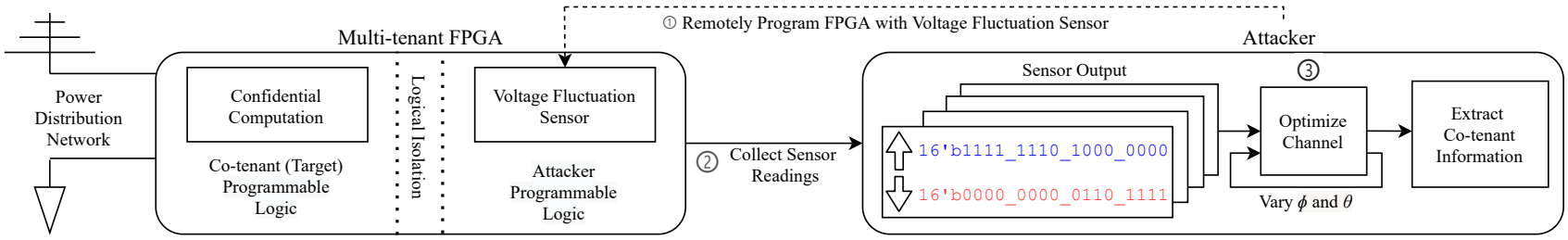
Classifying Computations on Multi-Tenant FPGAs

Colin Drewes [1] Mustafa Gobulukoglu [1] Olivia Weng [1]
Steven Harris [1] Winnie Wang [1] William Hunter [3]
Christopher McCarty [3] Ryan Kastner [1] Dustin Richmond [2]

[1]UCSD [2]UW [3]GTRI

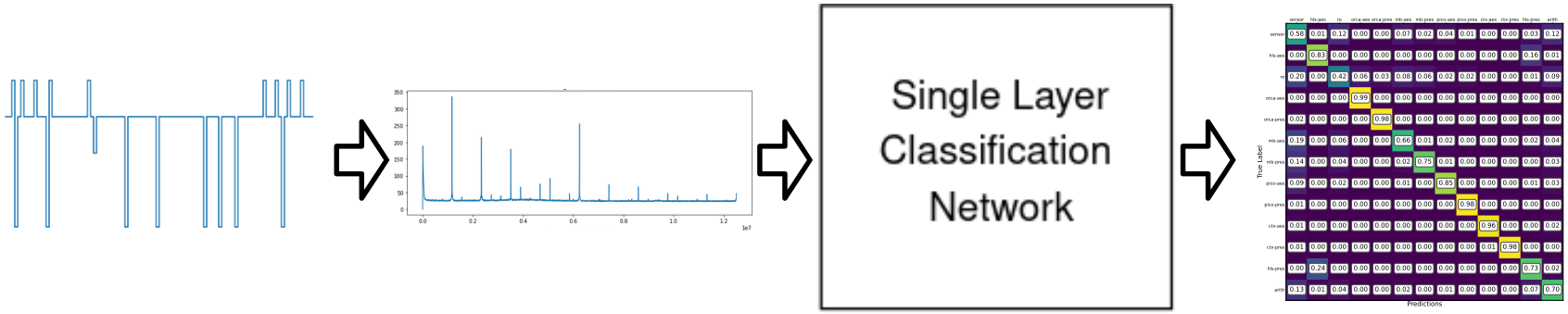
2021

Proposed Threat-Model



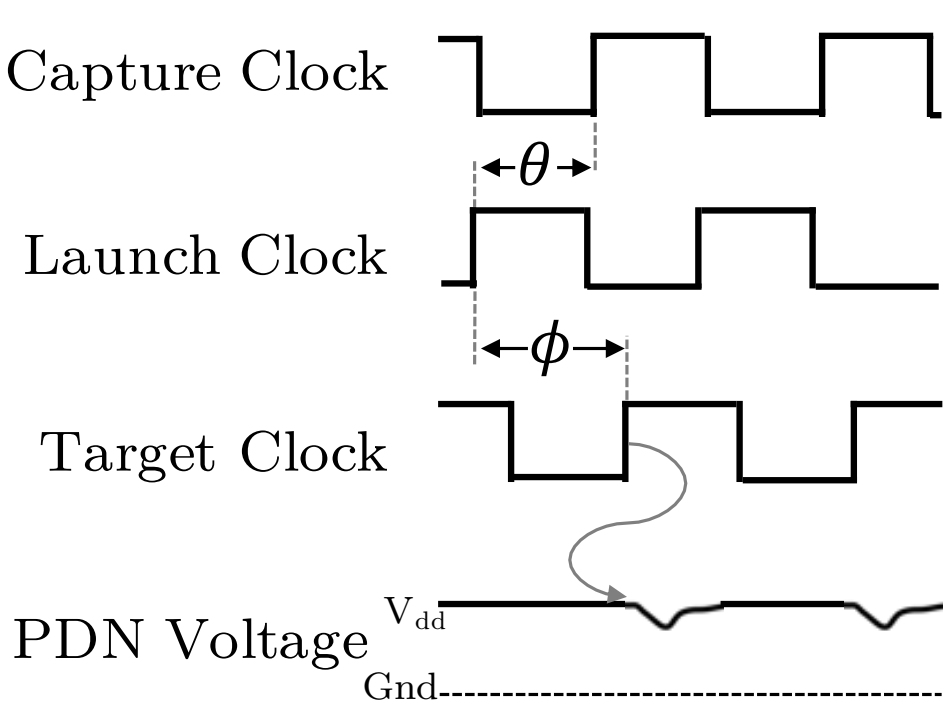
1. FPGAs are powerful, but expensive $\rightarrow$ time-sharing, virtualization, proprietary IP
2. All expose a side-channel through shared power distribution
3. Classifier trained on local board data—needs to generalize
4. Can be leveraged to determine aspects of co-located computation (Type of computation? Implementation?)

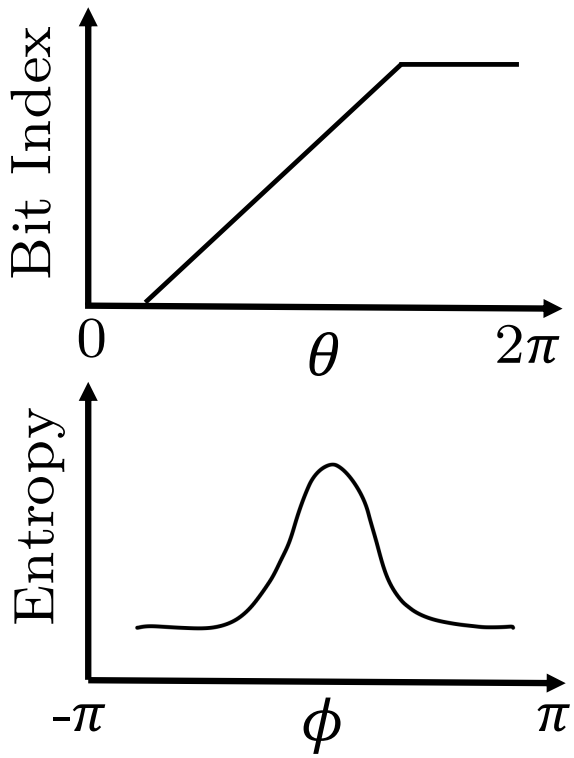
Co-Located Applications



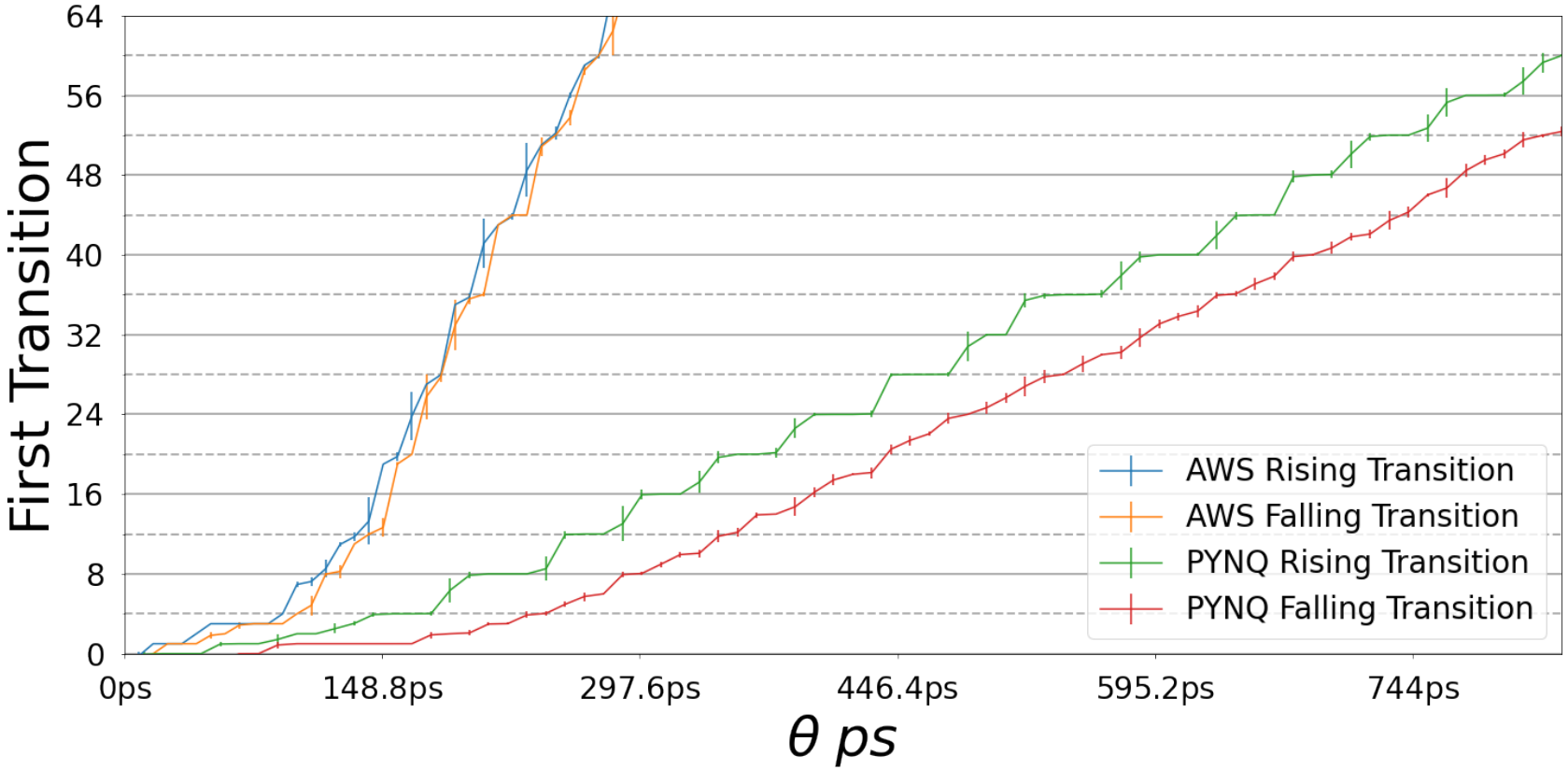
1. Sensor Only
2. Ring Oscillators
3. Arithmetic Heavy
4. Cryptographic Cores (AES, PRESENT)
   - 4.1 Custom IP AES + PRESENT
   - 4.2 Orca
   - 4.3 PicoRV
   - 4.4 Microblaze
   - 4.5 CortexM3

Refining the Problem





Capture/Launch Clock Tuning (θ) — First Index



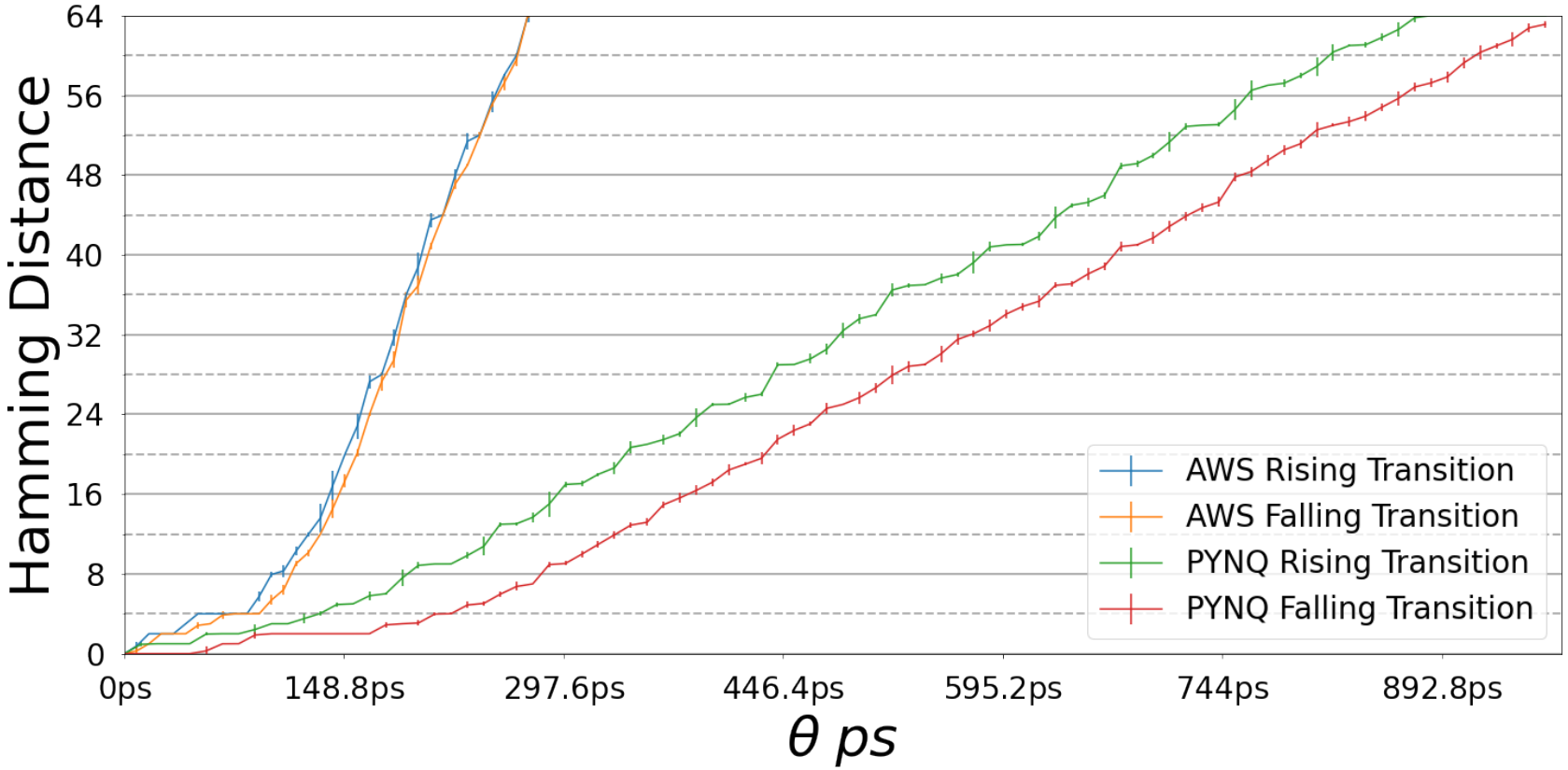
Output Sequence

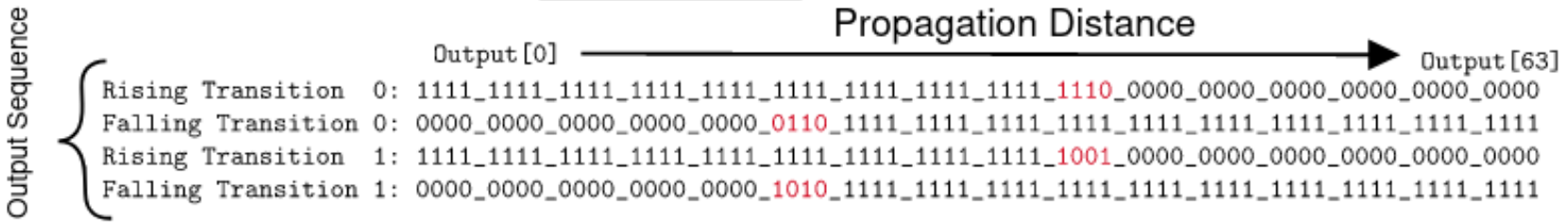
Propagation Distance

Output[0] → Output[63]

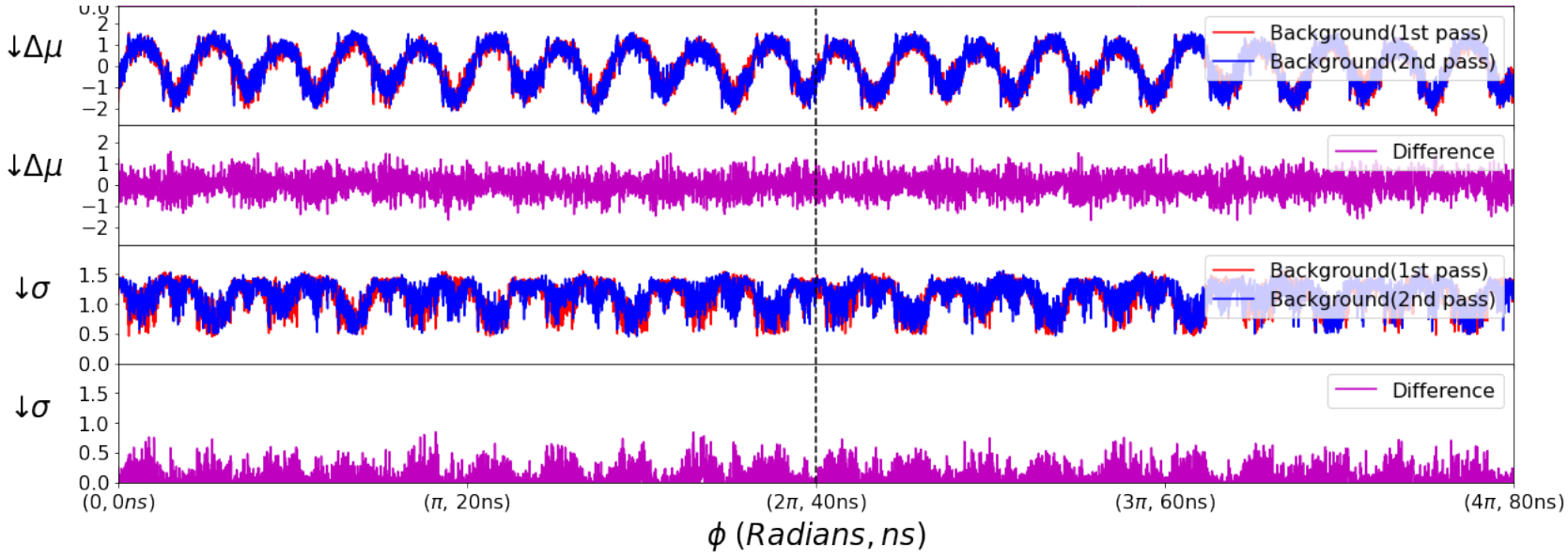
```
Rising Transition  0: 1111_1111_1111_1111_1111_1111_1111_1111_1111_1110_0000_0000_0000_0000_0000_0000
Falling Transition 0: 0000_0000_0000_0000_0000_0110_1111_1111_1111_1111_1111_1111_1111_1111_1111_1111
Rising Transition  1: 1111_1111_1111_1111_1111_1111_1111_1111_1111_1001_0000_0000_0000_0000_0000_0000
Falling Transition 1: 0000_0000_0000_0000_0000_1010_1111_1111_1111_1111_1111_1111_1111_1111_1111_1111
```

Capture/Launch Clock Tuning (θ) — Hamming Distance

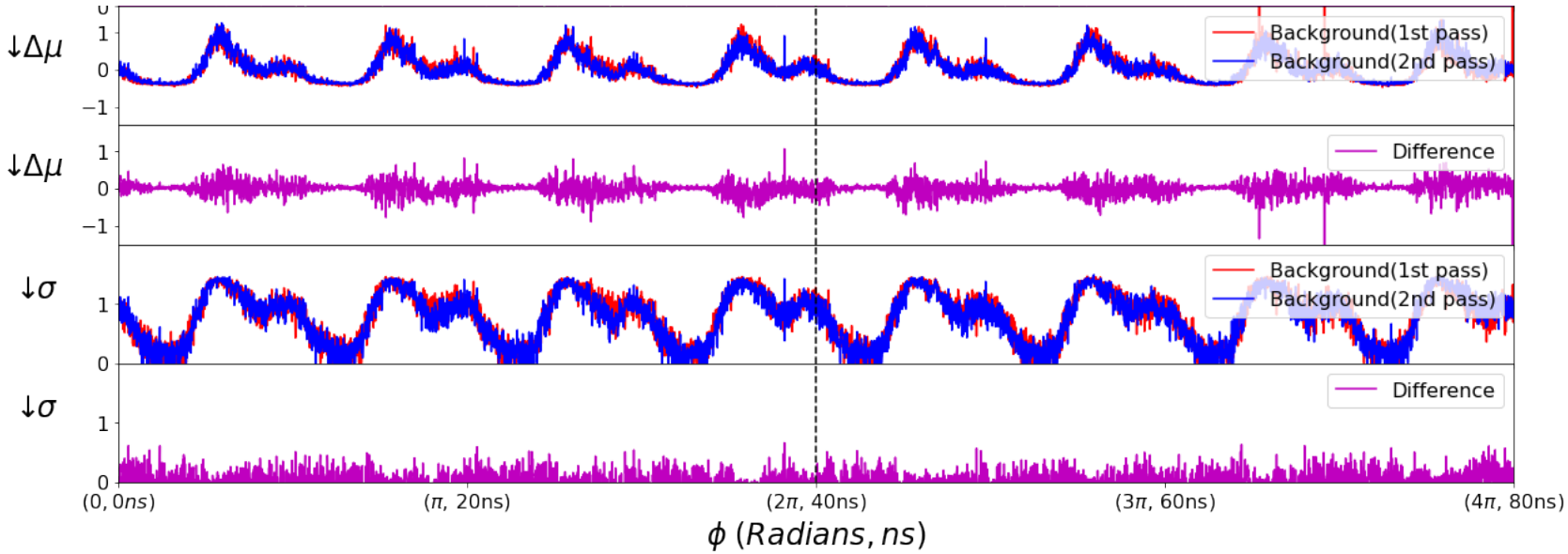




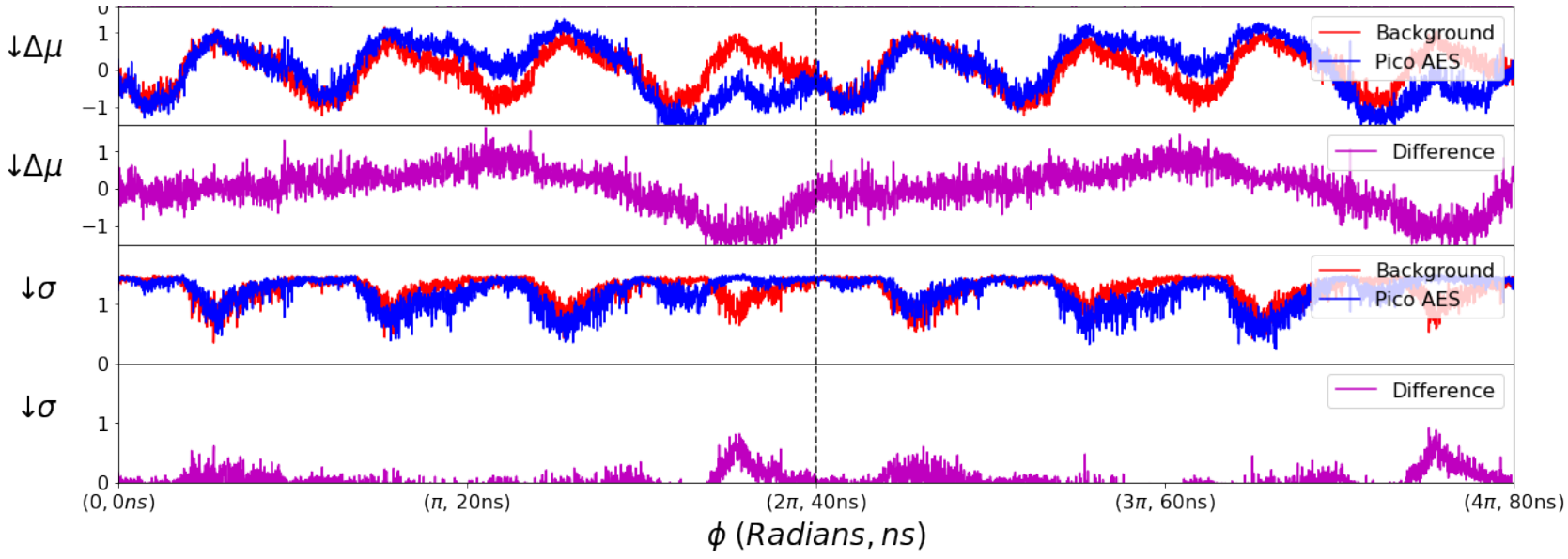
Target Clock Tuning (ϕ) — AWS Sensor Only



Target Clock Tuning (ϕ) — PNYQ Sensor Only



Target Clock Tuning (ϕ) — PNYQ Pico AES

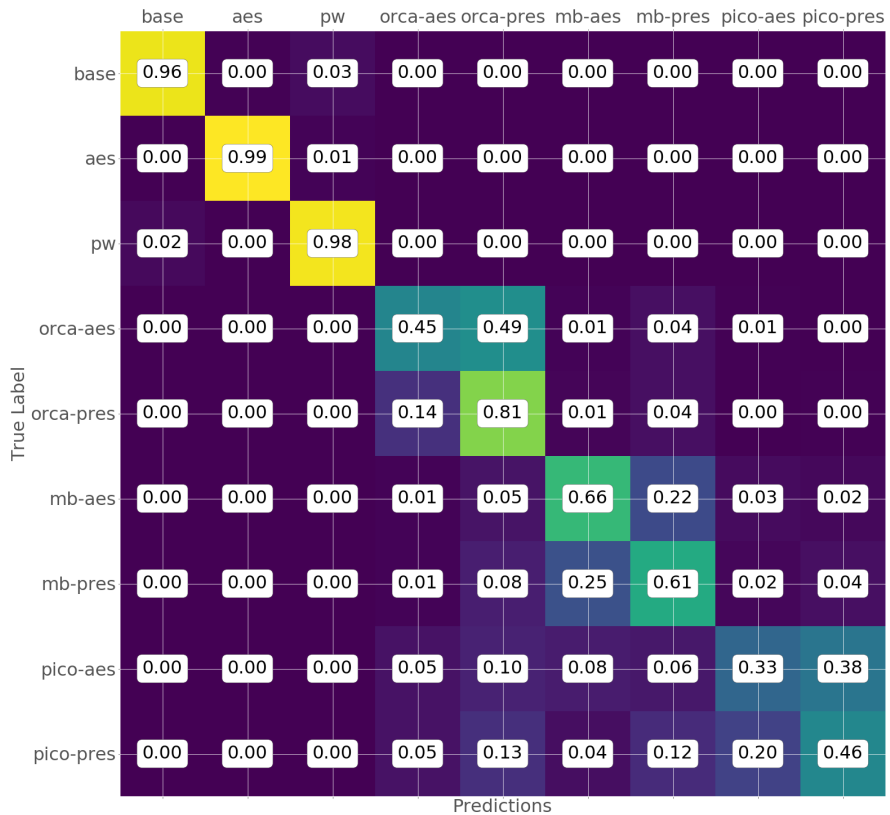


Classification

1. Sensor is tuned (θ, ϕ)
2. Sensor captures a long trace (many thousand samples)
3. FFT taken on output of sensor
4. Fed into single layer network
5. Examining the follow configurations of the sensor:
   - 5.1 ($\uparrow \theta_{min}$, ϕ_{min})
   - 5.2 ($\downarrow \theta_{max}$, ϕ_{min})
   - 5.3 ($\downarrow \theta_{max}$, ϕ_{max})
   - 5.4 ($\downarrow \theta_{max}$, ϕ_{back})
   - 5.5 ($\uparrow \theta_{max}$, ϕ_{back})

DAC Paper Limitations

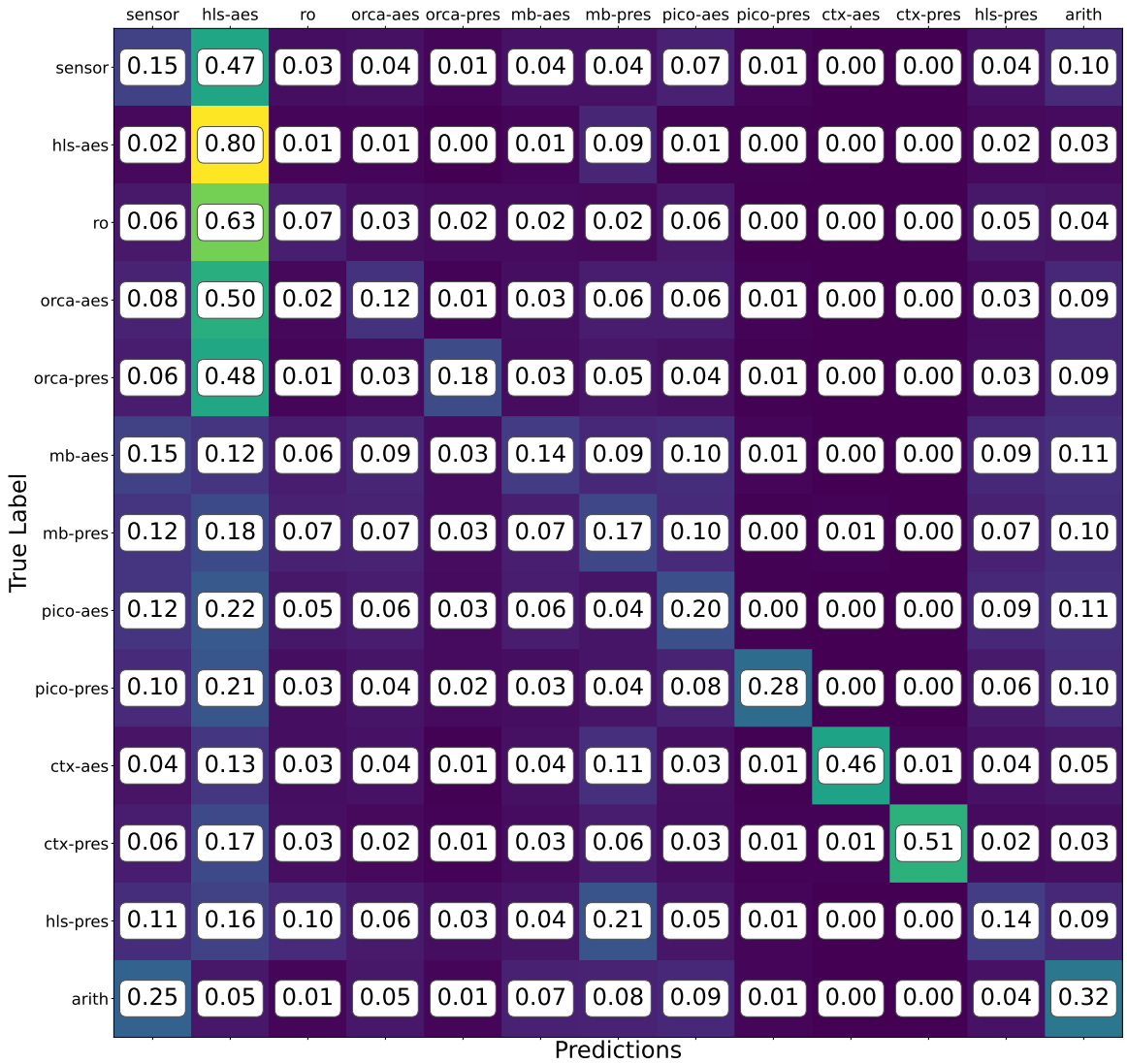
1. Left (θ, ϕ) as their default states
2. Showed that computations could be classified — clunky neural network classifying STFTs images
3. No cross-board generality
4. Poor classification within architectural class



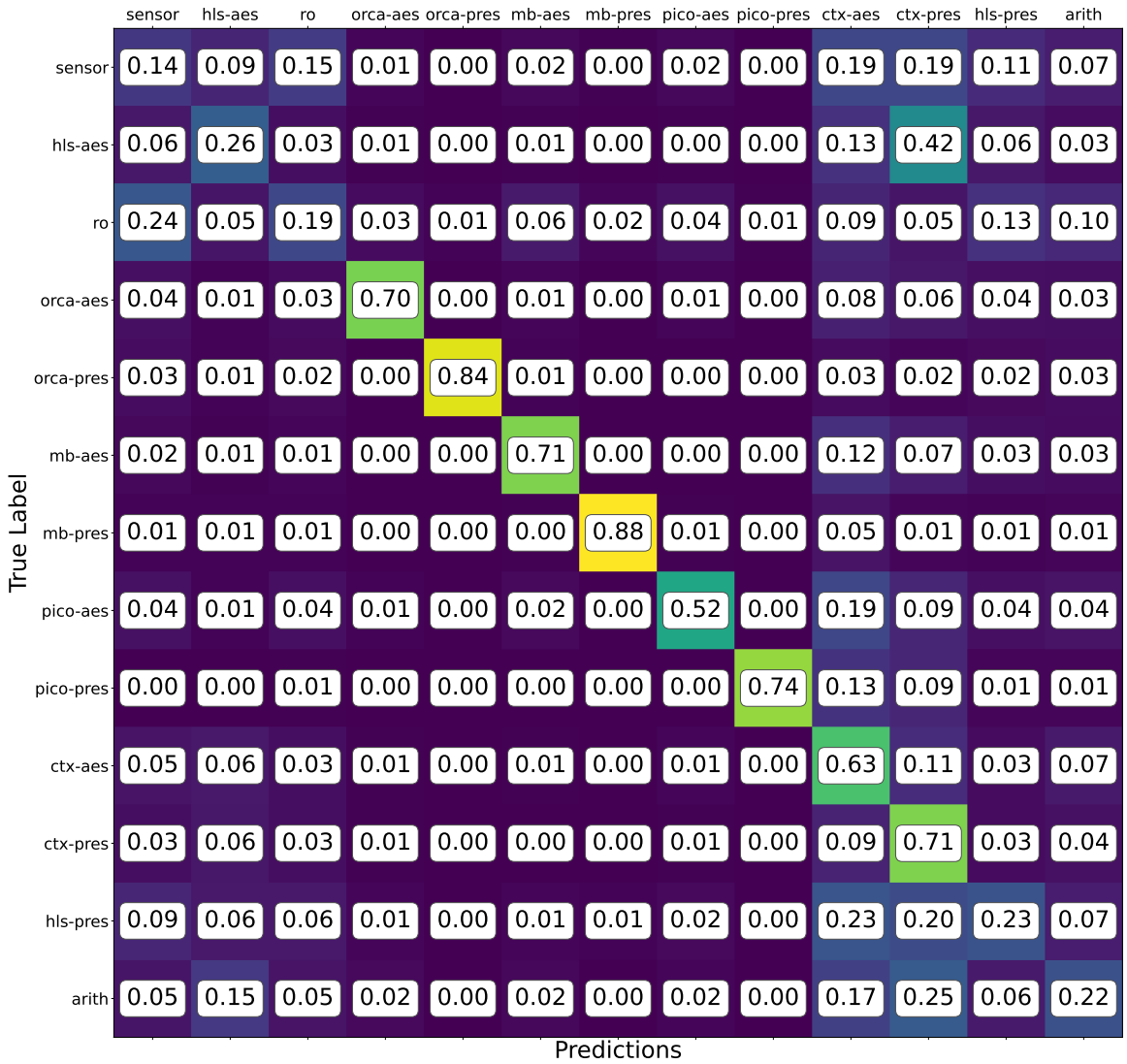
Resulting Accuracy

Tuning	Accuracy (%)	Loss
$(\uparrow \theta_{min}, \phi_{min})$	32.146	2.159
$(\downarrow \theta_{max}, \phi_{min})$	51.160	1.644
$(\downarrow \theta_{max}, \phi_{max})$	75.788	0.834
$(\downarrow \theta_{max}, \phi_{back})$	75.552	0.733
$(\uparrow \theta_{max}, \phi_{back})$	80.268	0.626

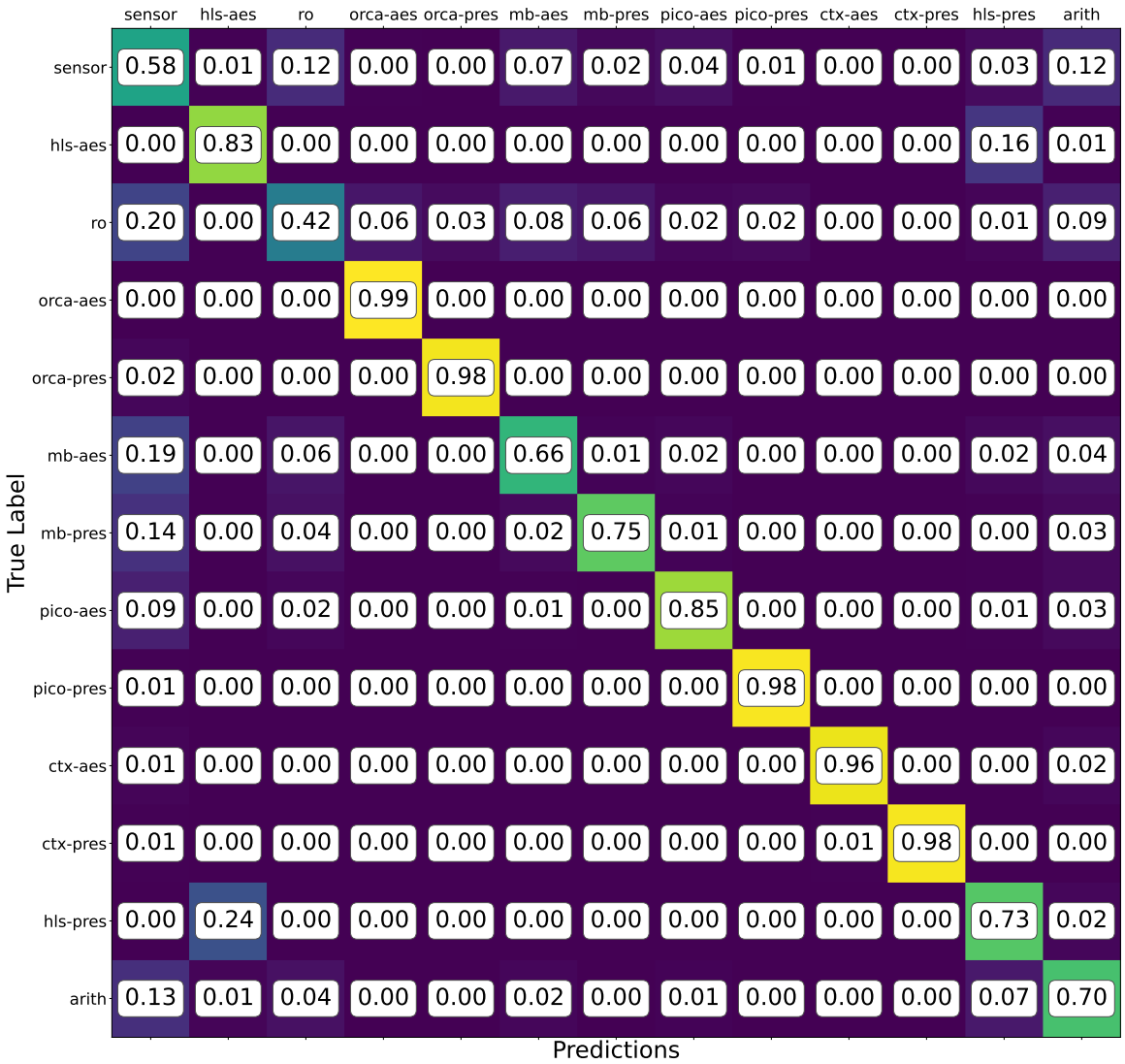
Resulting Accuracy ($\uparrow \theta_{min}, \phi_{min}$)



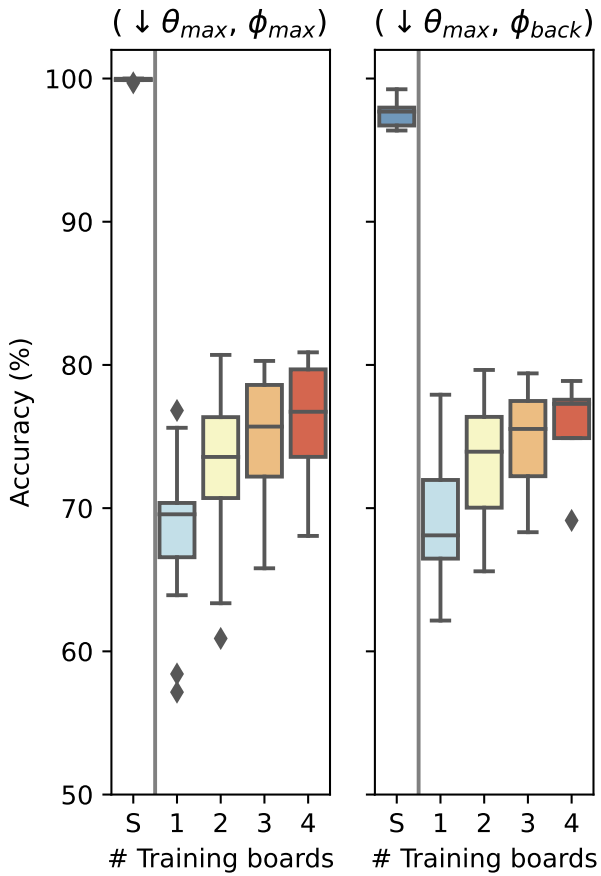
Resulting Accuracy ($\downarrow \theta_{max}$, ϕ_{min})

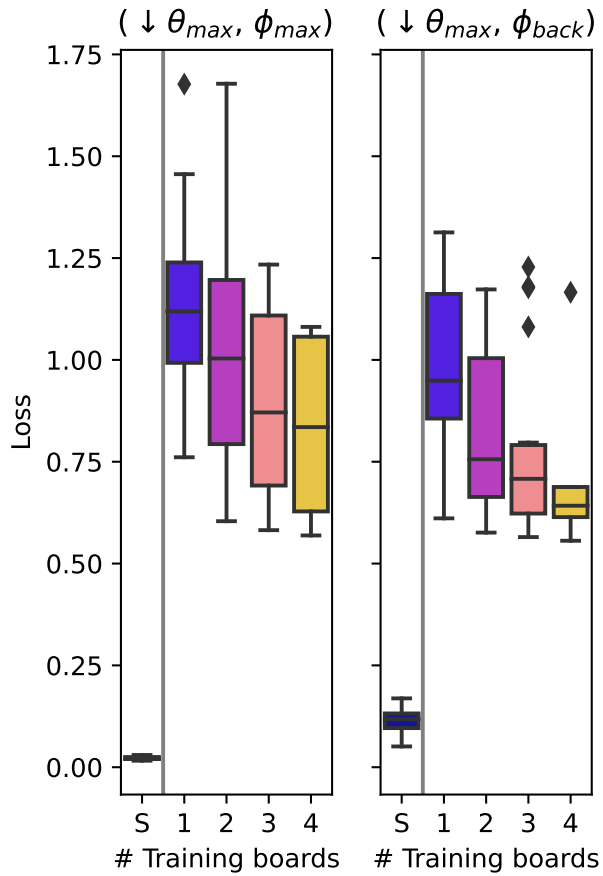


Resulting Accuracy ($\downarrow \theta_{max}$, ϕ_{max})



Background Subtraction *Does* Matter





Conclusions

1. The phase between the sensor's sampling and the clock (ϕ) of a co-tenant is essential in extracting side-channel information
2. Configuring the duration a transition is allowed to propagate through the sensor (θ) is important for avoiding architectural irregularities
3. Background subtraction is a useful tool for isolating co-tenant information in a noisy power distribution network

So what else do these sensors do?

1. Measure delay through elements, and changes in that delay
2. Launch Clock $\rightarrow$ Component $\rightarrow$ Carry Chain
3. Component's delay can change....and depends on its previous state
4. Can recover this previous state based on delay

Bonus: Voltage Fluctuation Sensor