GateBleed

Artifacts Evaluated — Functional

Results Reproduced

Artifacts Available

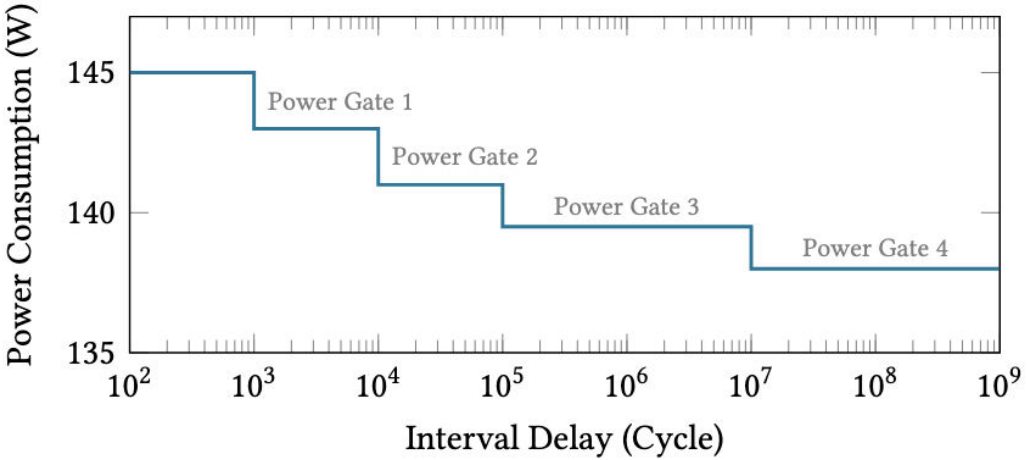
A Timing-Only Membership Inference Attack, MoE-Routing Inference, and a Stealthy, Generic Magnifier Via Hardware Power Gating in AI Accelerators

Joshua Kalyanapu, Farshad Dizani, Darsh Asher, Azam Ghanbari, Rosario Cammarota, Aydin Aysu, Samira M. Ajorpaz

NC STATE UNIVERSITY intel

AI Applications are Power Intensive

Optimizations like power gating deployed in AI accelerators to reduce power consumption



But, security and privacy risks of power optimizations on AI applications and remote settings are poorly understood.

Prior Hardware Side Channel Attacks

Leak **stored data**:

- Spectre
- Meltdown
- Cache Telepathy
- MDS

GateBleed is the first side channel to leak data **privacy** through power optimizations via timing

AI Privacy Inference Attacks

Infer information not stored on the target machine as bytes

training-set membership or expert routing choice.

AI Should not be Trained on Private or Copyrighted Data

1. Getty Images and Stability AI face off in British copyright trial that will test AI industry

Getty Images is facing off against artificial intelligence company Stability AI in a London courtroom for the first

BBC

Home News Sport Business Innovation Culture Arts Travel Earth Audio Video Live

2. LinkedIn accused of using private messages to train AI

23 January 2025

Share Save

3. Apple sued by authors over use of books in AI training

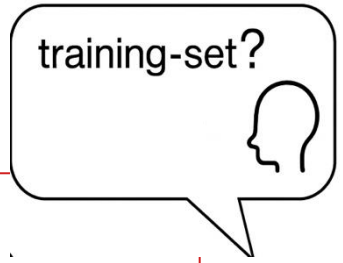
By Mike Scarcella

September 5, 2025 10:11 PM GMT · Updated September 5, 2025

Aa

Background

Membership Inference Attack (MIA)



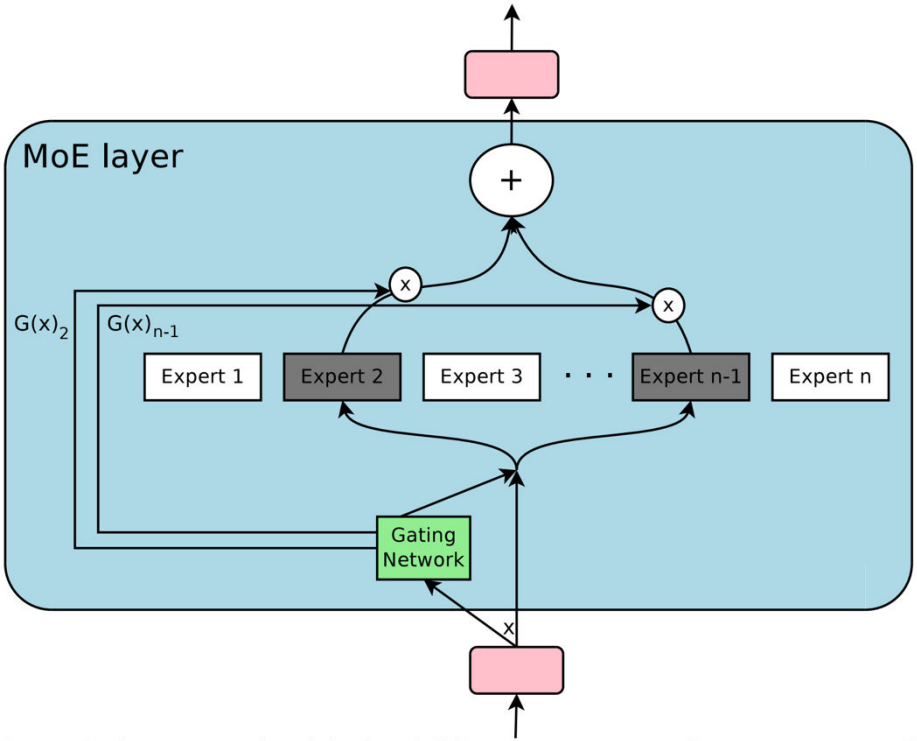
Determine whether specific data was used to
an AI model

Less effective when only labels are exposed

Best MIAs rely on access to confidence score

GateBleed bypasses MIA defenses via power optimization in Intel AMX

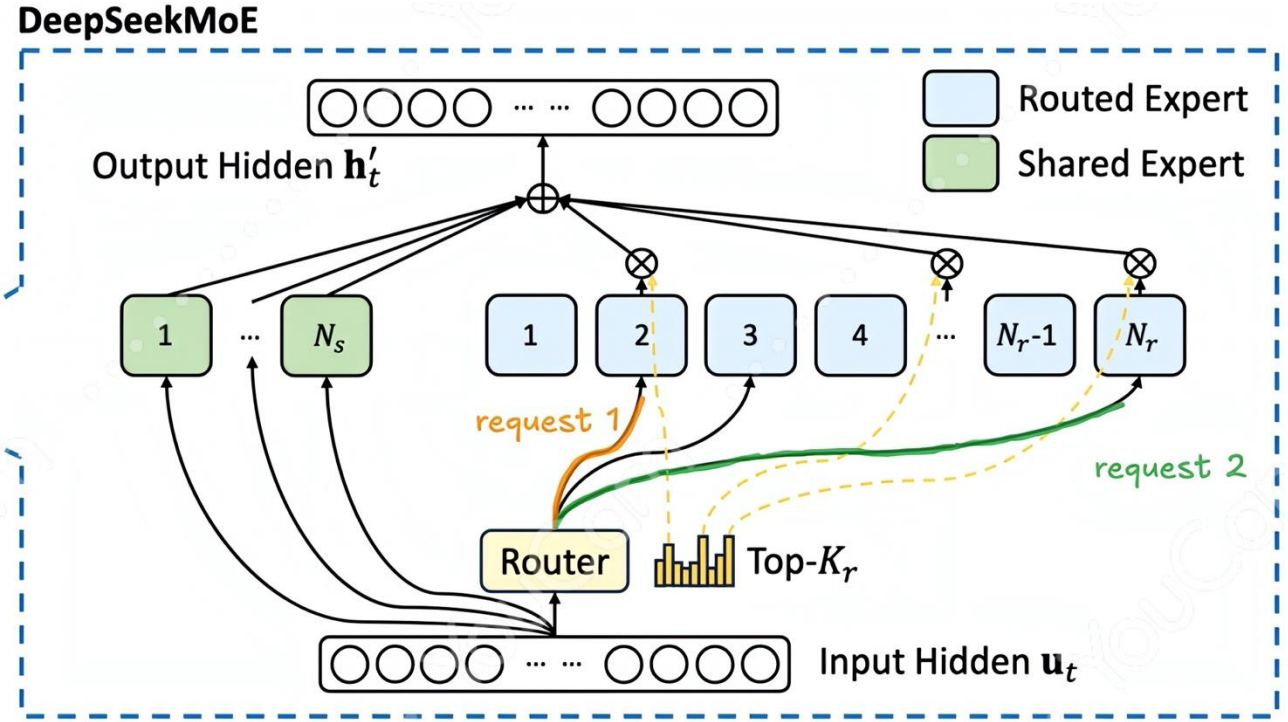
Adaptive NNs decrease runtime and power consumption of ML inference via shortcuts



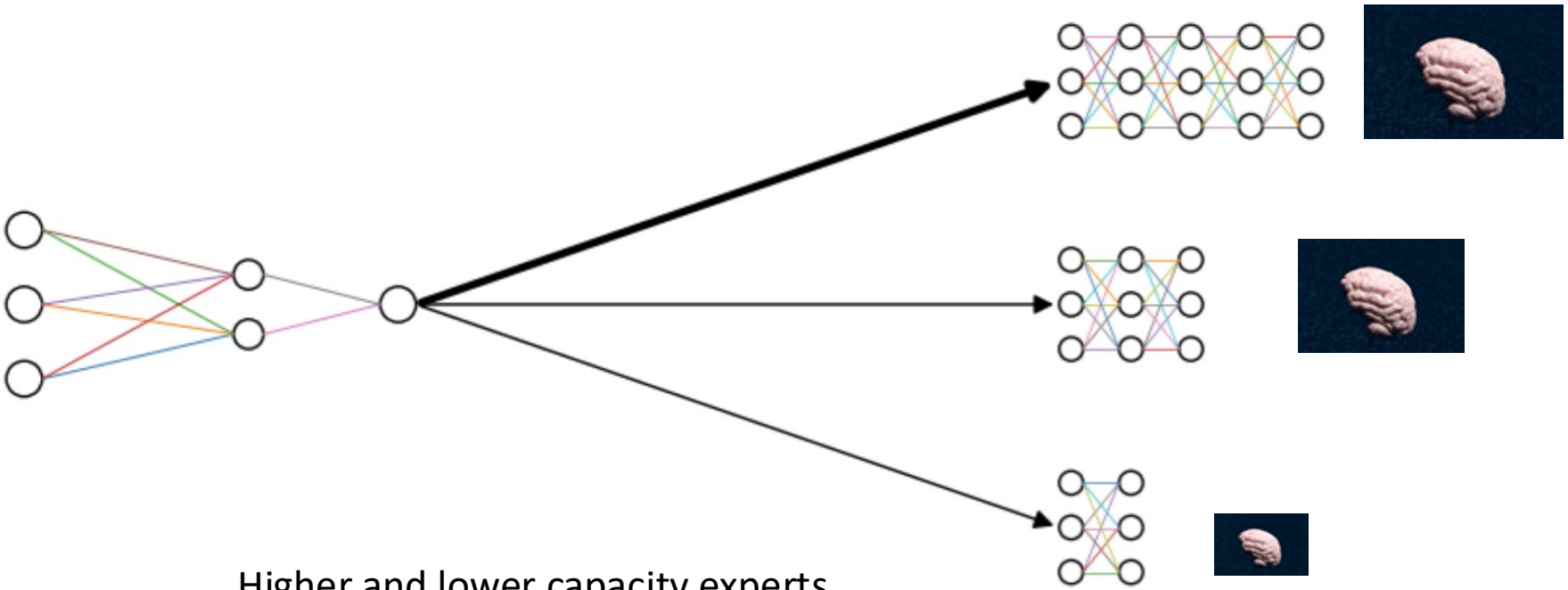
Sparse Mixture of Experts (MoE) selects two experts to perform computations.

MoE is shipping in production — from recommendation engines to generative AI

DeepSeek-MoE, 671 billion parameters, does not need a monster GPU!



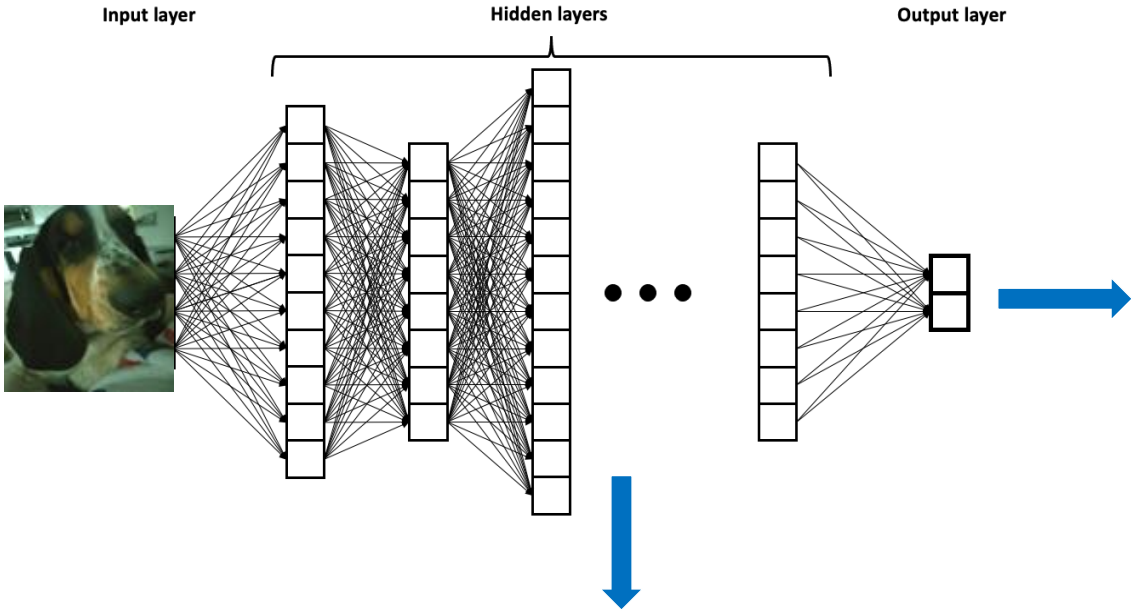
Heterogeneous Mixture of Experts (HMoE)



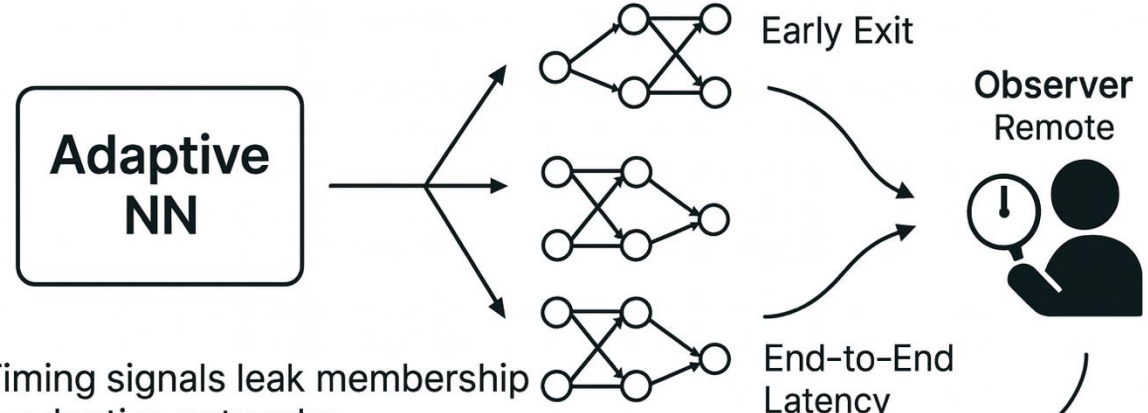
Higher and lower capacity experts
More or less experts selected

Exit Early if model is confident in its decision

- BranchyNet
- MSDNet
- SkipNet



BUT Adaptive Neural Networks introduce new vulnerabilities (Akinsanya et al., 2024)



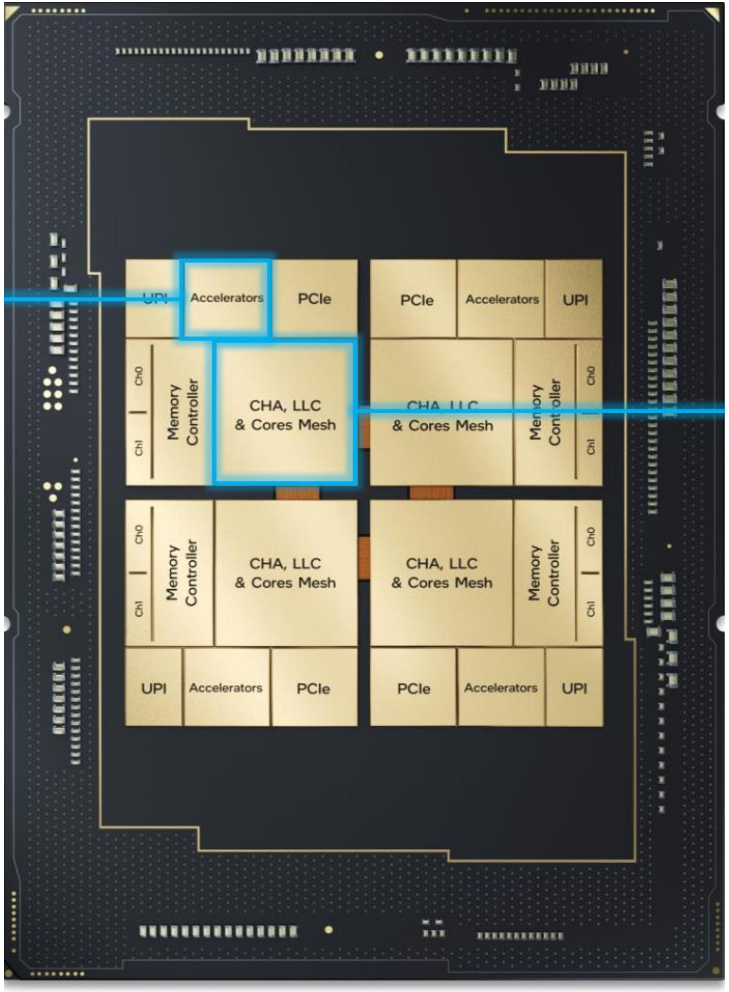
- Timing signals leak membership in adaptive networks:
  - Routing choice
  - Early exit
- Faster inference on training set members

Mitigation – pad shorter execution with dummy instructions to equalize timing

GateBleed bypasses defense through power gating!

We Reverse Engineer Intel Advanced Matrix Extensions

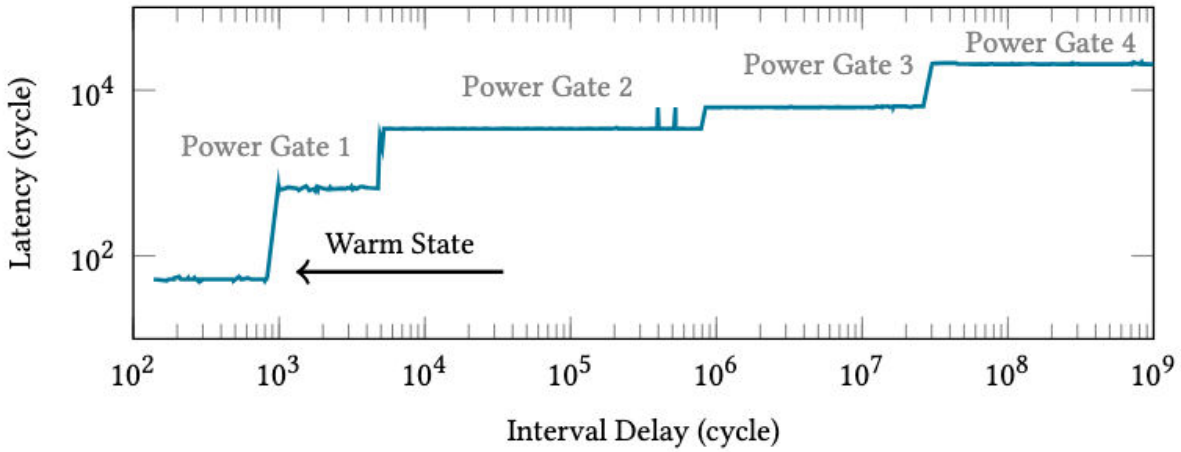
- 1024 multiply-accumulates (MACs) in 10 cycles **per core**
- No memory offload penalty
- **BUT…**significant power requirements!



Source: Intel

Power Gating in Intel AMX

Power Gating is the powering off of unused microarchitectural components

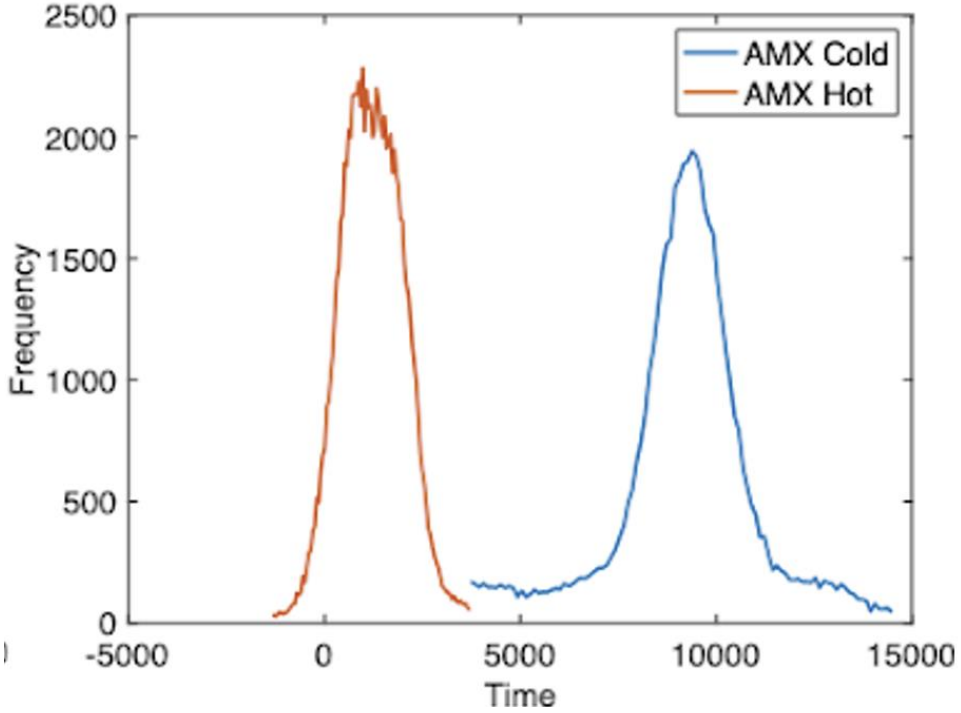


Reduces power consumption but incurs **an extra latency** when waking up – up to 20,000 cycles!!

GateBleed exploits this extra latency

AMX Will be Slower If the Unit was Idle.

If AMX invocation conditional on an AI secret, the timing of the code becomes correlated with the secret.



Threat Model & Gadgets

ML Attacks Threat Model

Leakage Target:

Training set membership (MIA)

Expert selection (Expert inference attack)

Attacker Capability:

Local, co-resident, unprivileged

Access to use the deployed MLaaS model

No knowledge of training data is required

Defense: Dummy (non AMX) instruction added. No diff in end-to-end timing **Note**: This leakage happens only with power-gating optimized AI accelerator.

Generic Magnifier Threat Model

Target

Ability to distinguish < 5 us timing with a 5 us timer

Victim

Web browser visiting attacker-controlled website

Attacker Capability:

Serve custom Javascript

Serve Javascript that runs AMX

Remote Arbitrary Address Leakage Threat Model

Leakage Target:

Value at arbitrary address

Victim

Network endpoint with Spectre gadget

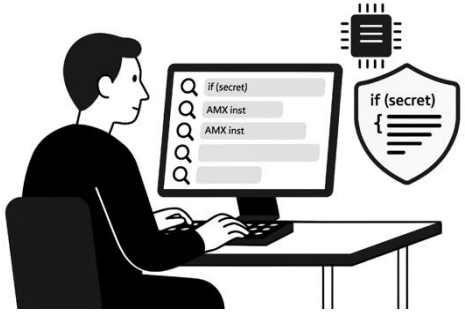
Branch predicate depends on packet contents

Code contains AMX instructions

Attacker Capability:

Query network endpoint

Searched ML Libraries for:



ML secret data/parameter links to AMX reuse.

Any such conditional-AMX code path can exhibit GateBleed leakage.

▸ GateBleed Enables a Broader Family of Accelerator-Diven Privacy Leaks

Three Gatebleed gadget categories:

- Input-Dependent Routing Gadgets
- Confidence and Early-Exit Gadgets
- Session, Configuration, and Static Context

Input-Dependent Routing Implementation

AdaptiveLogSoftmax:

- Cluster membership/target label → Cluster-based compute → timing reveals true class.

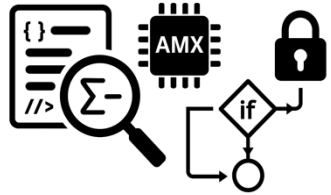
Mixtral (HF):

- Expert routing index/logits → AMX matmul only for selected experts → Query + Time reveals routing.

TensorFlow MoE:

- Router activation threshold → AMX only for active experts → timing reveals routing threshold.

Conditional expert activation in MoE models leaks internal routing decisions through AMX power gating.



Confidence and Early-Exit Implementations

BranchyNet:

•Exit-stage decision/confidence → AMX pattern reveals model confidence.

MSDNet:

•Early-exit threshold/entropy → prediction entropy modulates AMX use.

SkipNet / BlockDrop:

• Layer skipping mask → conditional AMX reuse; latency encodes execution path.

HF Agent:

•Action logits → decoder invoked only for tool tokens; timing reveals action type.

AutoGen:

Planner state → AMX usage conditional on planner loop.

Confidence-driven exits or planner loops leak decision confidence and internal control flow through AMX.

Session, Config, and Agentic AI Context Gadgets

LLaMA KV Cache:

- KV reuse vs recompute → query + timing reveals prompt reuse/history. ***If reuse AMX won't run (cold)***

ONNX Runtime KV Cache:

- Session reuse → warm session reduces AMX setup time. ***If reuse the AMX won't run (cold)***

llama.cpp Quant Dispatch:

- Quantization type → timing distinguishes int8 vs fp32. ***AMX will operate on int8, can't operate on fp32***

GoogLeNet:

- Training flag → auxiliary classifier only in training → latency reveals mode.

Generic CNN:

- Layer type toggle → conv vs MLP alters AMX pattern → timing leaks architecture.

OpenAI Agentic Function API:

- Completion signal → AMX only for function tools → latency leaks endpoint behavior.

Methodology

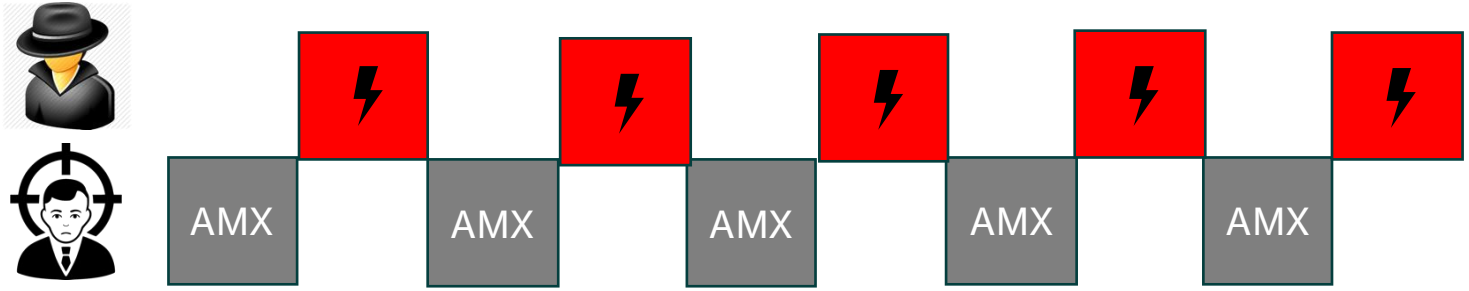
Methodology: HMoE Expert Inference Attack

Heterogeneous Mixture-of-Experts mirror expert asymmetry or sparsity

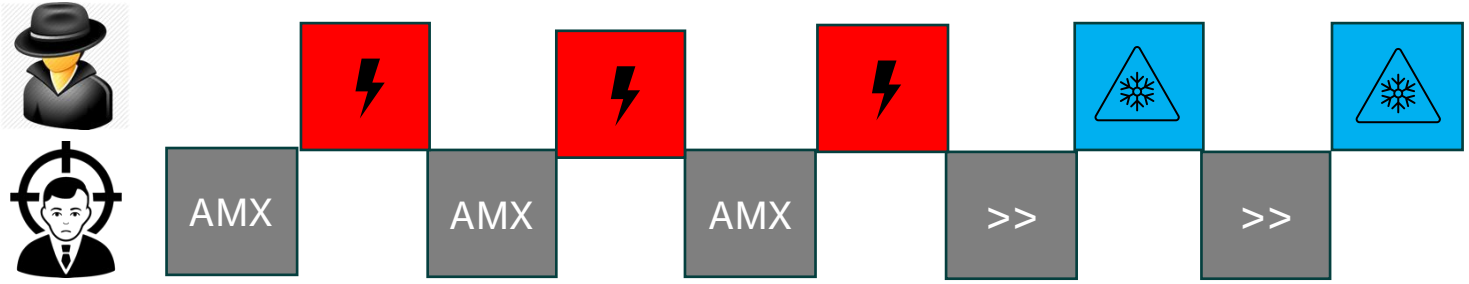
Training set: 784 English sentences
Test set size: 300 English sentences
.

Parameter	Big Expert	Little Expert
Hidden Size	256	256
Intermediate size	256	256
Number of heads	4	4
Embedding size	256	256
Number of layers	24	**10-22 (varied)**

HMoE Expert Inference Attack



Larger capacity expert routed



Smaller capacity expert routed

Methodology: Early-Exit Transformer Membership Inference Attack

- 24 Transformer layers
- Early-exit at layer 12
- Identical training and test set to HMoE transformer

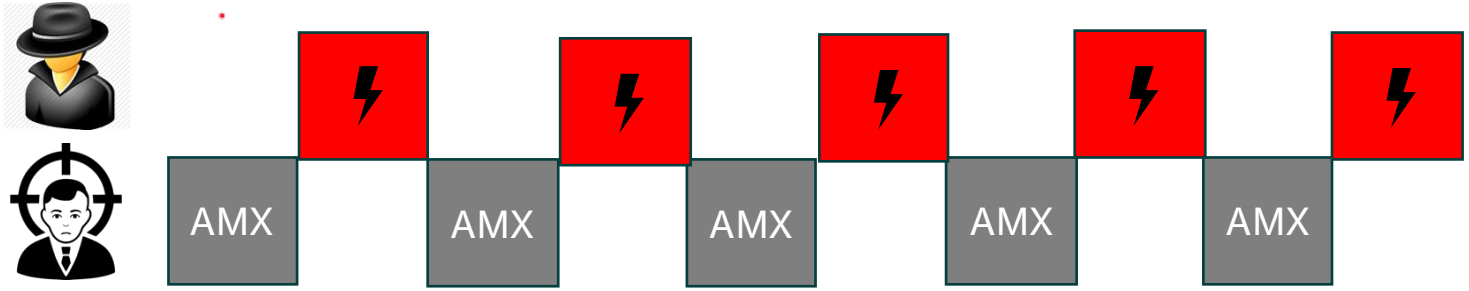
Parameter	Value
Hidden Size	256
Intermediate size	256
Number of heads	4
Embedding size	256
Number of layers	24

Methodology: CNN Membership Inference Attack

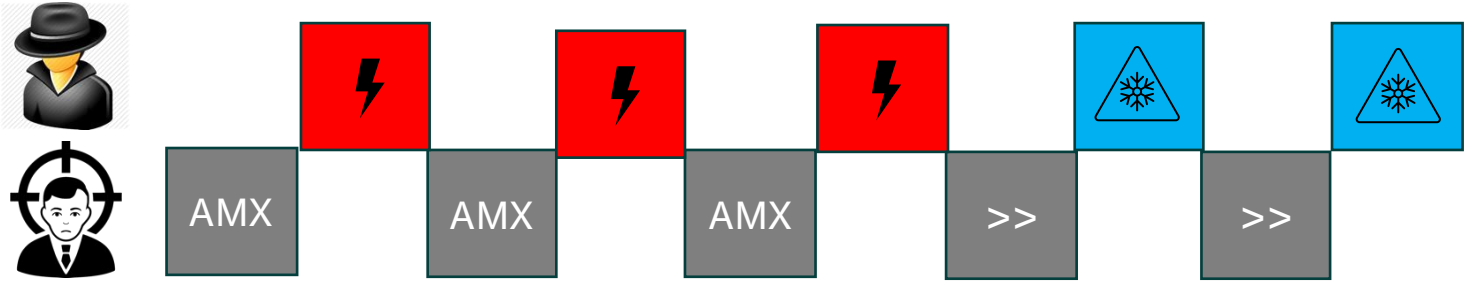
- 6-layer CNN, early exits placed after layer 2
- Trained on MNIST handwritten digit dataset

Parameter	Value
Layers	Conv2d MaxPool ReLU FullyConnected ReLU FullyConnected
Intermediate size	256
Number of heads	4
Embedding size	256
Number of layers	24

Transformer/CNN Membership Inference Attack

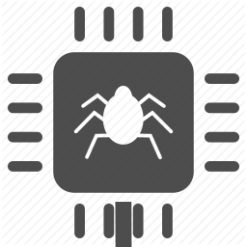


Likely non member data



Likely member data

Generic Magnifier Steps



AMX

Execute instruction

AMX dependent on instruction

AMX Cold State

AMX dependent on instruction

Remote Arbitrary Address Leakage Threat Model

1. Attacker mistrains branch predictor with in-bounds requests
2. Attacker performs out-of-bounds request
3. Response time leaks arbitrary out-of-bounds value

(1)

(2)

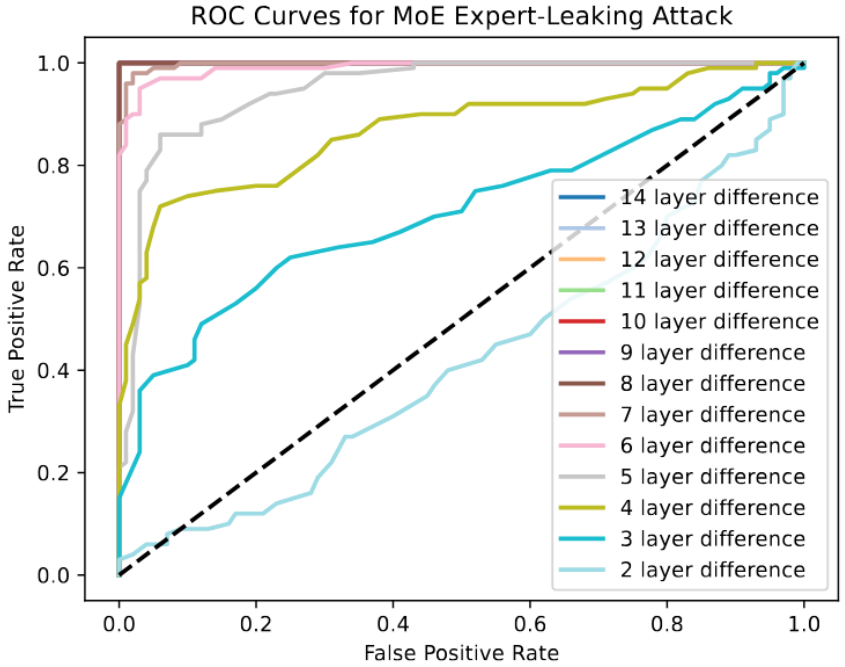
```
if (x < bound)
  if (array[x]) {
    AMX()
  }
}
```

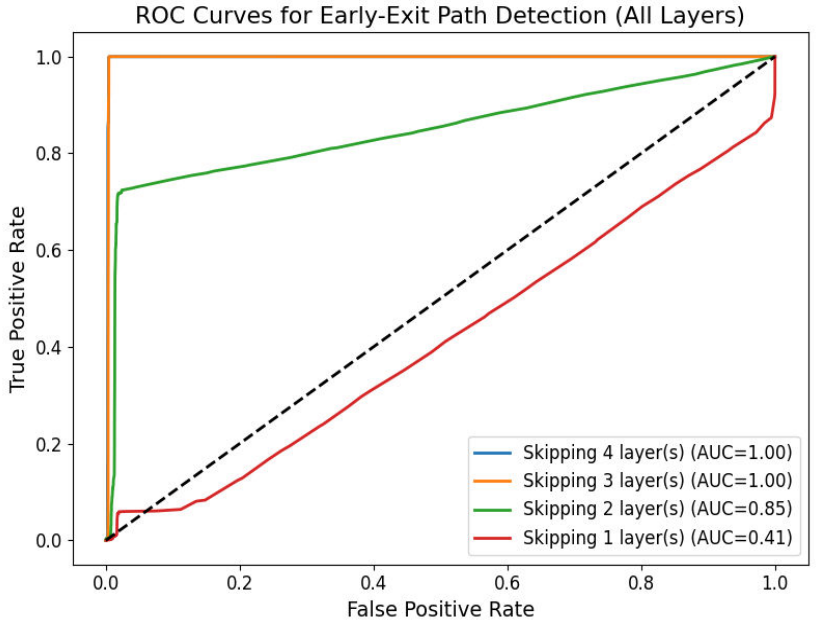
(3)

AMX()

Results

GateBleed Adaptive Neural Network Attacks

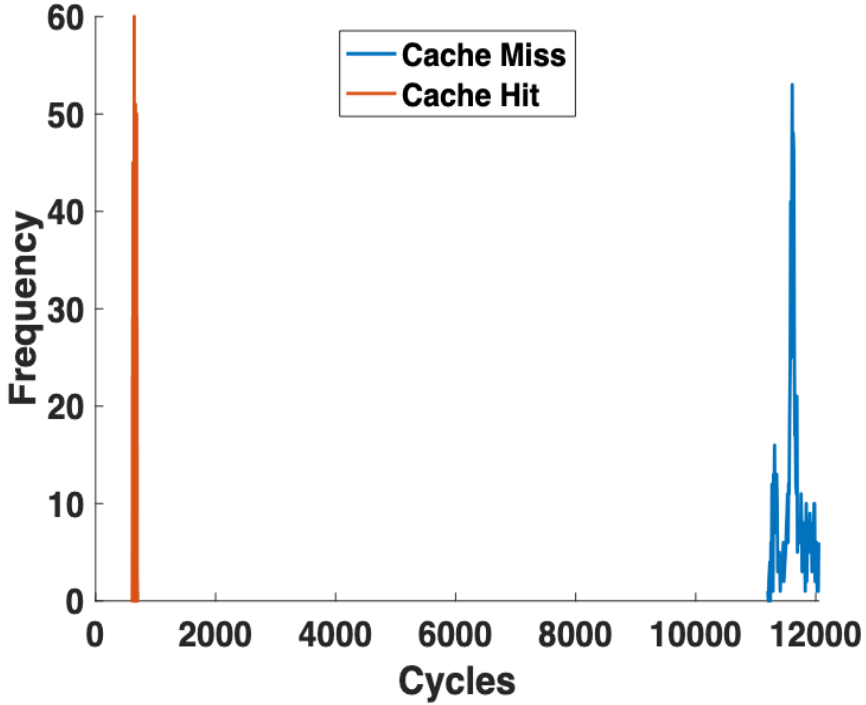




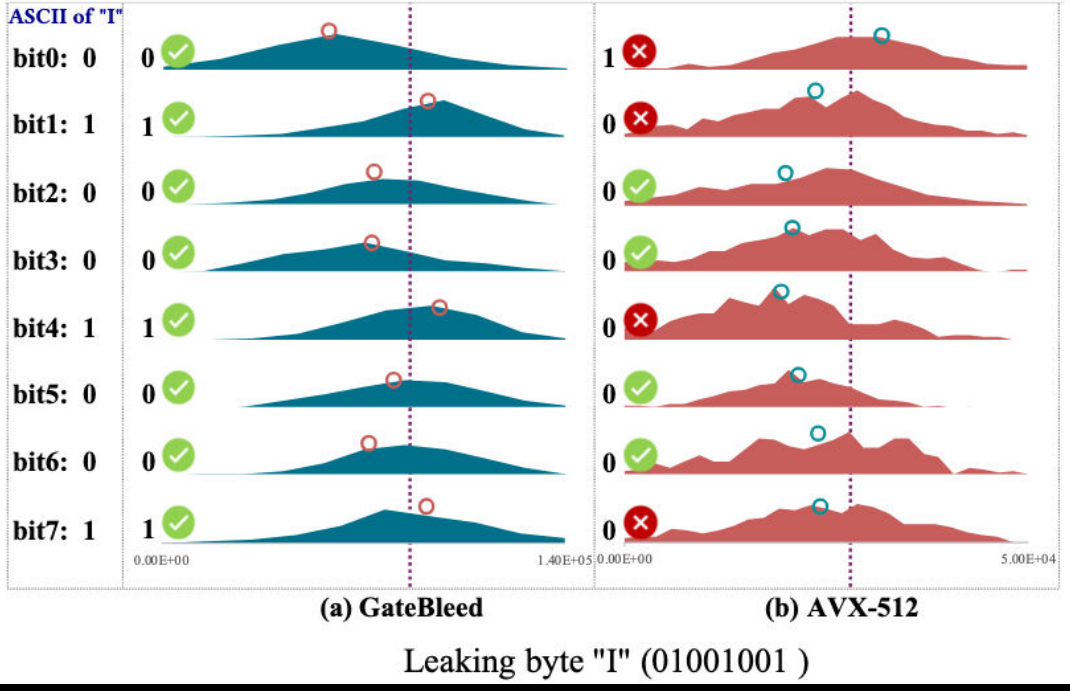
89% precision leaking expert choice with 100% accuracy.
81% accuracy on a Transformer
99% accuracy on an early-exit CNN classifier.

Generic magnifier

L1 hit/L3 miss turns into 5.5 µs timing difference with a **single** instruction



GateBleed arbitrary address leakage



70,000x Higher Leakage Rate than Net-Spectre. 0.07 bps vs. 0.000001 bps for NetSpectre (3 months to leak a single byte!!)

Mitigations

Evading Detection with Power Gating

(i) Low repetition rates required for GateBleed to succeed
(ii) Reset phase of GateBleed has no anomalous instructions
(Passive vs. Active Reset due to automatic cool down of AMX)
(iii) Few AMX performance counters

Attack	EVAX	PerSpectron	RHMD
GateBleed	**10%**	**9%**	**6%**
Microscope	80%	78%	63%
Flush+Flush	99%	87%	72%
Binoculars	98%	97%	85%
NetSpectre	97%	95%	94%
Hacky Racers	100%	98%	90%

Ineffective Mitigations

Cache mitigations
Disable TurboBoost
Disable C-States
Disable SMT

Confidence score masking
Dummy low power inst insertion

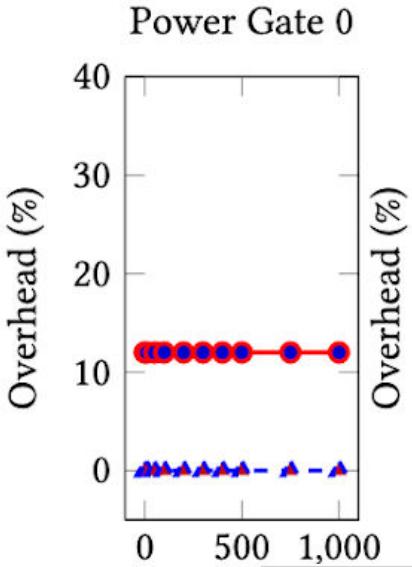
Lenovo UEFI update 3.20 or RHEL 9.5,

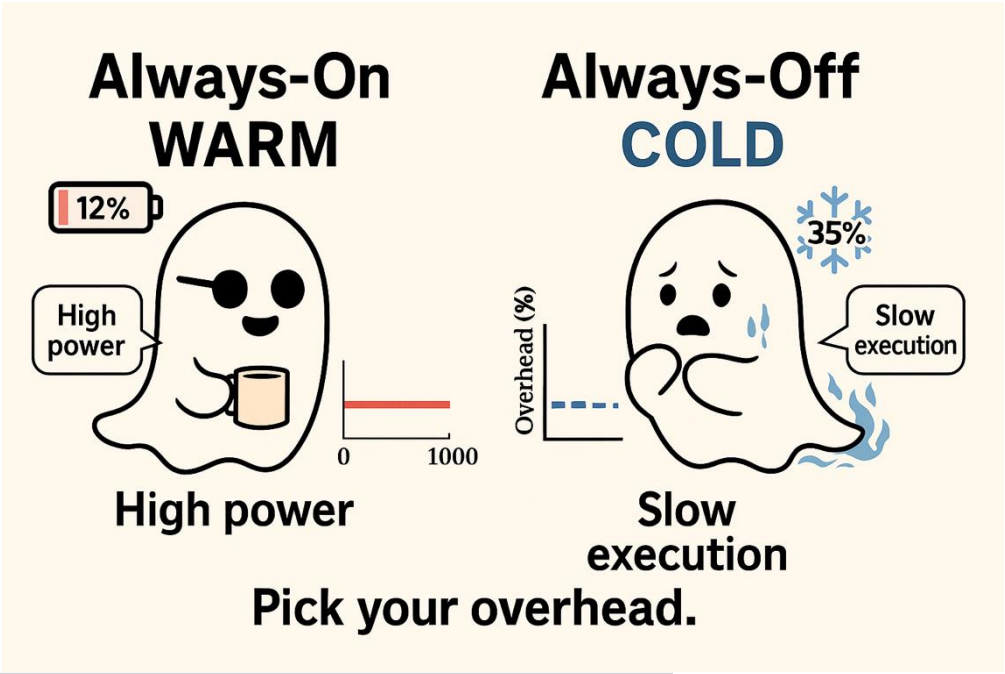
- The last 2 power gate stages are eliminated
- Gatebleed attacks on AI still work due to other 3 stages being present.
- Remote Spectre transmission channel mitigated, as the network latency dominates the power gating times

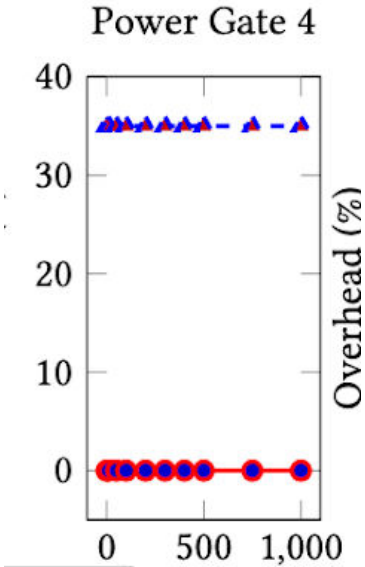
Mitigation: Power vs. Performance Trade Off

12% power

35% performance



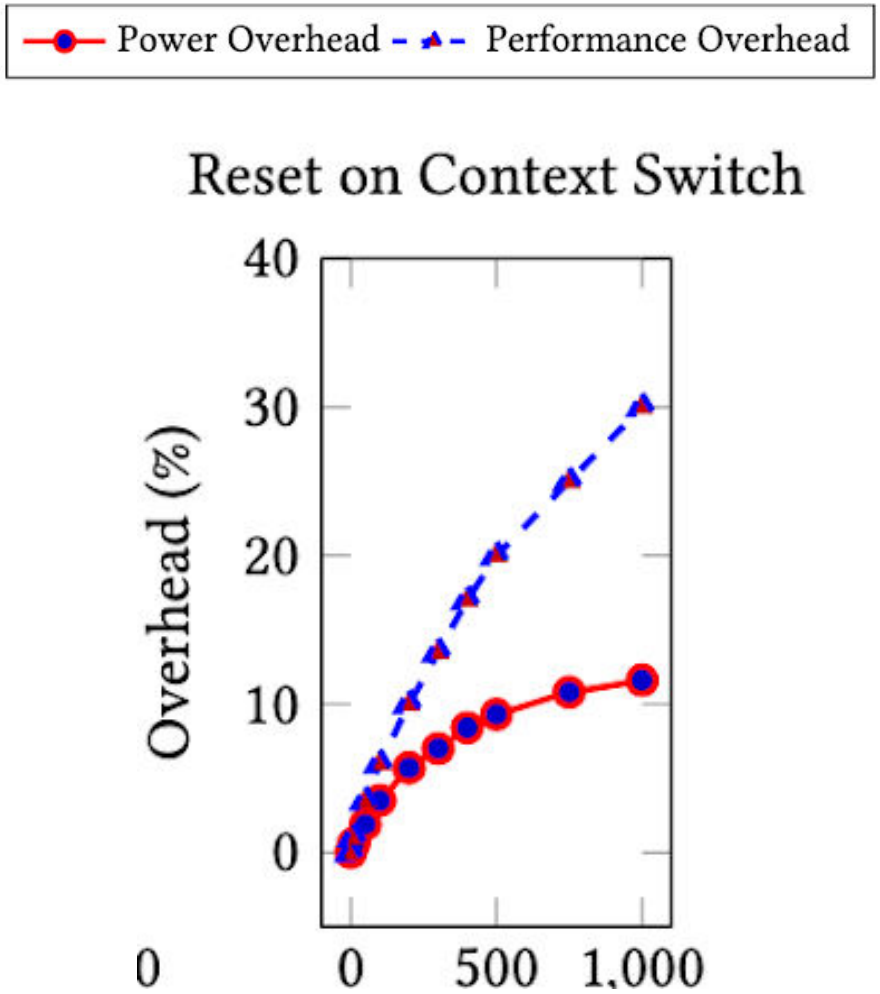




Power Overhead – Performance Overhead

AMX workload kept in a certain power stage

Powering off AMX on context switch mitigates GateBleed with tunable power and performance overhead



ACM Reproducibility Badges

Artifacts Available

Artifacts Evaluated — Functional

Results Reproduced

Code available at https://zenodo.org/records/17019733.
All results in this paper arereproducible within 40 minutes following the instructions at https://github.com/jkalya/gatebleed, and have been verified through the MICRO artifact evaluation.

News Releases

Hardware Vulnerability Allows Attackers to Hack AI Training Data

October 8, 2025 | Matt Shipman | 7-min. read

https://news.ncsu.edu/2025/10/ai-privacy-hardware-vulnerabili

https://github.com/jkalya/gatebleed.

NC STATE UNIVERSITY intel

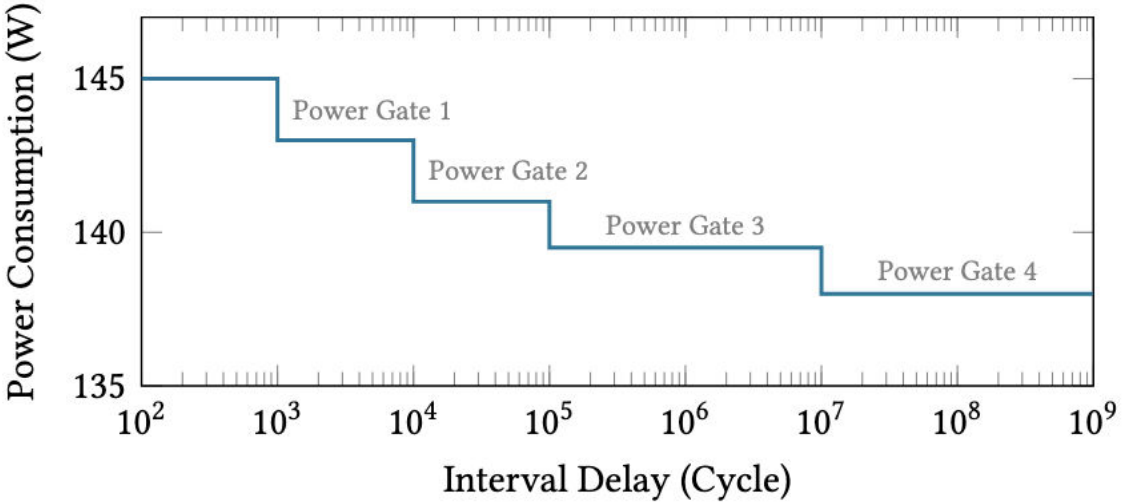
THANK YOU!

DO YOU HAVE ANY QUESTIONS?

GATEBLEED™ ATTACK

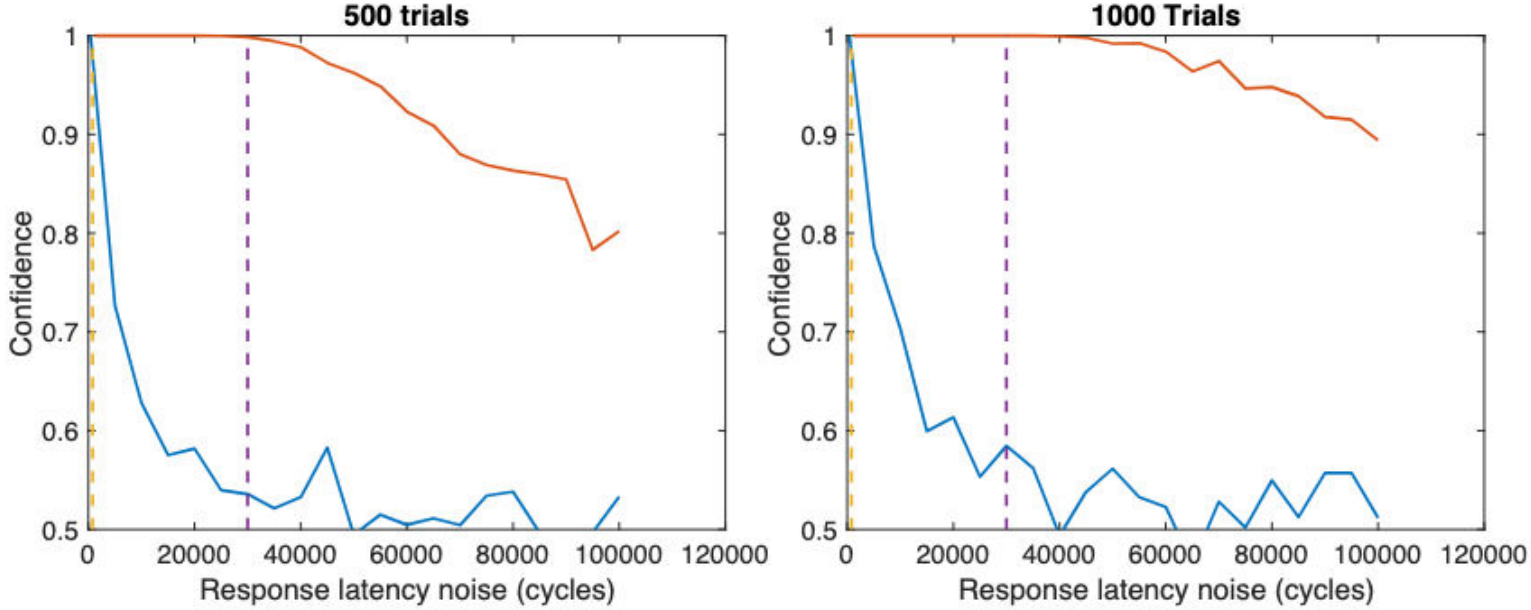
BAckup

The Root Cause is Power Gating.



Power Consumption Reduces Sharply with Interval Delay
The gradual decrease from stage 0 (142.08W) to stage 4 (138.49W) aligns with staged **power gating!**

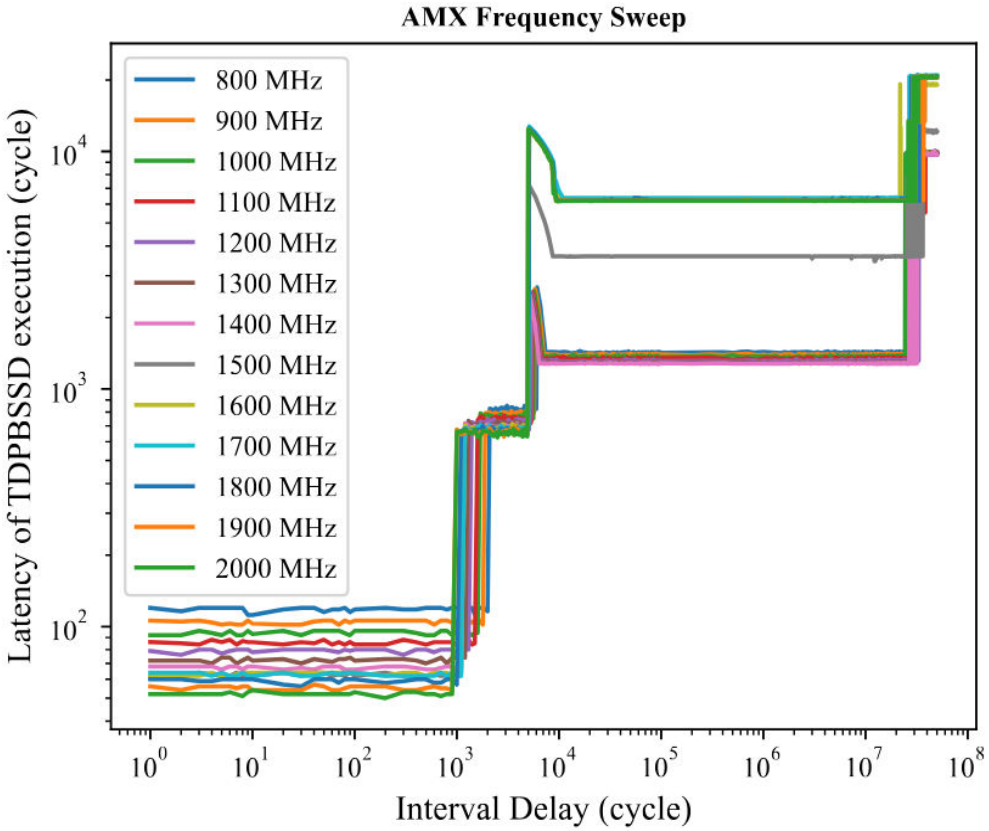
Superior Noise Resilience to Netspectre



As network noise increases, the NetSpectre with AVX-512 experiences a sharp decline in accuracy, approaching 0% almost immediately,
GateBleed achieves 100% accuracy with only 500 trial on production network

Root Cause Analysis

Timing Stages Remained with Fixed Frequency



Not the same as Hertzbleed!

Root Cause Analysis

Prefetching? NO!

C-State? NO!

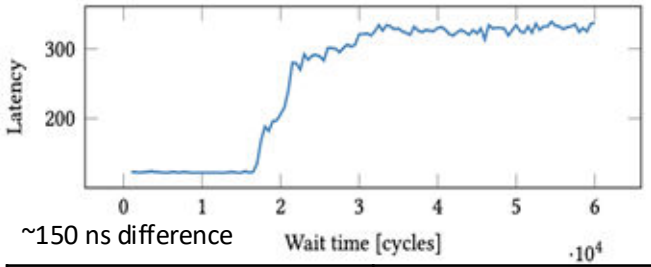
SMT? NO!

Kernel? NO!

Operand? NO!

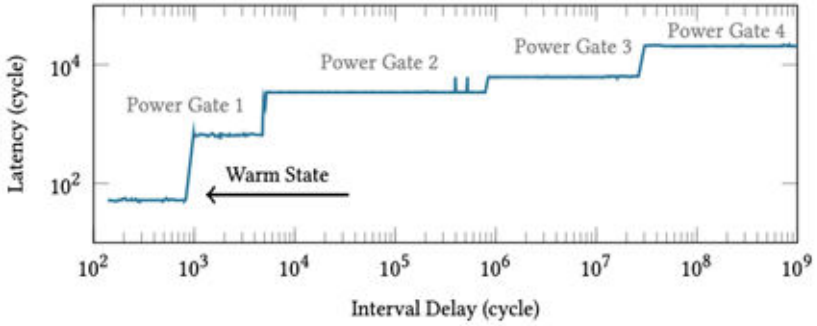
Attribute	Value	GB
P-state control	Autonomous (hardware-only)	✓
P-state control	Legacy (OS-only)	✓
P-state control	Cooperative	✓
P-state control	Disabled	✓
OS	RHEL 9.4	✓
OS	RHEL 9.5	○
OS	Ubuntu 22.04	✓
UEFI Version	SRV650-v3-3.14 (May 2024)	✓
UEFI Version	SRV650-v3-3.20 (June 2024)	○
Platform Power	Minimal Power	✓
Platform Power	Maximum Performance	✓
Platform Power	Efficiency, Favor Power	✓
Platform Power	Efficiency, Favor Performance	✓
Turbo Boost	Enabled	✓
Turbo Boost	Disabled	✓
All prefetchers	Disabled	✓
C-States	Enabled	✓
C-States	Disabled	✓
C1E	Enabled	✓
C1E	Disabled	✓

Cycle differences are massive



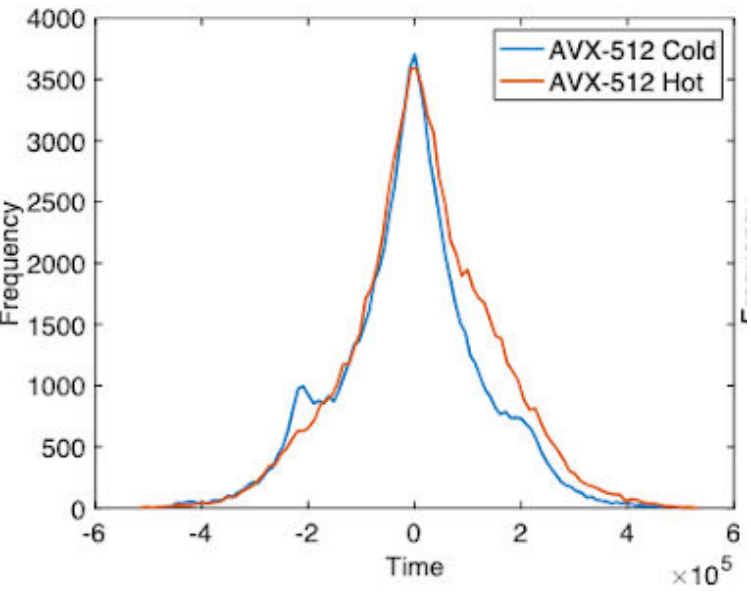
~150 ns difference

Hardware State	Timing
L1 hit	15 ns (30 tsc)
L3 miss	100 ns (200 tsc)
VPMULD hot	15 ns (30 tsc)
VPMULD cold	50 ns (100 tsc)



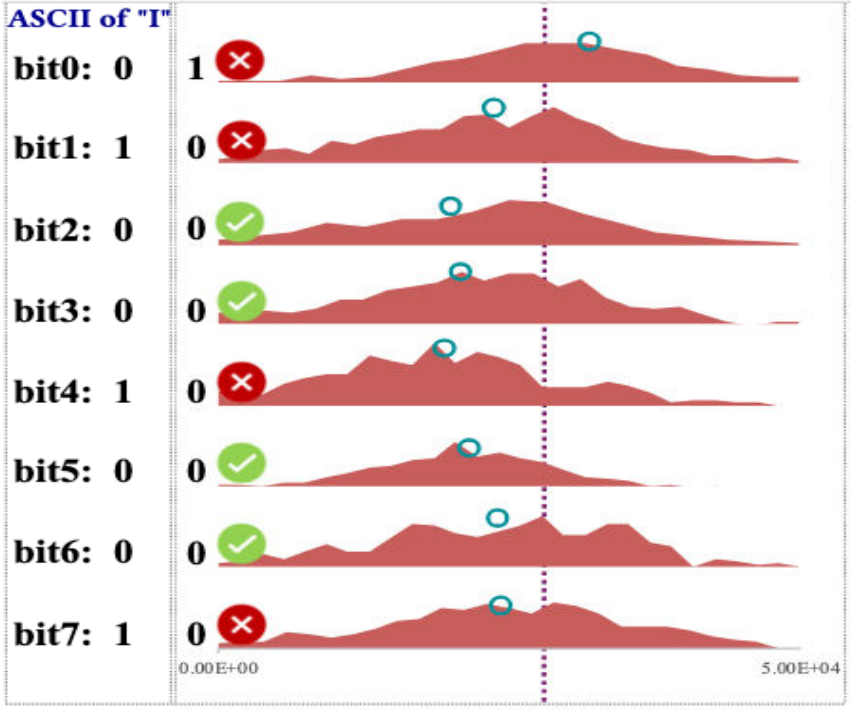
~10,000 ns difference between warm and power gate 4

Hardware State	Timing
AMX warm	25 ns (50 tsc)
AMX PG1	300 ns (600 tsc)
AMX PG2	3,000 ns (6,000 tsc)
AMX PG3	5,000 ns (10,000 tsc)
AMX PG4	10,000 ns (20,000 tsc)



(c) AVX-512 1 hop

Remote Spectre Attack Net-Spectre leaks 0.000001 bps on a real production network that's about 3 months to leak a single byte!! Malware detectors flag such high repetition attacks before they leak.

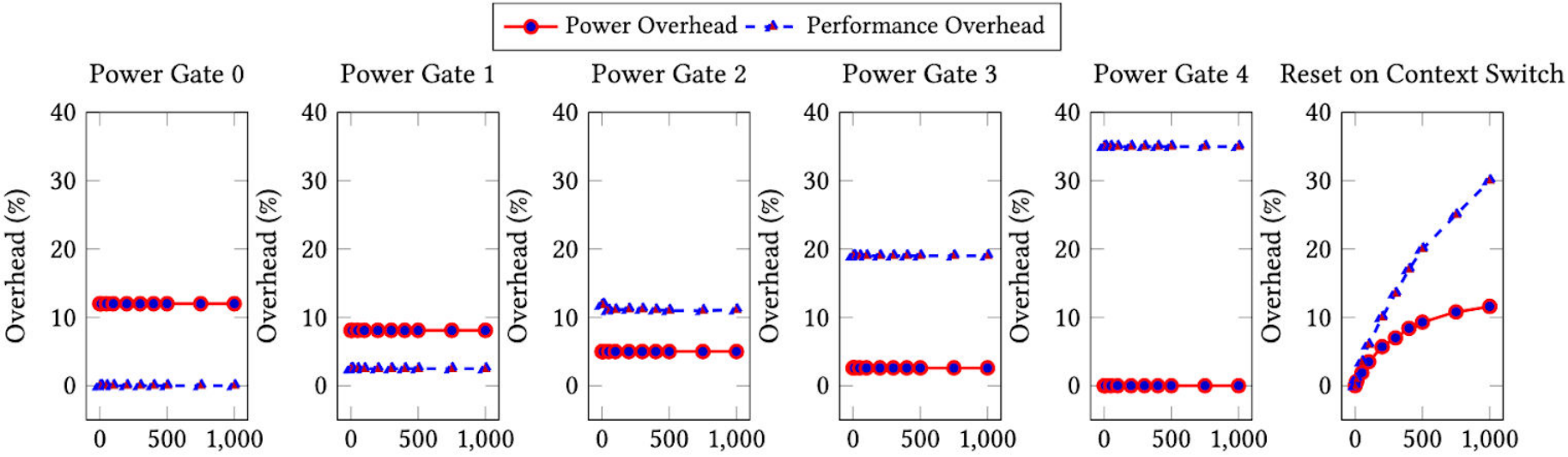


(b) AVX-512

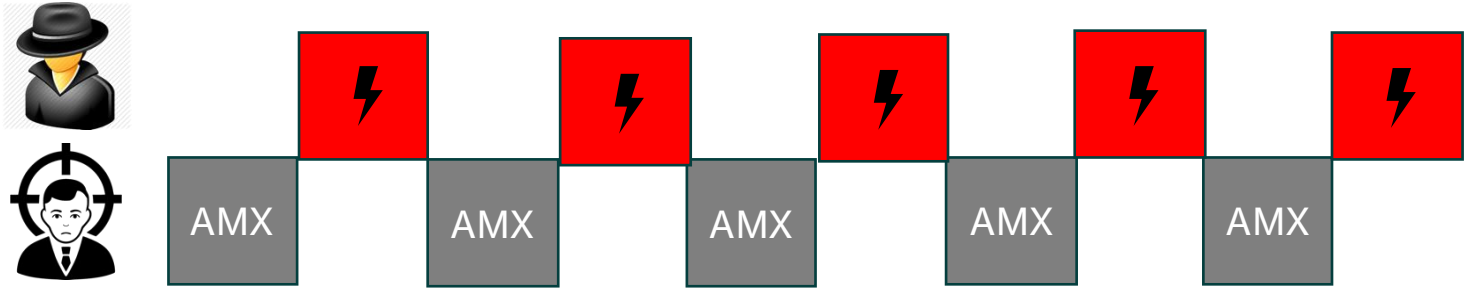
"I" (01001001)

We wanted to see if AI power gating can make microarchitectural attacks realistic on production network for the first time

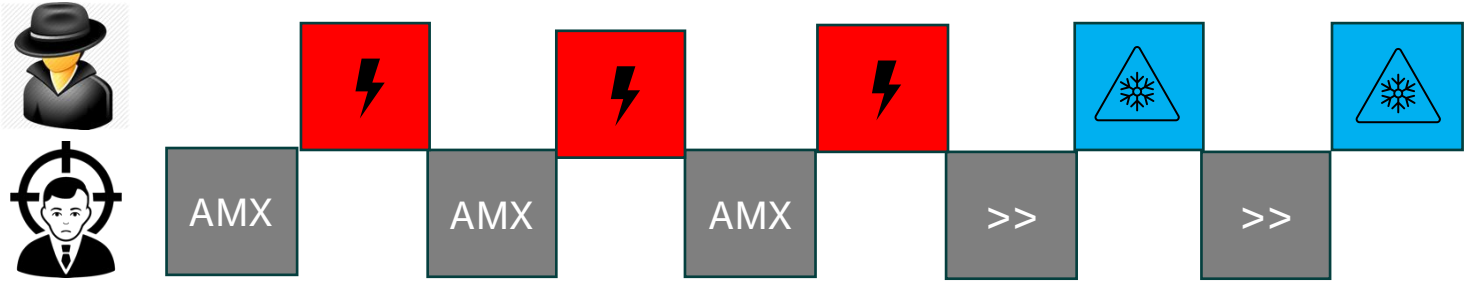
Powering off AMX on context switch mitigates GateBleed



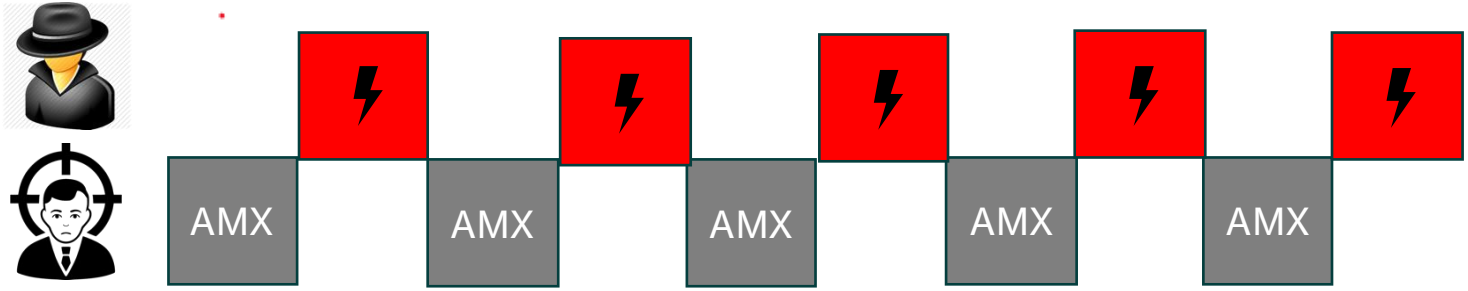
At low switching rates (e.g., <10 switches/sec), overhead is negligible
Less than 2% for both power and performance



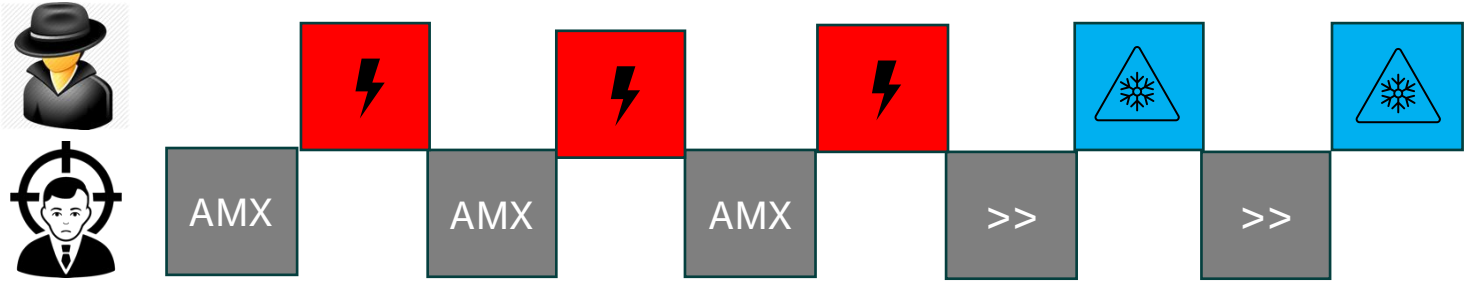
Larger capacity expert routed



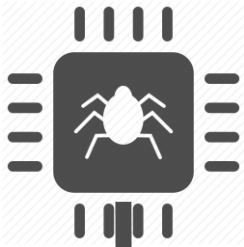
Smaller capacity expert routed



Likely non member data



Likely member data



AMX

Execute instruction

AMX dependent on instruction

AMX Cold State

AMX dependent on instruction

1. Attacker mistrains branch predictor with in-bounds requests
2. Attacker performs out-of-bounds request
3. Response time leaks arbitrary out-of-bounds value

(1)

(2)

```
if (x < bound)
  if (array[x]) {
    AMX()
  }
}
```

(3)

AMX()