There’s Always a Bigger Fish:

A Clarifying Analysis of a Machine-Learning-Assisted Side-Channel Attack

Jack Cook, **Jules Drean**, Jonathan Behrens, Mengjia Yan

ML-Assisted Side-Channel Attacks

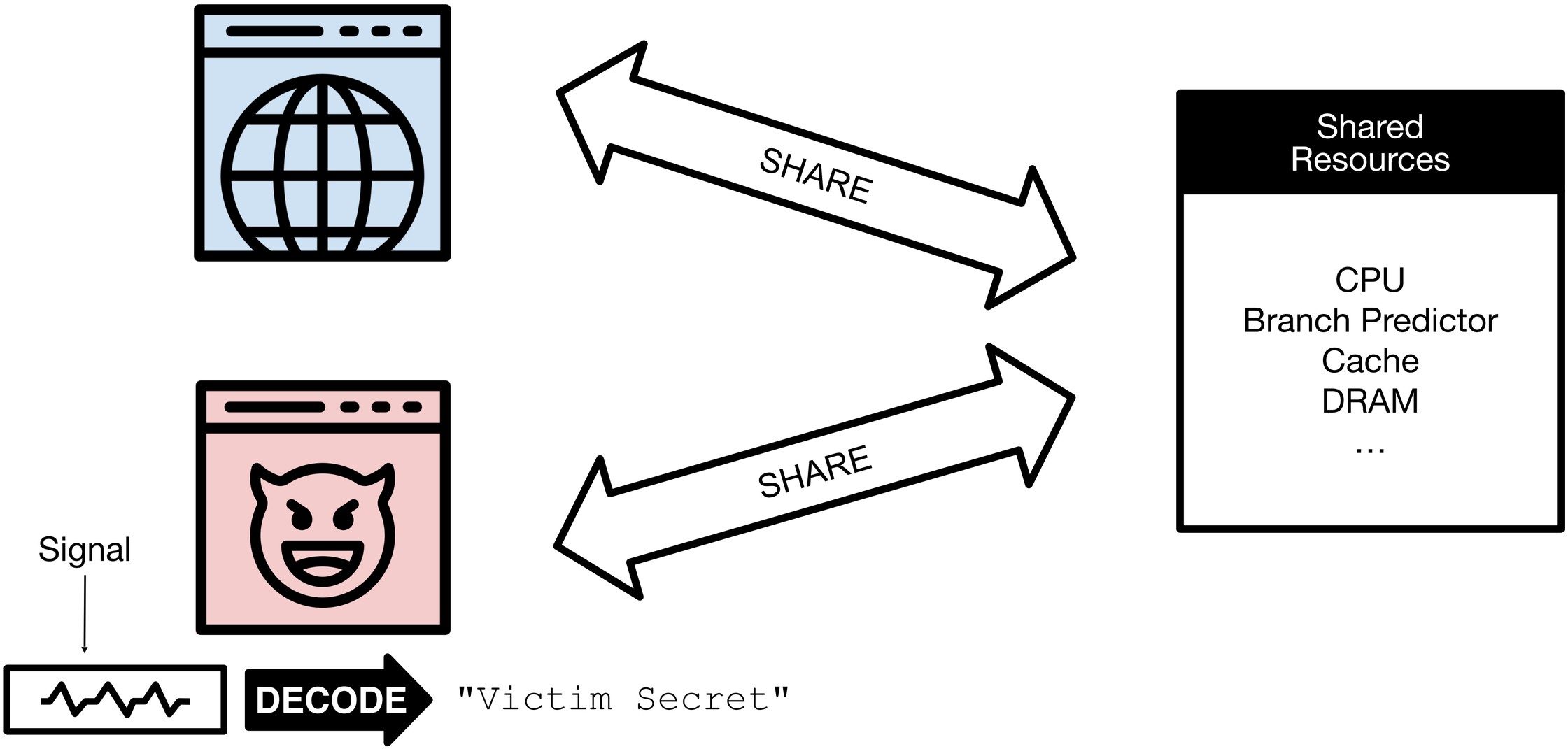
- Are highly effective and even work with noise
- Work as a black box and **are hard to interpret**

Bigger Fish is a detailed analysis of a misunderstood side-channel attack

Agenda

1. Background
2. A Surprising Experiment
3. In Depth Security Analysis
4. Findings & Conclusion

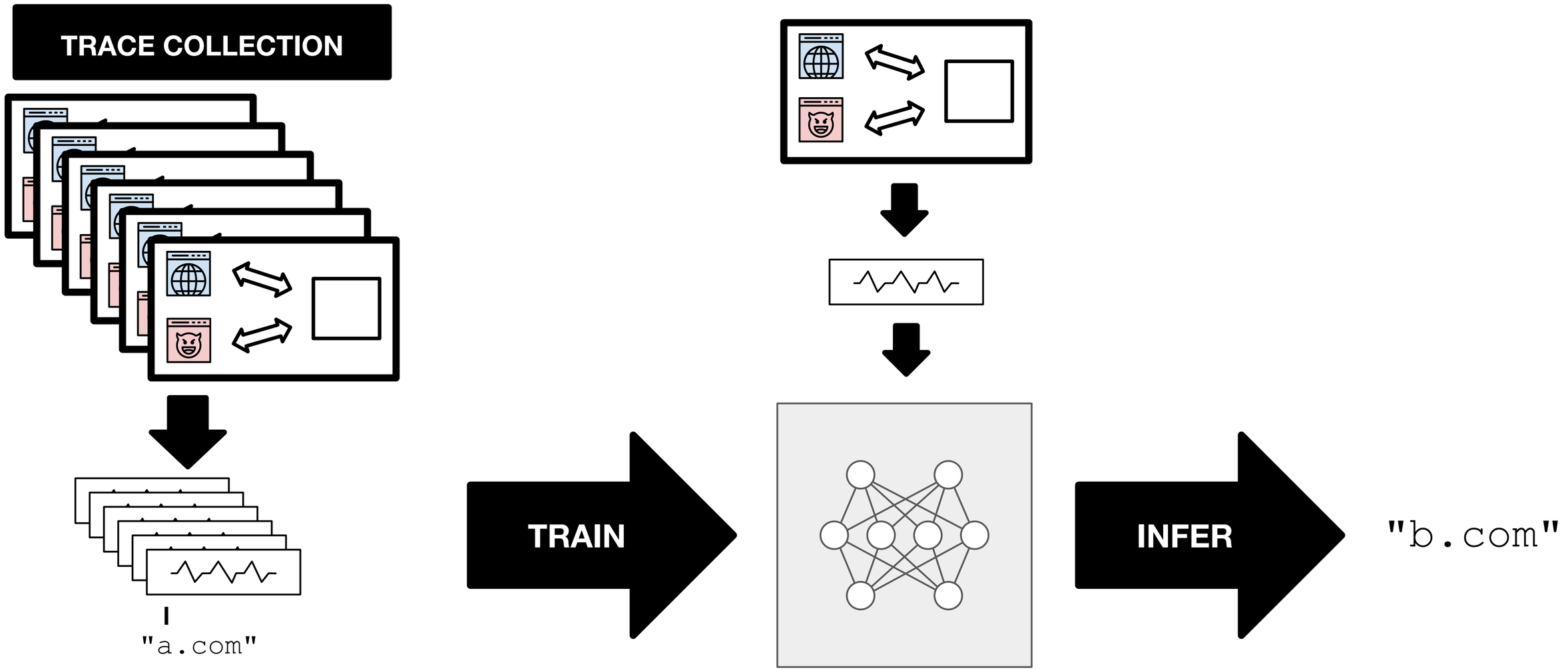
Timing Side Channels



Website Fingerprinting Attacks

- Very serious privacy implications
- Can be mounted from JavaScript
- Good benchmark for side channels

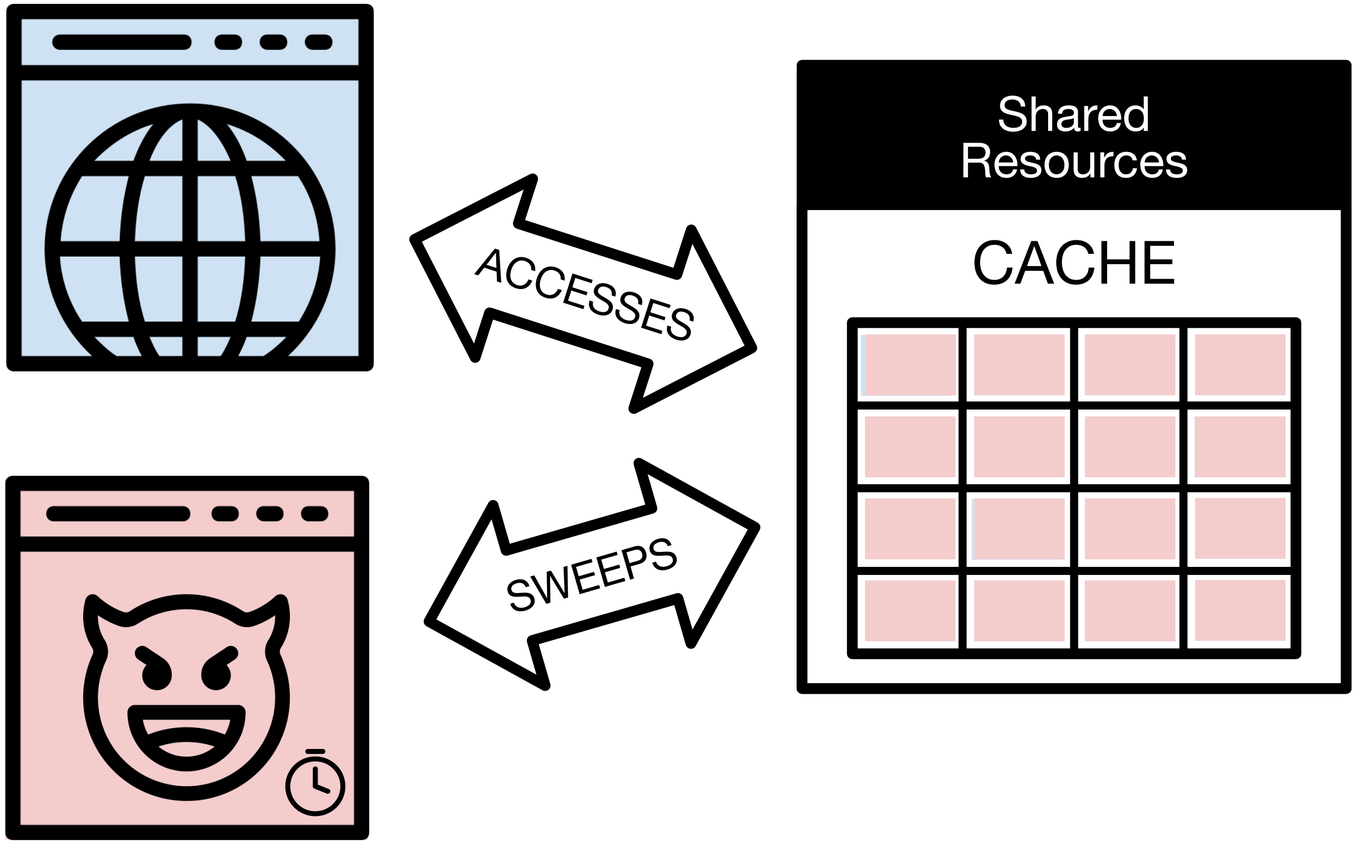
Website Fingerprinting: Machine-Learning Classifier



A Cache-Occupancy Attack*

ATTACKER'S CODE

```
loop {
  start = time()
  counter = 0;
  while(time() - start < 5ms) {
    counter++;
    SWEEP_CACHE();
  }
  Trace[start] = counter;
}
```



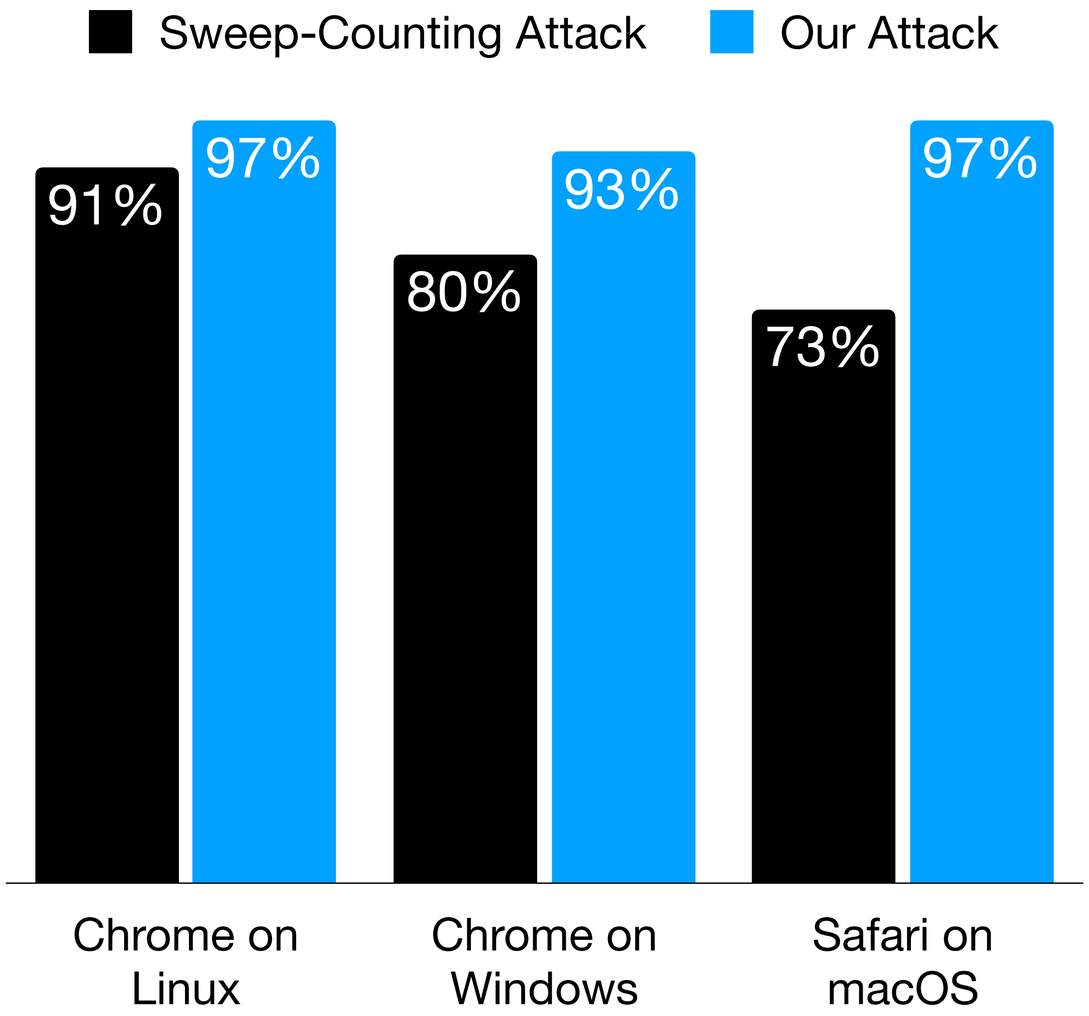
* Shusterman, et al. "Prime+Probe 1,JavaScript 0: Overcoming Browser-based Side-Channel Defenses." *30th USENIX Security Symposium (USENIX Security 21)*. 2021.

A Surprising Experiment

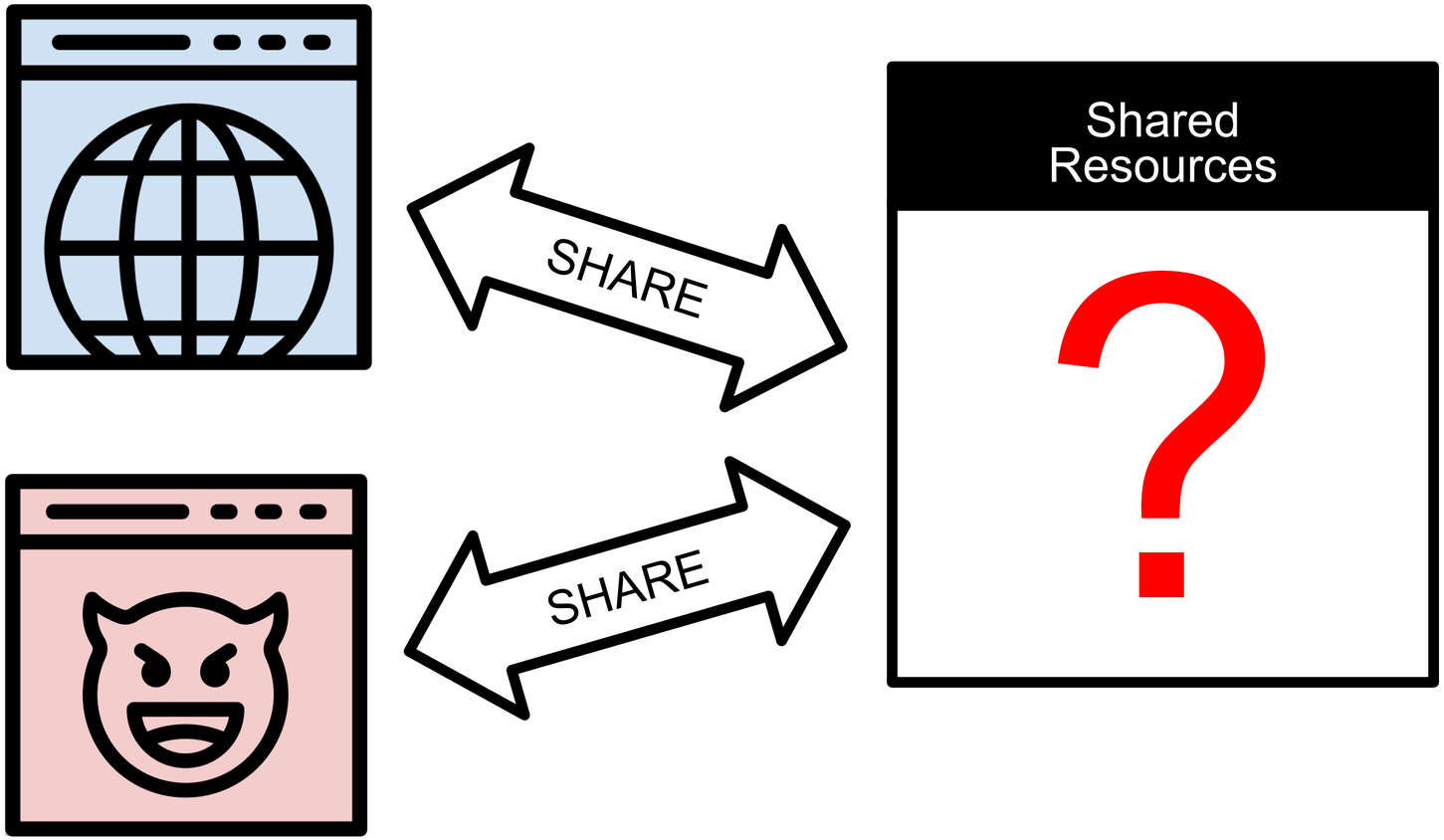
ATTACKER'S CODE

```
loop {
  start = time()
  counter = 0;
  while(time() - start < 5ms) {
    counter++;
  }
  Trace[start] = counter;
}
```

REMOVE MEMORY ACCESSES



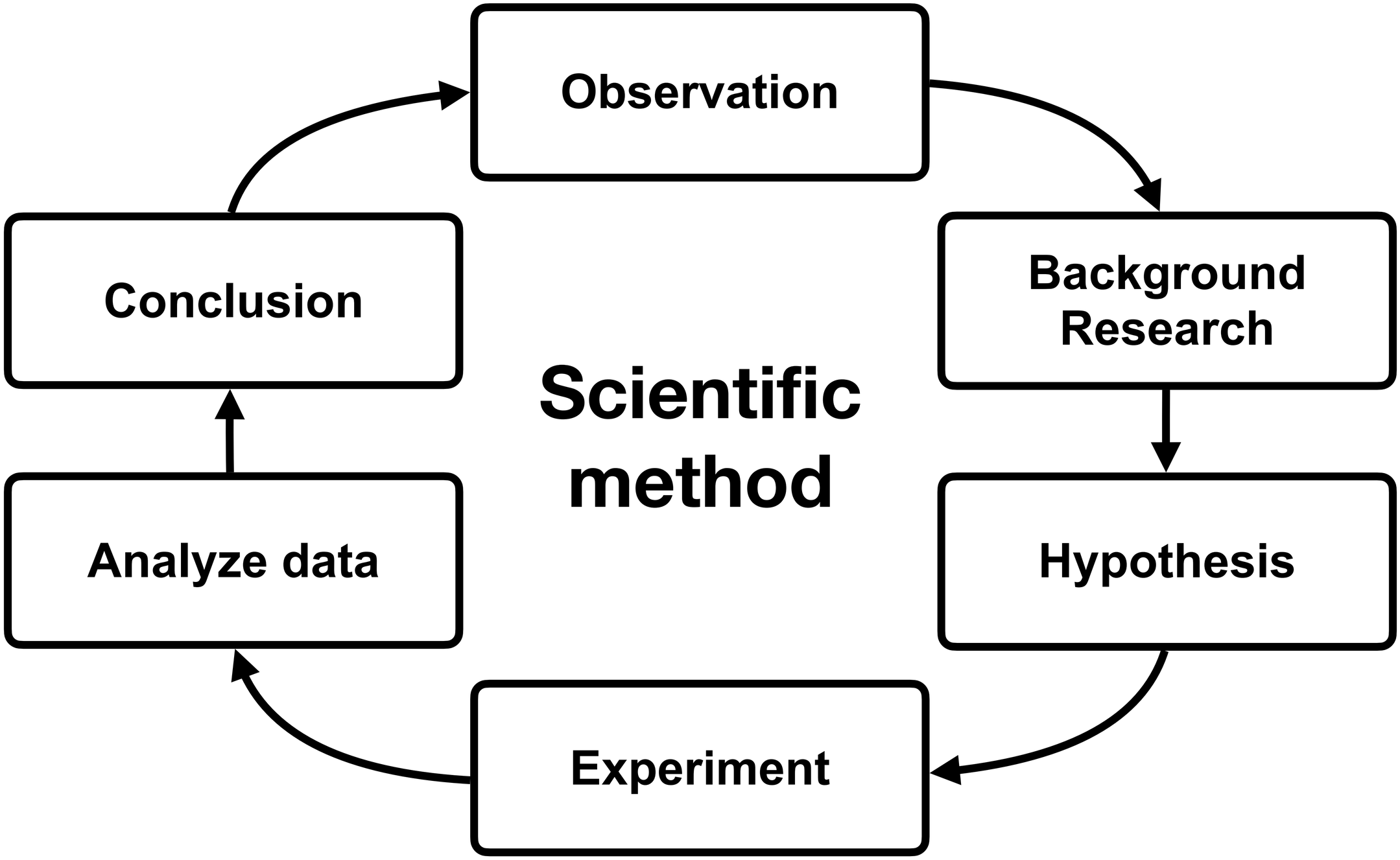
What is the primary side channel?

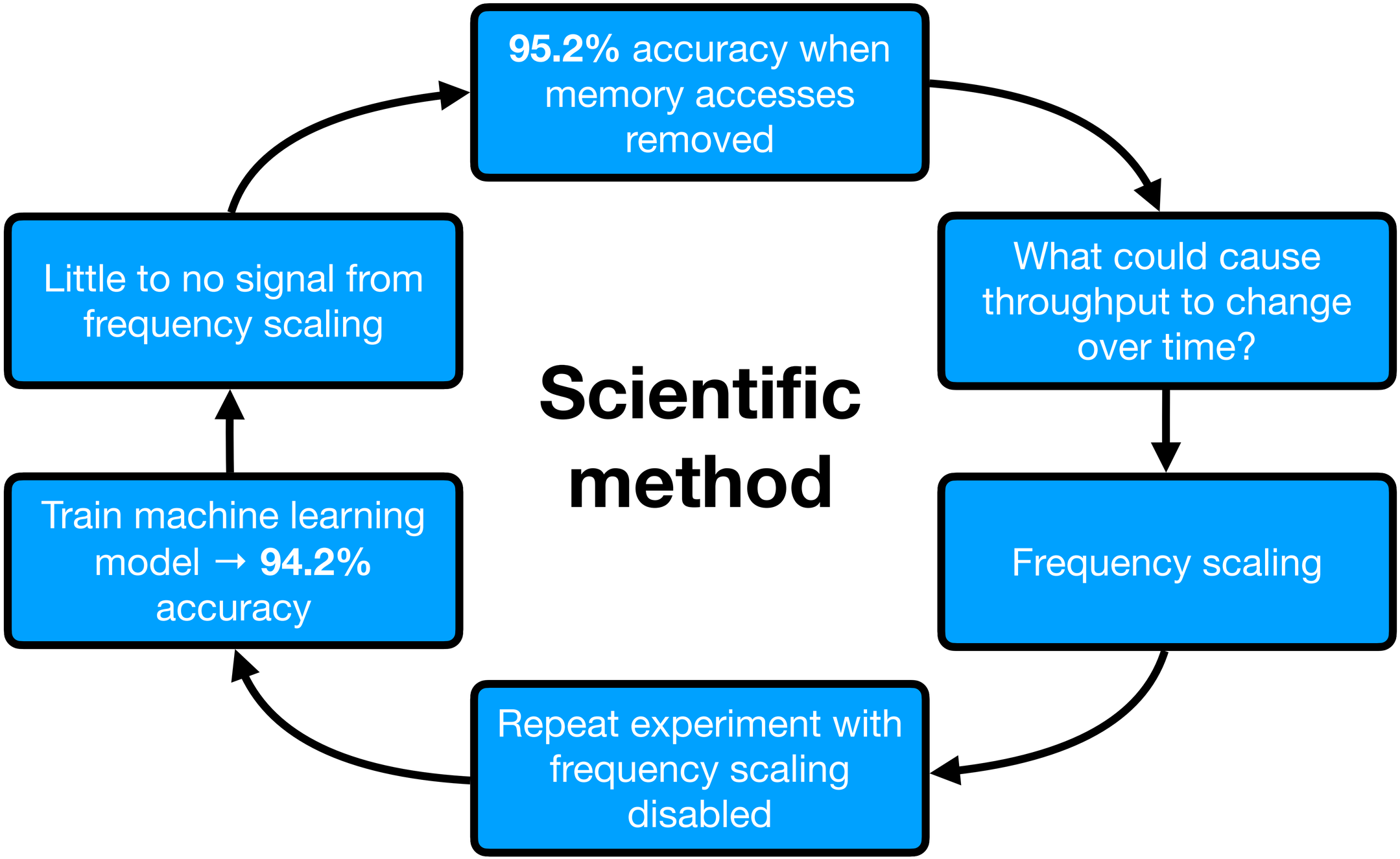


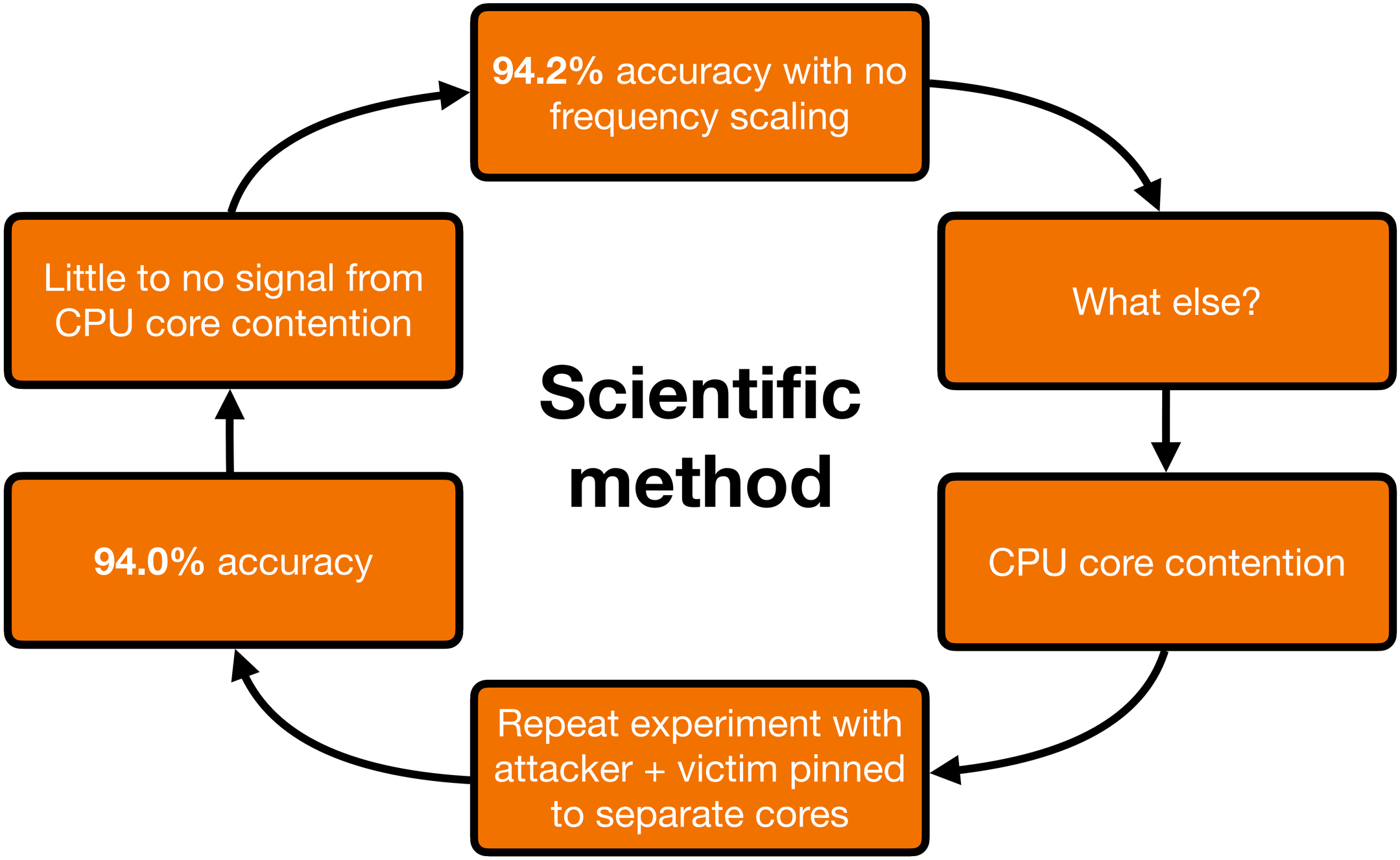
ML-Assisted Side-Channel Attacks

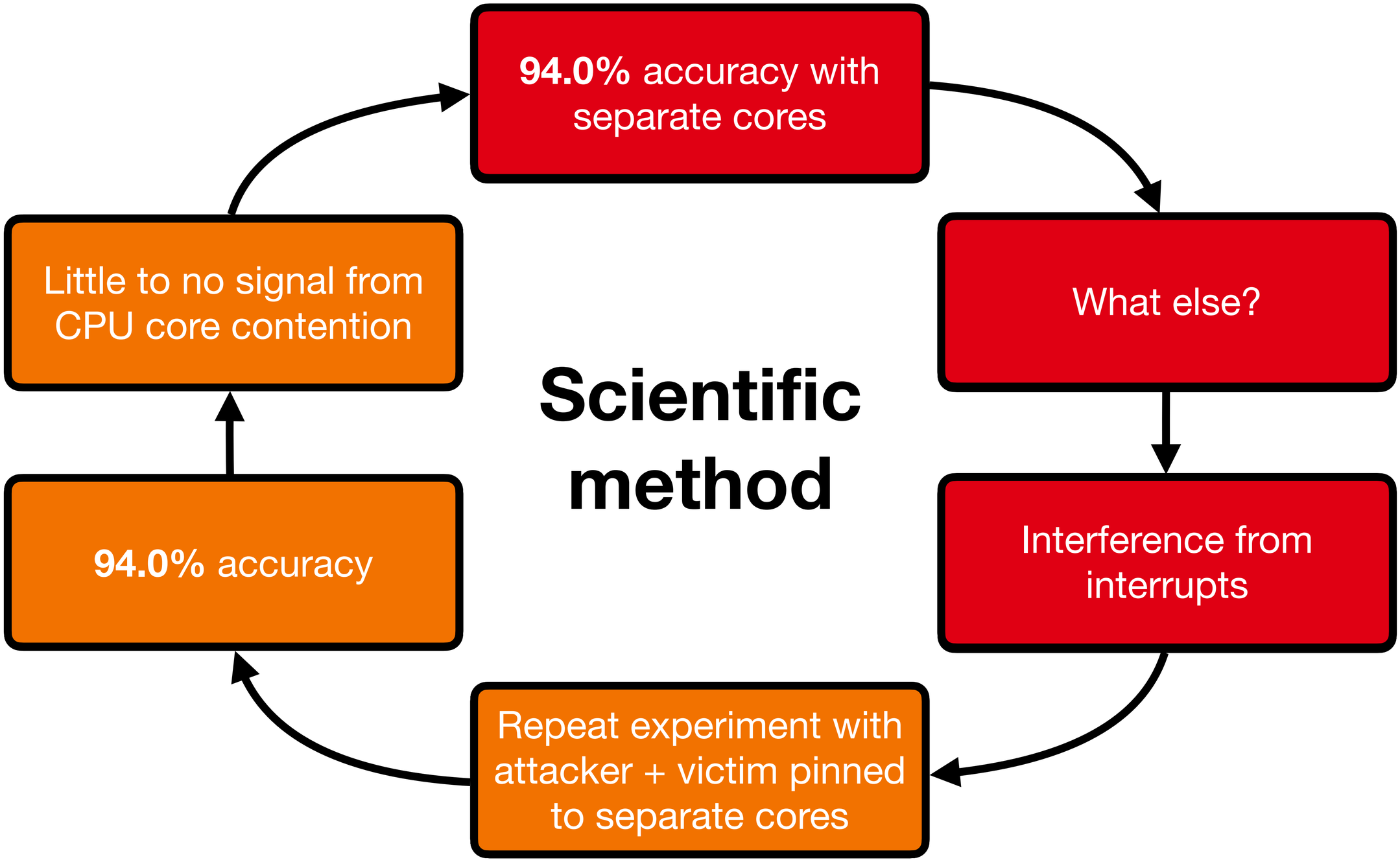
- Work as a black box and **are hard to interpret**

Bigger Fish is a detailed analysis of a misunderstood side-channel attack


Observation
Background Research
Hypothesis
Experiment
Analyze data
Conclusion
Scientific method

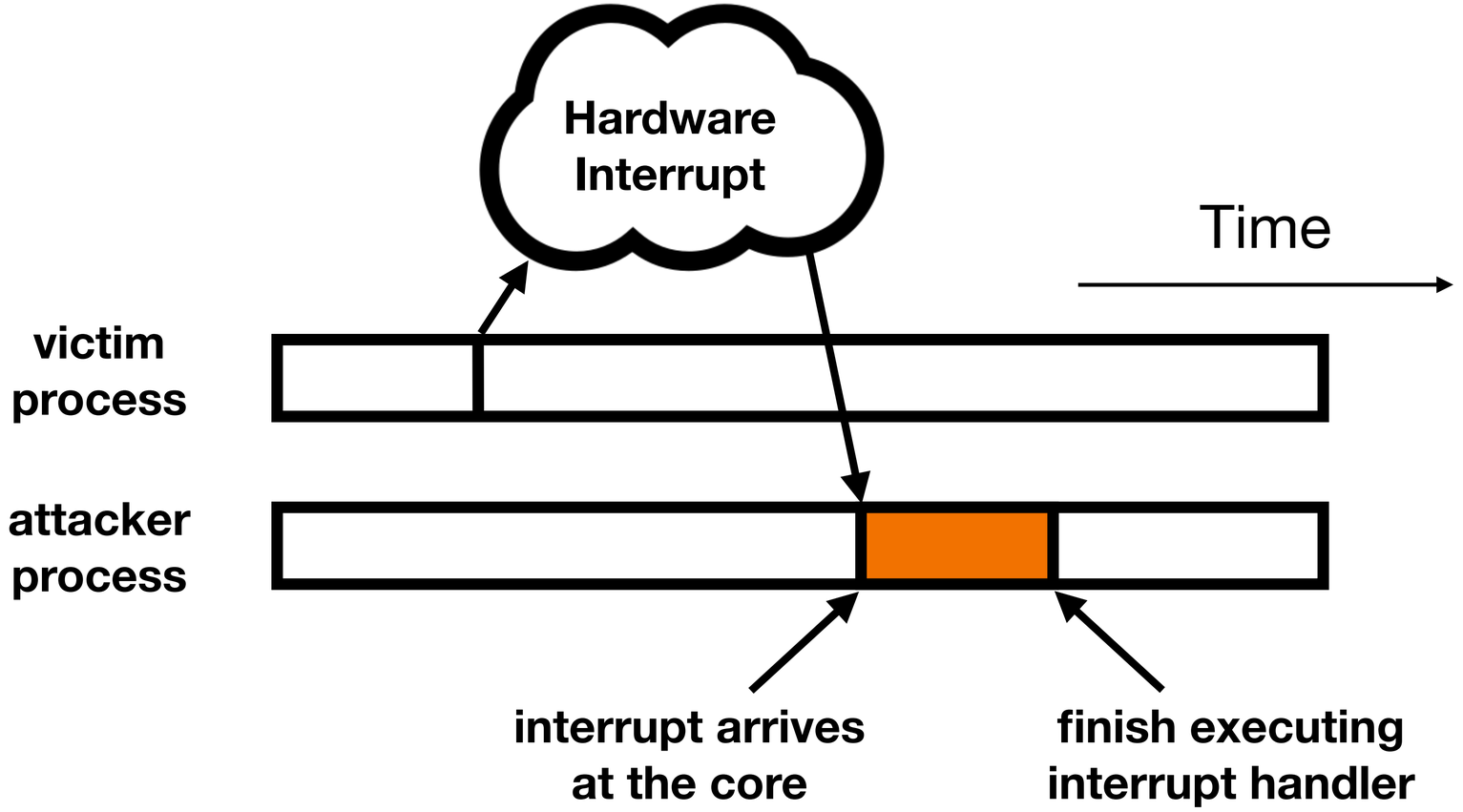

95.2% accuracy when memory accesses removed
What could cause throughput to change over time?
Frequency scaling
Repeat experiment with frequency scaling disabled
Train machine learning model → 94.2% accuracy
Little to no signal from frequency scaling
Scientific method

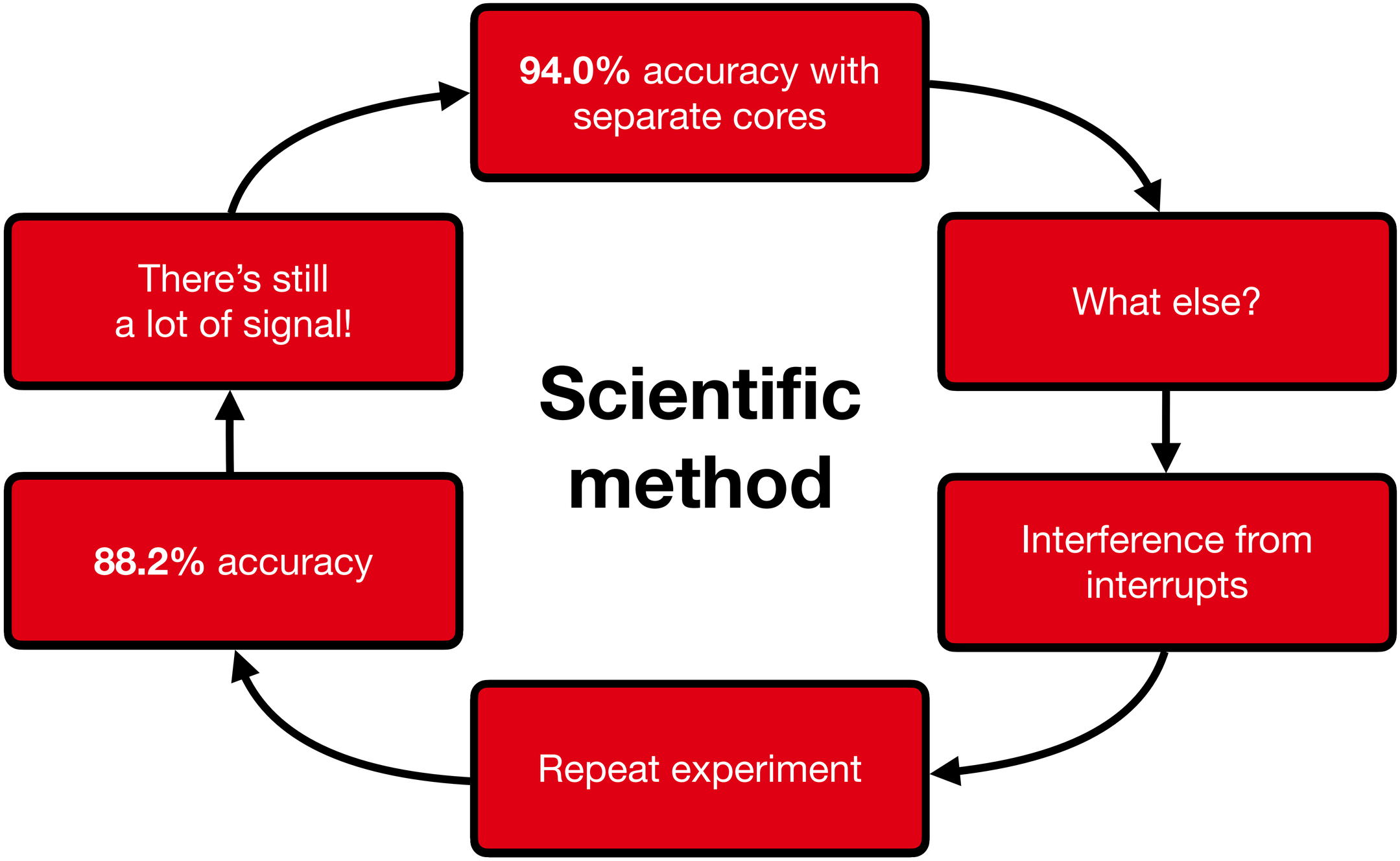

94.2% accuracy with no frequency scaling
What else?
CPU core contention
Repeat experiment with attacker + victim pinned to separate cores
94.0% accuracy
Little to no signal from CPU core contention
Scientific method


94.0% accuracy with separate cores
What else?
Interference from interrupts
Repeat experiment with attacker + victim pinned to separate cores
94.0% accuracy
Little to no signal from CPU core contention
Scientific method

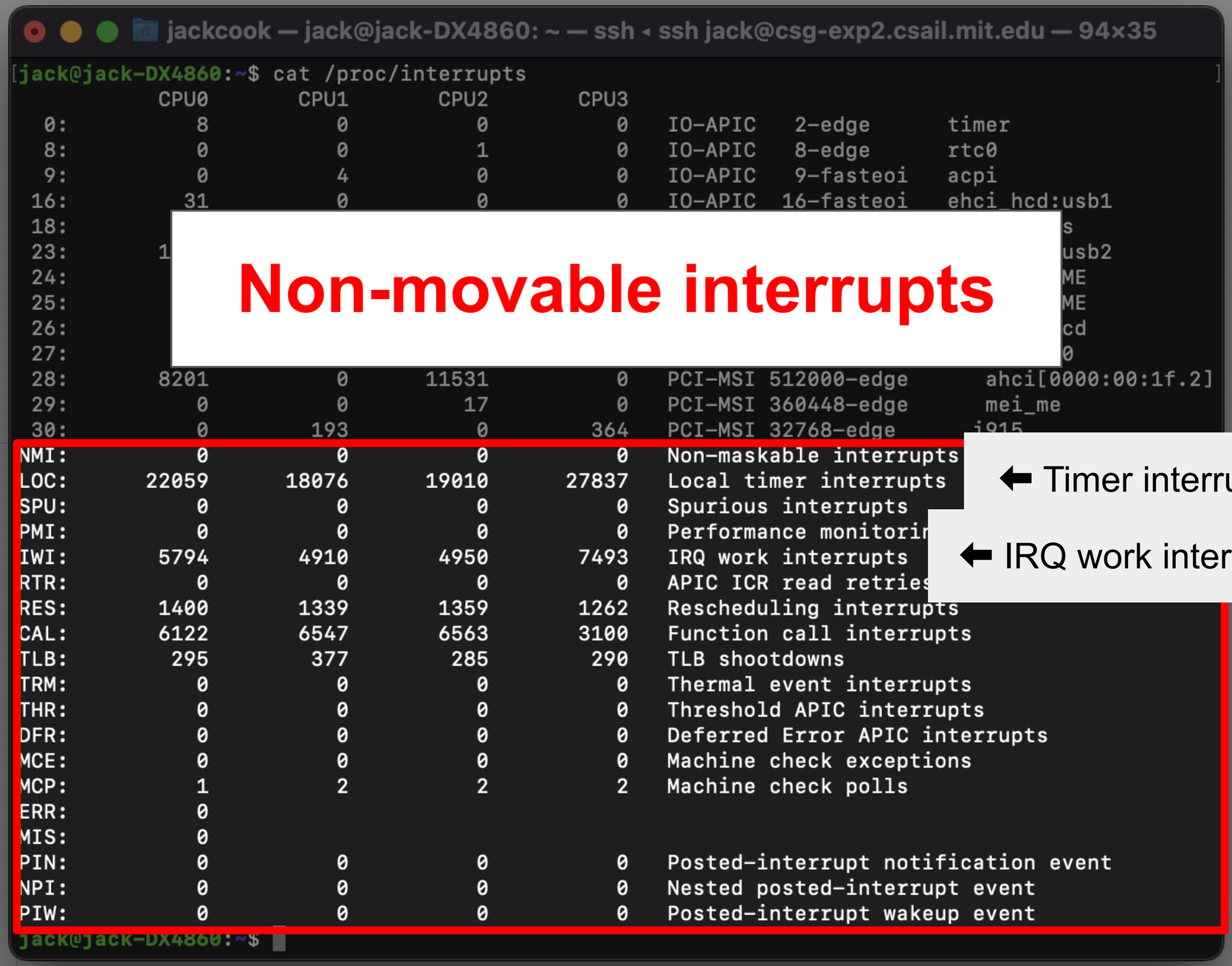
System Interrupts

- Used to deal with asynchronous events
  - e.g. Graphics interrupts render content on a display
- Some can be “pinned” to specific cores, some can’t




94.0% accuracy with separate cores
What else?
Interference from interrupts
Repeat experiment
88.2% accuracy
There's still a lot of signal!
Scientific method


jackcook — jack@jack-DX4860: ~ — ssh ◂ ssh jack@csg-exp2.csail.mit.edu — 94×35
jack@jack-DX4860:~$ cat /proc/interrupts
CPU0 CPU1 CPU2 CPU3
0: 8 0 0 0 IO-APIC 2-edge timer
8: 0 0 1 0 IO-APIC 8-edge rtc0
9: 0 4 0 0 IO-APIC 9-fasteoi acpi
16: 31 0 0 0 IO-APIC 16-fasteoi ehci_hcd:usb1
18: 0 8 0 0 IO-APIC 18-fasteoi i801_smbus
23: 1943 934 0 0 IO-APIC 23-fasteoi ehci_hcd:usb2
24: 0 0 0 0 PCI-MSI 458752-edge PCIe PME
25: 0 0 0 0 PCI-MSI 468992-edge PCIe PME
26: 0 0 0 0 PCI-MSI 524288-edge xhci_hcd
27: 0 376 0 10880 PCI-MSI 1048576-edge enp2s0
28: 8201 0 11531 0 PCI-MSI 512000-edge ahci[0000:00:
29: 0 0 17 0 PCI-MSI 360448-edge mei_me
30: 0 193 0 364 PCI-MSI 32768-edge i915
NMI: 0 0 0 0 Non-maskable interrupts
LOC: 22059 18076 19010 27837 Local timer interrupts
SPU: 0 0 0 0 Spurious interrupts
PMI: 0
IWI: 5794
RTR: 0
RES: 1400
CAL: 6122
TLB: 295
TRM: 0 0 0 0 Thermal event interrupts
THR: 0 0 0 0 Threshold APIC interrupts
DFR: 0 0 0 0 Deferred Error APIC interrupts
MCE: 0 0 0 0 Machine check exceptions
MCP: 1 2 2 2 Machine check polls
ERR: 0
MIS: 0
PIN: 0 0 0 0 Posted-interrupt notification event
NPI: 0 0 0 0 Nested posted-interrupt event
PIW: 0 0 0 0 Posted-interrupt wakeup event
jack@jack-DX4860:~$
Movable interrupts
16: Mouse
23: Keyboard
27: Network card
30: Graphics card



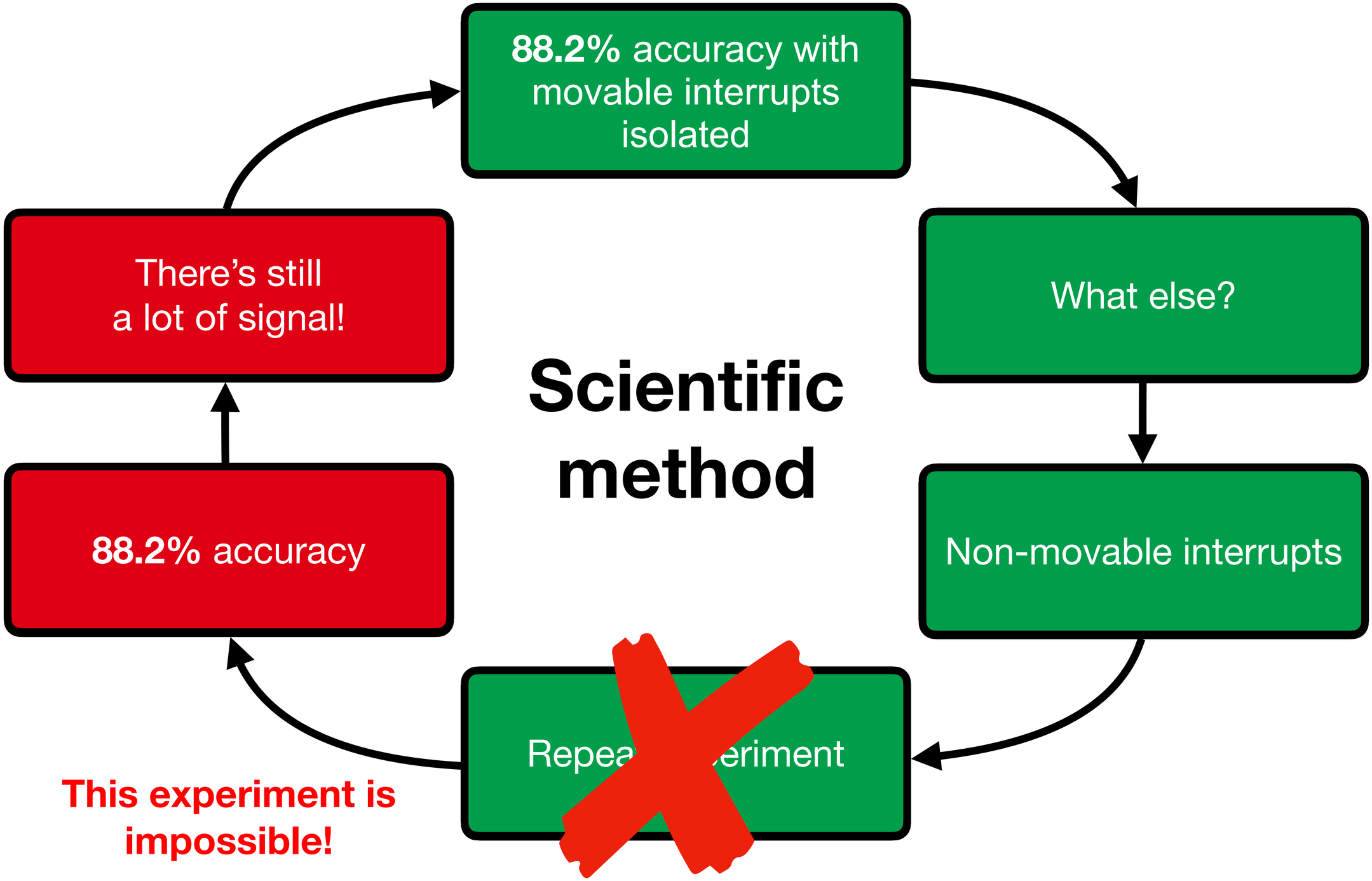
Non-movable interrupts

⬅ Timer interrupts

⬅ IRQ work interrupts

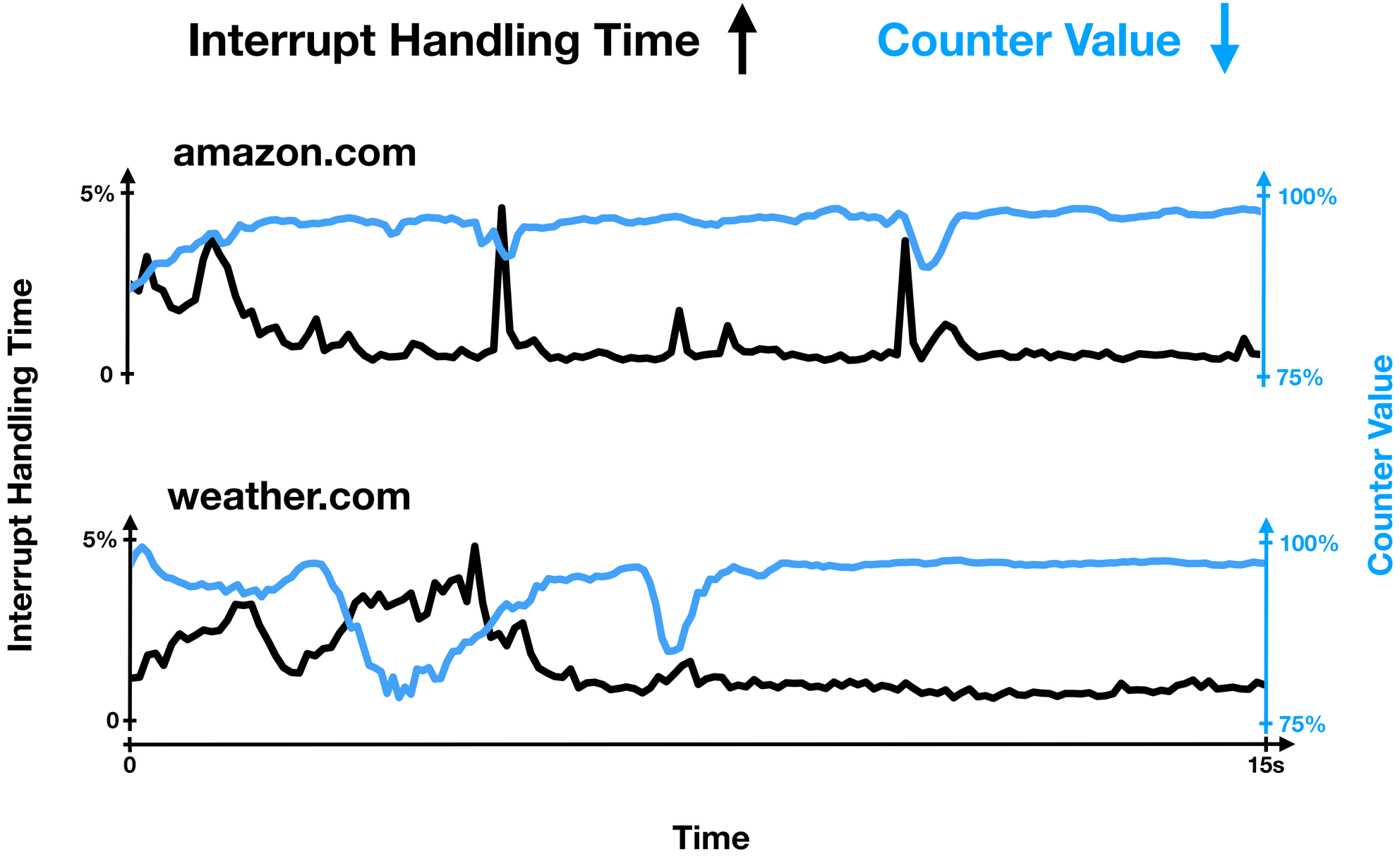
Non-Movable Interrupts

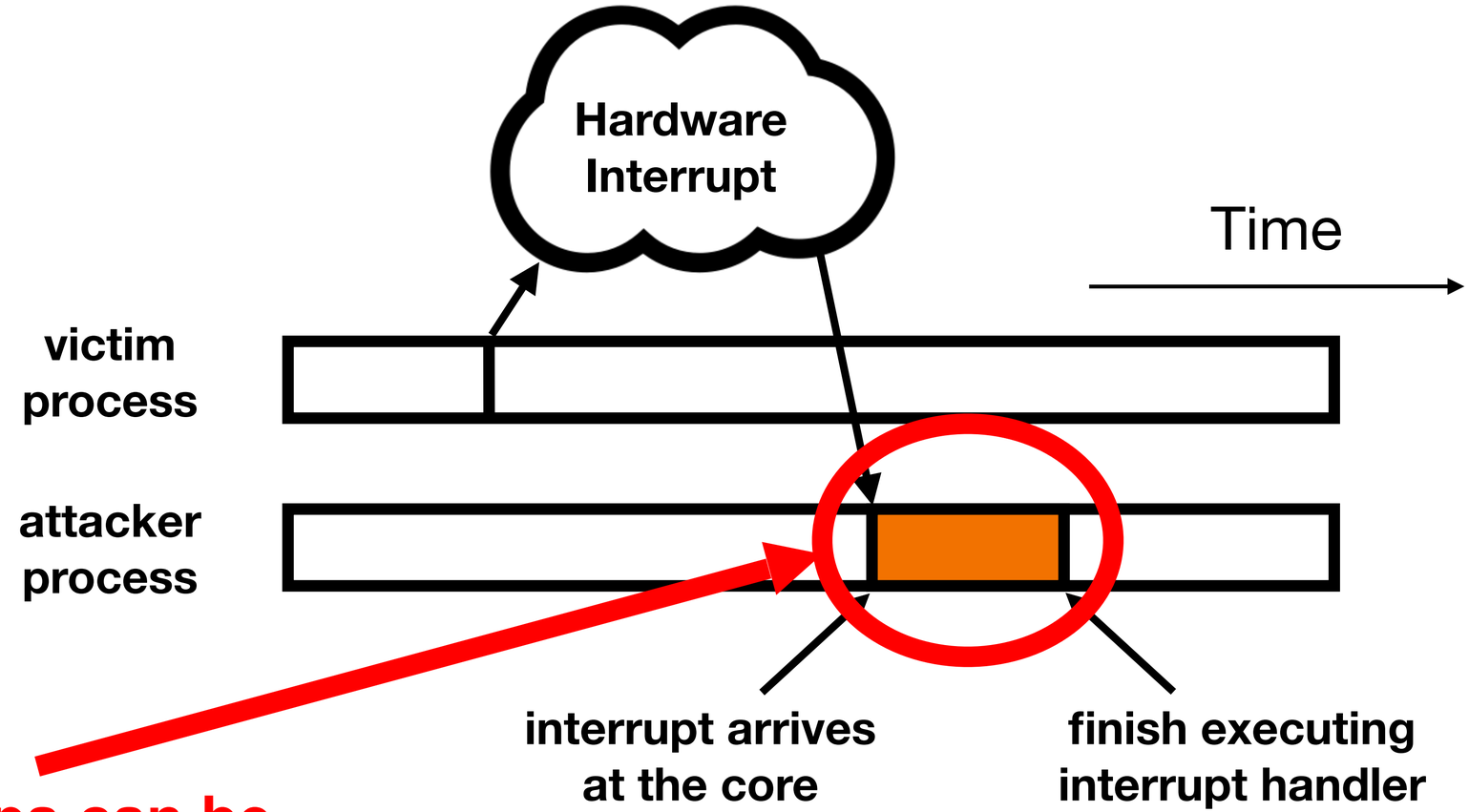
- Can’t be isolated from any cores
- Are necessary for the operating system to function
- Have not been studied in detail for side channels


88.2% accuracy with movable interrupts isolated
What else?
Non-movable interrupts
Repea eriment
88.2% accuracy
There's still a lot of signal!
Scientific method
This experiment is impossible!

eBPF

- Allows instrumentation of the Linux kernel at runtime
- We developed a tool to monitor interrupt characteristics
- Records time at beginning and end of interrupt handlers


Interrupt Handling Time
Counter Value
amazon.com
5%
0
100%
75%
weather.com
5%
0
100%
75%
0
15s
Interrupt Handling Time
Counter Value
Time



99% of gaps can be explained by the presence of interrupts

More in the paper!

- Randomized timer countermeasure
- Cache + interrupt noise experiments
- Virtual machine isolation
- Further discussion of non-movable interrupts
- Analysis of web browser timers
- And more!

Collect trace | Download traces

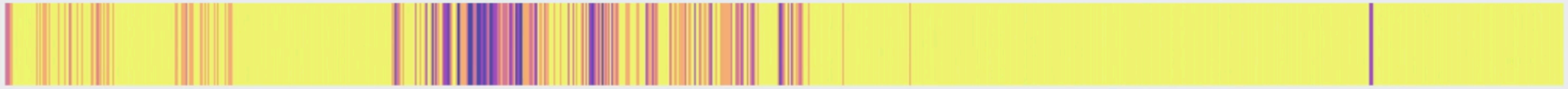
#1: amazon.com, 5 seconds

#2: amazon.com, 5 seconds

#3: amazon.com, 5 seconds

#4: weather.com, 5 seconds

#5: weather.com, 5 seconds



#6: weather.com, 5 seconds

Key Takeaways

Findings and Conclusion

- Machine-learning-assisted attacks are powerful but hard to interpret
- Sweep-counting “cache-occupancy” attack* primarily exploits system interrupts
- Non-movable interrupts have strong security implications
- We release our analysis toolset at https://github.com/jackcook/bigger-fish

* Shusterman, et al. "Prime+Probe 1,JavaScript 0: Overcoming Browser-based Side-Channel Defenses." *30th USENIX Security Symposium (USENIX Security 21)*. 2021.

Demo

jackcook.github.io/
bigger-fish