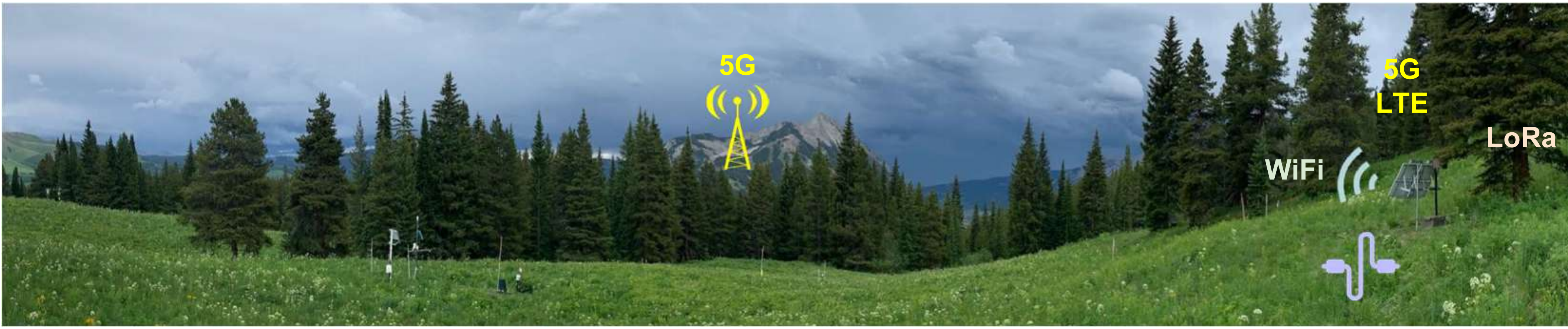
Pulse shift: no numerical effect
Pulse disappear or appear: Small and bounded error

Erroneous pulse

Reliable Computation on Unreliable Devices and Environments

Reliable Distributed Computing With Unreliable Hardware in Unreliable Environments



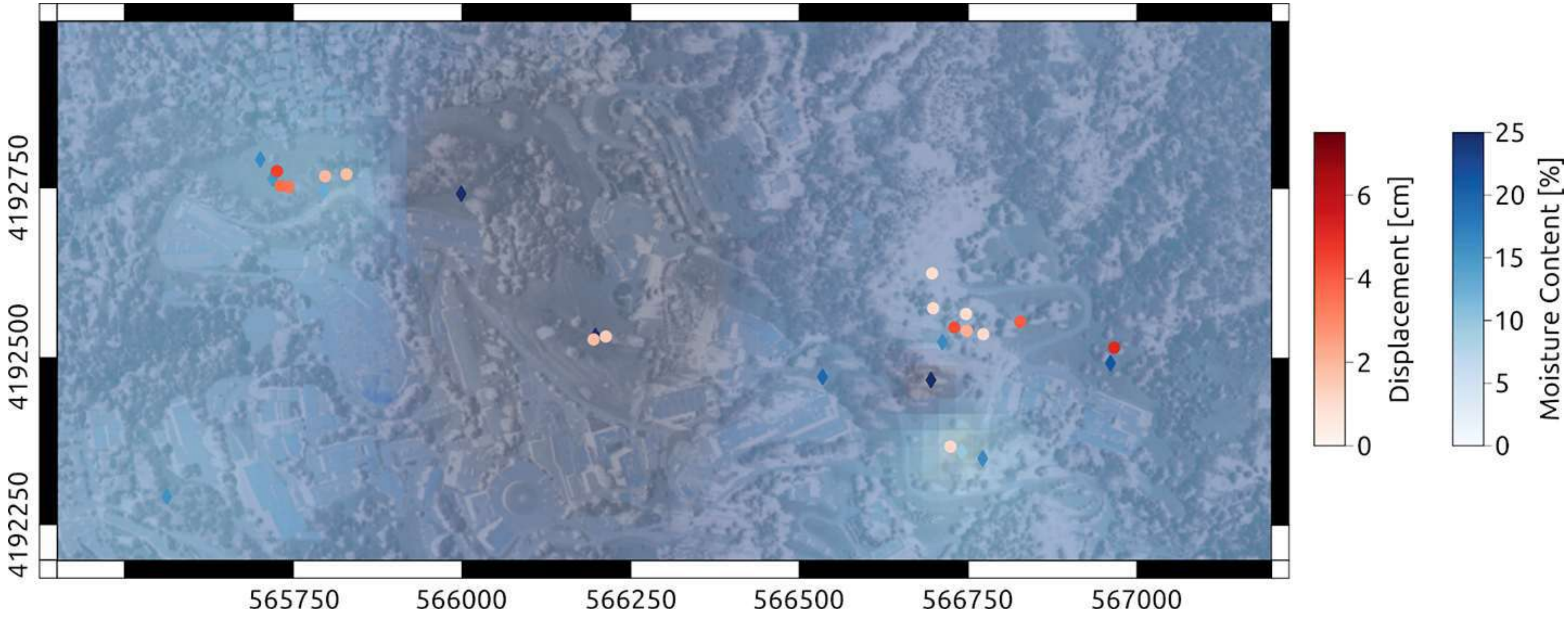


Landslide Monitoring

Geophysical sensors to derive a predictive understanding

Sensor density of more than 175 sensors per square kilometer

Map of the study site showing the measured and interpolated moisture content (diamond symbols and blue shades) and the measured deformation at a subset of the monitored locations (white to red dots)

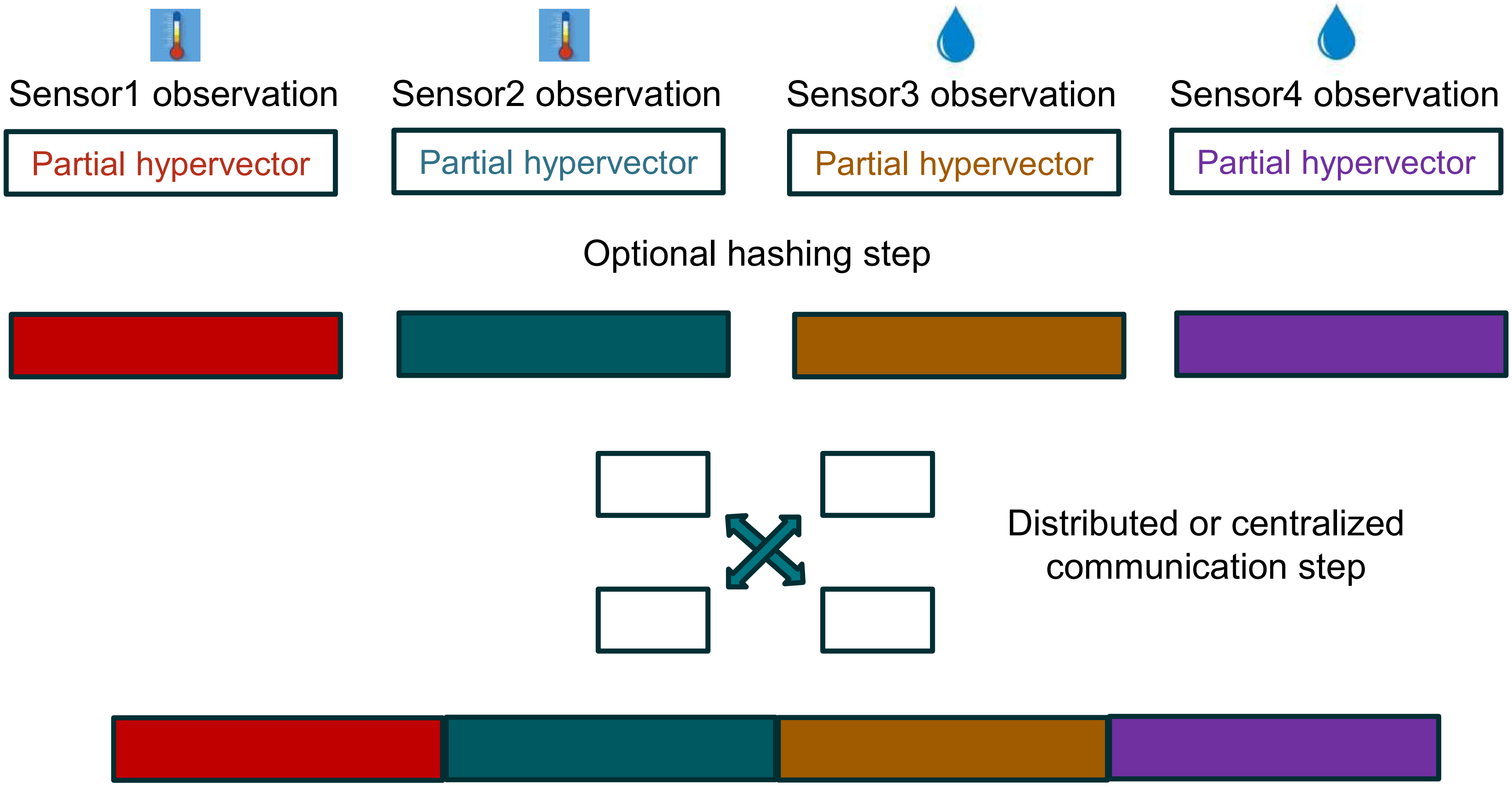


“Predictive monitoring of urban slope instabilities using geophysics and wireless sensor networks”, The Leading Edge. 2023;42(9):634-643

Future work

Reliable AI in an Unreliable Environment

With privacy in case communication is monitored or sensors get reverse engineered



Sensor1 observation

Partial hypervector

Sensor2 observation

Partial hypervector

Sensor3 observation

Partial hypervector

Sensor4 observation

Partial hypervector

Optional hashing step

Distributed or centralized communication step

Entire hypervector in one location. Can produce final result

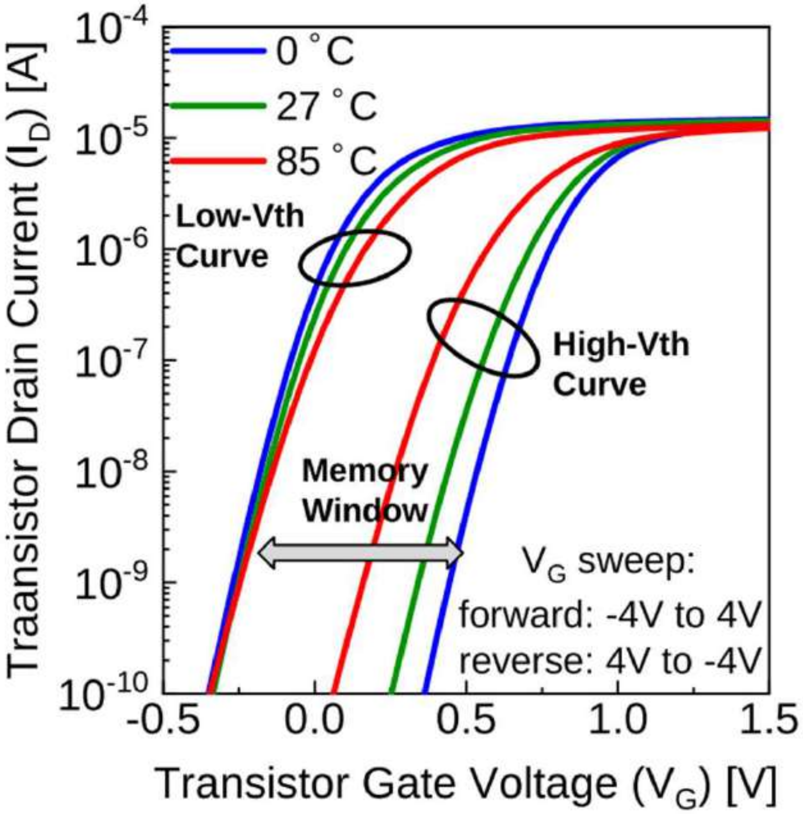
With many sensors, a minority of the hypervector missing will not affect final classification result, with high probability

Alternative models such as CNNs also possible but resilience has not been in focus yet

Bit Error Rate (BER) in Emerging Devices

Ferroelectrics as an example

- BER depends on applied gate voltage, process variation, and temperature
  - BER is affected asymmetrically for 0s and 1s with temperature
  - 0 to become 1 more probable
  - Almost a linear relationship
- At 85C and different read voltages:
  - 14nm FeFET, 10nm Fe layer
  - 0.1V
    - $P_{01} = 2.198\%$
    - $P_{10} = 1.090\%$
  - 0.25V
    - $P_{01} = 2.098\%$
    - $P_{10} = 0.190\%$

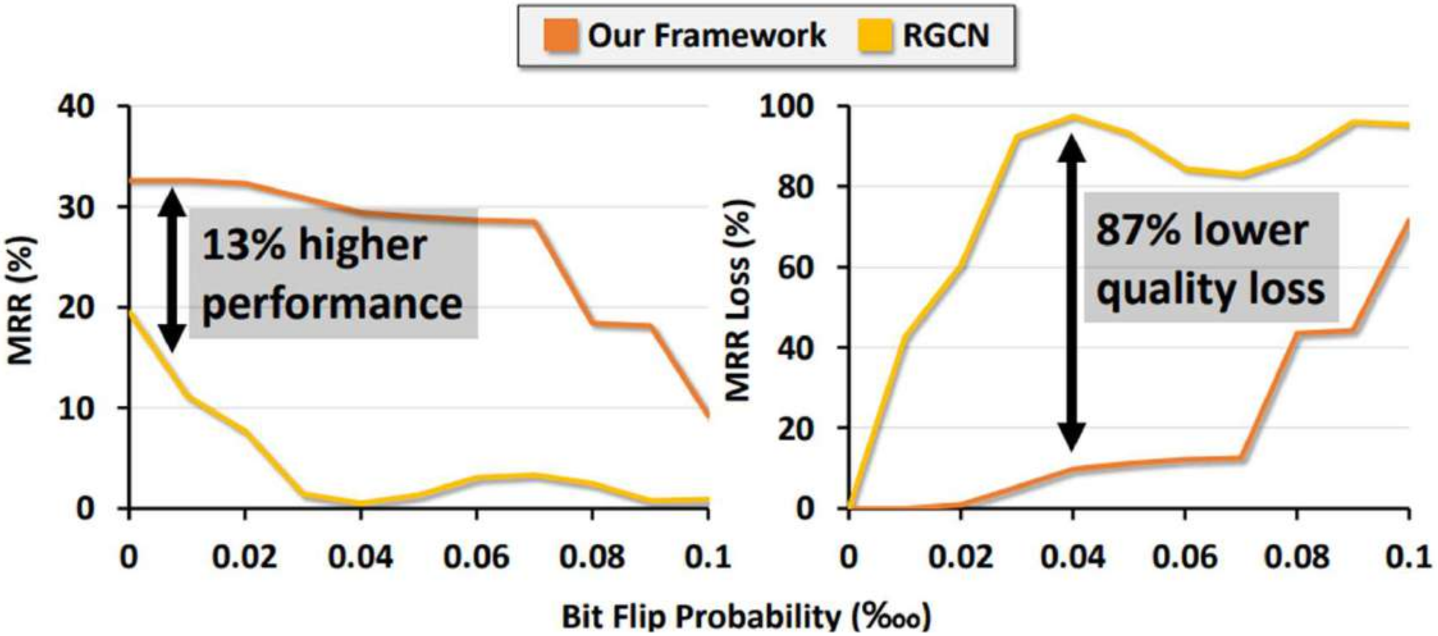


"FeFET-Based Binarized Neural Networks Under Temperature-Dependent Bit Errors", IEEE Transactions on Computers, July 2022

Reliable AI Models can Mask Bit Errors

So we can allow devices to be error prone thus more efficient

- Framework based on hyperdimensional computing (HDC) and compared against a graph neural network (GNN)
  - Study includes models of FeFETs as well as FeFET hardware blocks for memory, element-wise vector product, and vector-matrix product



MRR: Mean reciprocal rank. X-axis on a per-thousand scale

MRR measures the average reciprocal rank of predictions, with higher scores indicating more accurate predictions ranked closer to the top

“Reliable Hyperdimensional Reasoning on Unreliable Emerging Technologies”, IEEE ICCAD, 2023

Conclusion

Lets adapt compute models to mask unreliability, reduce area, and reduce power

This will allow more efficient compute and in domains that are impractical now

Questions?