THEME ARTICLE: TOP PICKS FROM THE 2025 COMPUTER ARCHITECTURE CONFERENCES

GATEBLEED: Exploiting On-Core Accelerator Power Gating for High Performance and Stealthy Attacks on AI

Joshua Kalyanapu*, Darsh Asher*, Farshad Dizani*, Azam Ghanbari*, Rosario Cammarota†, Aydin Aysu*, Samira Mirbagher Ajorpaz*, † *University of California Irvine, *NC State University*

Abstract—AI accelerators are being integrated directly into the CPU. This paper discloses GATEBLEED*, a family of timing side channels caused by aggressive power gating of one such on-core accelerator–Intel AMX. The attacks expose a four-axis shift in the microarchitectural attack threat model, not rooted in a bug but in a power optimization that makes efficient and seamless on-core AI acceleration possible: the target moves from leaking stored bytes to inferring AI training-data properties, specifically, hardware-observed membership inference attacks (MIAs) for the first time. The channel bypasses software defenses such as padding and confidence masking; it provides magnification that turns theoretical microarchitectural side channels, previously requiring controlled, low-traffic networks to be observable, into attacks with realistic leakage rates on production networks; it bypasses timer-coarsening defenses; it remains stealthy through a passive reset and low repetition requirement; and it escapes state-of-the-art anomaly detectors. Defense becomes a performance–power–privacy trilemma that requires rethinking how computer architects design the future of power-efficient, low-cost AI accelerators from the ground up.*

AI applications are power-intensive and power optimizations such as power gating are being deployed in on-core accelerators. Proposals on aggressive power gating are also surging: recent work ReGate [1][1], presented at MICRO 2025, shows that power gating on-core NPUs can save up to 32.8% power with only 0.5% performance overhead. However, the privacy and security risks of on-core accelerator power gating are poorly understood.

MLaaS interfaces and multi-tenant cloud services are widely deployed, yet little attention has been paid to the privacy impact of hardware-accelerator optimizations. Prior work extracts architectures, hyperparameters [4], or training-set membership through output logits and proxy models [5]. No prior work has studied whether such leaks can originate *below* the API—in the hardware optimization itself, where software defenses such as output masking [5] and Logit-Differential Privacy [6] training are insufficient.

Microarchitectural optimizations can introduce side channels that expose sensitive data. Microarchitectural transient attacks Spectre [2] and Meltdown [3] showed how speculative and out-of-order execution leak private data through subtle timing; subsequent research uncovered an avalanche of vulnerabilities exploiting the cache, prefetchers, execution ports, the micro-op cache, running power average (RAPL) interface, and dynamic voltage and frequency scaling (DVFS), showing that hardware-based timing channels undermine higher-level isolation and privacy guarantees.

Conventional timing side channels that target AI privacy leak data *stored* in memory or registers. Cache

XXXX-XXX © IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

[1]Historically, the juxtaposition of the ReGate paper [1] and GATEBLEED is rare: GATEBLEED is akin to publishing *Spectre* [2] in 1981—the same year the seminal branch prediction work appeared—rather than waiting until 2018, and akin to publishing *Meltdown* [3] [2018] in 1967—the year Tomasulo's algorithm was introduced—rather than 2018.

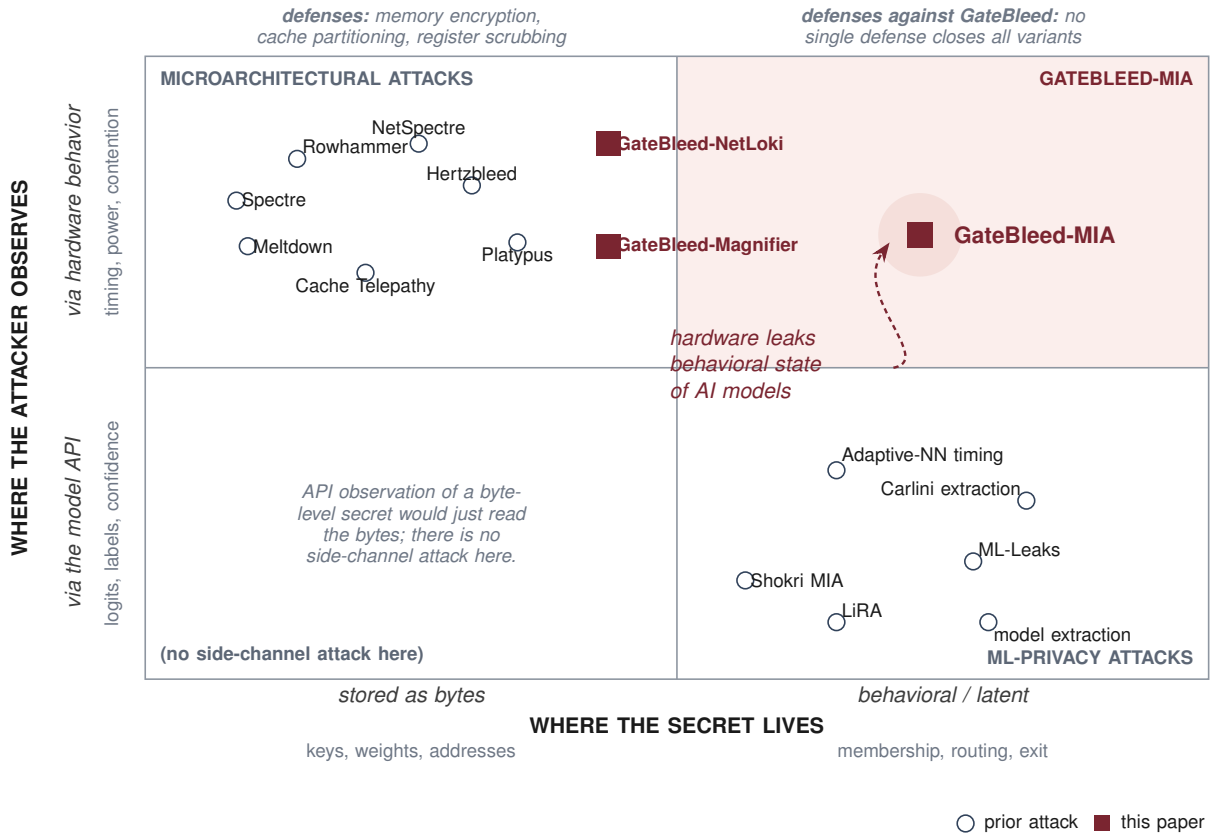


FIGURE 1: **Where GATEBLEED sits in the attack landscape.** Axes: where the secret lives (stored bytes vs. behavioral / latent) and where the attacker observes (model API vs. hardware behavior). GATEBLEED is a family of three variants. GATEBLEED-MIA occupies the previously empty top-right quadrant, which is for hardware observation of behavioral state of AI models. GATEBLEED-Magnifier and GATEBLEED-NetLoki extend the prior microarchitectural quadrant. No single defense closes all three.

Telepathy [4] uses cache timing to recover model architectures from memory; Hertzbleed [7], Platypus [8], and IdleLeak [9] use operand-dependent frequency throttling, power consumption, and CPU sleep states to leak register contents. *AI privacy attacks*, by contrast, obtain information that is *not* stored as bytes, but as training-set membership [5] and expert routing (Figure 1), which exist only as latent properties of the learned model. This paper extracts these properties via timing discrepancies from on-core accelerator power gating.

MIAs ask one question: "*Was this exact record in the training set*?": whether a person's cancer scan, private message, or copyrighted manuscript was used to train a model. Membership is legally and societally actionable: knowing that a record was used for training can establish violations of medical privacy, confidentiality agreements, or copyright, even if no data values are recovered. 2025 headlines show Getty Images sued Stability AI for training on copyrighted images [10]. Apple was sued for allegedly using pirated books to train AI models [11]. These cases show that the ability to determine what a model was trained on has major implications for user privacy and corporate liability: proving a model was trained on private or copyrighted data can become the basis for litigation, compliance violations, and reputational harm. This is why MIA papers are exploding in the privacy field, dominating the literature, e.g. the foundational MIA by Shokri et al. [5] (IEEE S&P 2017, over 8000 citations), ML-Leaks [12] (NDSS 2019, over 1400 citations), and LiRA [13] (IEEE S&P 2022, over 1400 citations).

Traditional MIAs often rely on access to confidence scores or logits because models tend to behave more confidently on training samples. Thus, one of the most relied on and fundamental defenses against confidence-score-based MIAs is output discretization, i.e., rounding, quantizing, or binarizing the model's

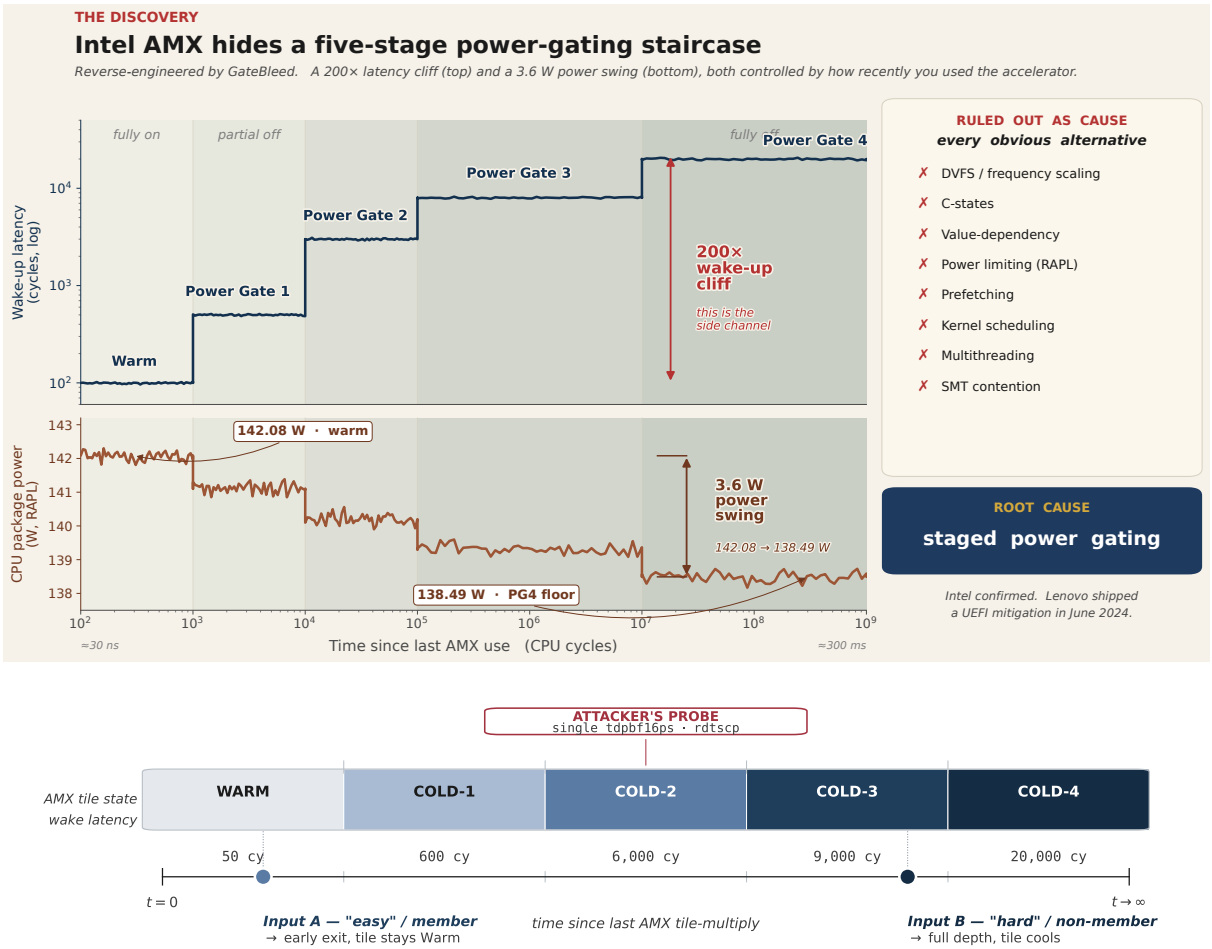


FIGURE 2: Performance states of Intel AMX due to staged power gating. Top: wake-up latency staircase versus time since the last AMX use, with a 200× latency gap across five plateaus reaching ∼ 20,000 cycles at the deepest stage. Middle: CPU package power (W, RAPL) on the same x-axis; the 3.6 W swing from 142.08 to 138.49 W is the corroborating signal. Bottom: a single attacker probe (`tdpbf16ps · rdtscp`) reads the plateau the accelerator occupies, encoded by the input the model processed.

confidence scores to reduce the statistical fingerprint of the training data in the outputs [5]. Therefore, in production, logits are often hidden, and many deployed MLaaS systems expose only labels.

KEY INSIGHT

Microarchitectural side channels traditionally leak **stored bytes**. GATEBLEED leaks **properties of the learned model training data, not weights or hyperparameters**, information that is never materialized in memory and that Spectre, Meltdown, or Rowhammer cannot reach *in principle*.

We uncover and characterize undocumented power gating in AMX, revealing five distinct latency states that vary based on time since last use: An undocumented power gate inside Intel AMX produces a 50 → 20,000 cycle latency signature on a single instruction, observable across processes, virtual machines, and SGX enclaves as shown in Figure 2.

Intel's Advanced Matrix Extensions, shipped in Sapphire Rapids and available in recent Intel datacenter CPUs, present an instruction interface that looks, on its face, like vanilla SIMD. Underneath, however, AMX is a discrete on-core accelerator sharing thermal budget with the rest of the core. To stay within that budget it implements five distinct power states, a Warm state and four progressively colder gated states, with wake latencies that span two and a half orders of magnitude.

A fixed-frequency sweep showed that the same staged

FIGURE 3: **The GATEBLEED gadget taxonomy across production ML libraries.** Three recurring structural classes appear in publicly available source: input-dependent routing, confidence-gated early exits, and session- or configuration-sensitive toggles. The right panel lists the minimum measurement capability the channel requires; presence of a pattern is necessary, not sufficient, for exploitation.

AMX pattern remains visible at every operating frequency, with only the absolute latencies scaling. This proves the phenomenon cannot be an artifact of DVFS or Turbo Boost, unlike Hertzbleed [7], and thus cannot be mitigated by turning off Turbo Boost. Other alternatives such as C-states, value dependence, RAPL power limiting, prefetching, kernel scheduling, multithreading, and simultaneous multithreading (SMT) contention are similarly ruled out.

We present GATEBLEED, the first microarchitectural *privacy* side channel that exploits on-core accelerator power gating to infer training-set membership, MoE routing, and early exits without access to logits or stored artifacts. The attack succeeds against three classes of AI privacy targets, namely mixture-of-experts routing, early-exit decisions, and training-set membership, without ever observing model outputs, logits, or confidences.

Because GATEBLEED exploits hardware design, the vulnerability cannot be fixed with a simple software update. Effective mitigation likely requires hardware changes that take years to reach deployed systems and have high power overhead. Because GATEBLEED doesn't rely on model outputs such as confidence scores, it evades existing defenses built for inference attacks. For example, proposed software-level defenses such as confidence-score masking [5] and differential privacy [6] do not mitigate GATEBLEED.

This article also presents GATEBLEED-NETLOKI, a remote Spectre-v1 attack and a microarchitectural *magnifier* for environments with only a coarse timer. As a magnifier, GATEBLEED amplifies a subtle 200-cycle timing difference into an observable 10,000-cycle signal, enabling the distinction of an L1 cache hit from an L3 miss even under exceptionally coarse timing conditions. As a covert channel, the remote attack achieves a leakage rate 70,000× higher than NETSPECTRE [14] on an identical production network while

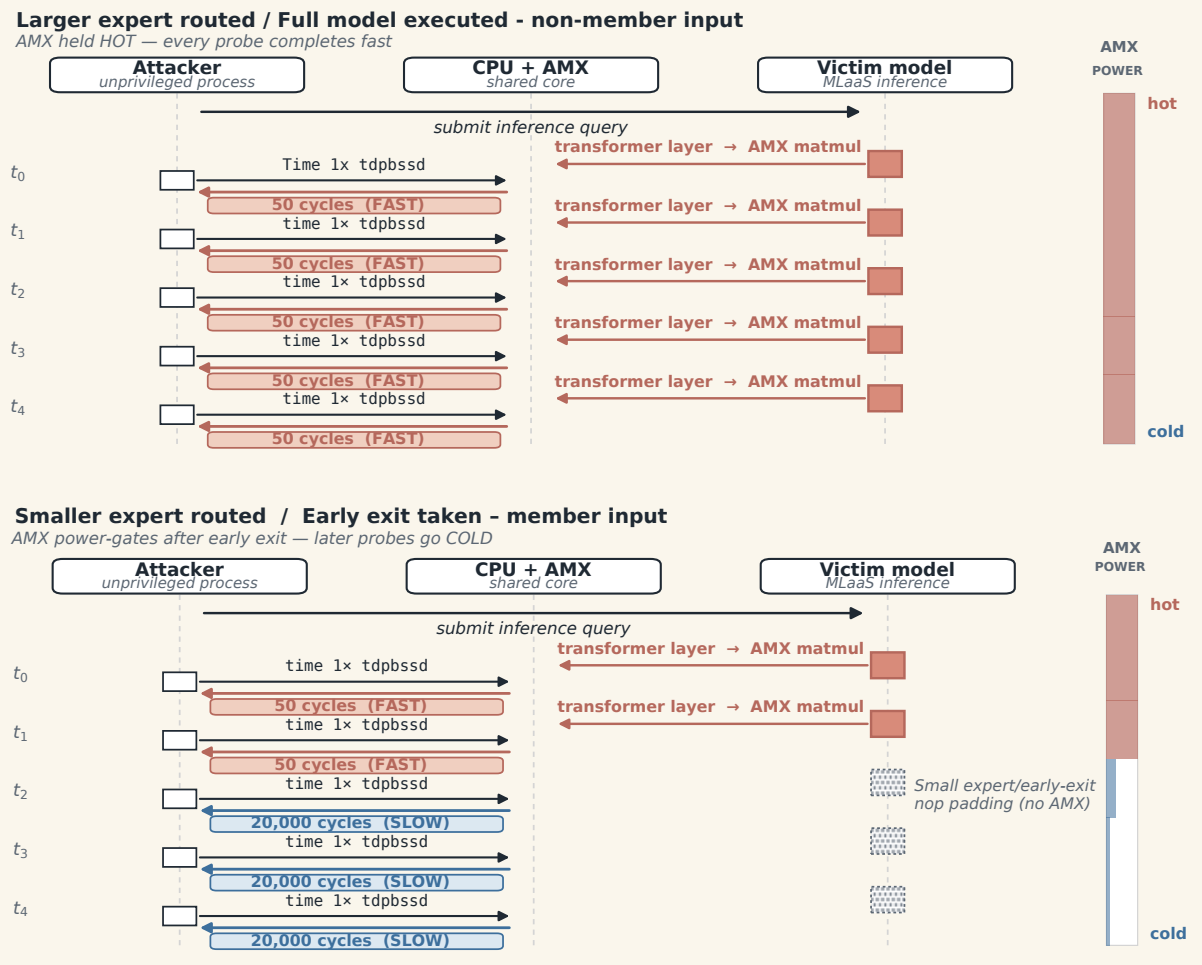


FIGURE 4: GATEBLEED Attack on heterogeneous mixture-of-experts and early-exiting models. Top: a non-member input routes to the larger expert; AMX stays *hot* ($\sim$ 50 cycles per probe). Bottom: a member input engages the small expert or an early exit; AMX power-gates and probes take $\sim$ 20,000 cycles.

remaining undetected by Evax [15], PerSpectron [16], and RHMD [17].

We revisit state-of-the-art detectors and show they achieve near-100% accuracy of recent attacks due to visible execution patterns and high repetition rates. They fail to detect GATEBLEED because they (i) ignore misuse of hardware *power* optimizations, (ii) cannot resist low repetition rates, (iii) miss its passive reset phase, and (iv) lack accelerator-specific counters.

We discuss mitigations for GATEBLEED, including a microcode update that maintains AMX in a single power stage and compiler-inserted dummy AMX operations to defend against the cross-process variant increasing power overhead by up to 12%. We further evaluate the associated power and performance costs, finding that powering off AMX on a context switch incurs a power overhead of 2–12% depending on the context-switch rate.

Responsible Disclosure. We disclosed this issue to Intel between May 2023 and May 2024. Intel confirmed our findings, and Lenovo released a UEFI firmware mitigation (Version 3.20, Build ID ESE126H, Critical) in June 2024, which mitigates the most critical GATEBLEED variants particularly those exploiting AMX power gates 3, 4, and 5. We have open-sourced the GATEBLEED proofs-of-concept[2].

[2]Code available and all results in this paper are reproducible within 40 minutes following the instructions at https://github.com/jkalya/gatebleed, and have been verified through the MICRO artifact evaluation.

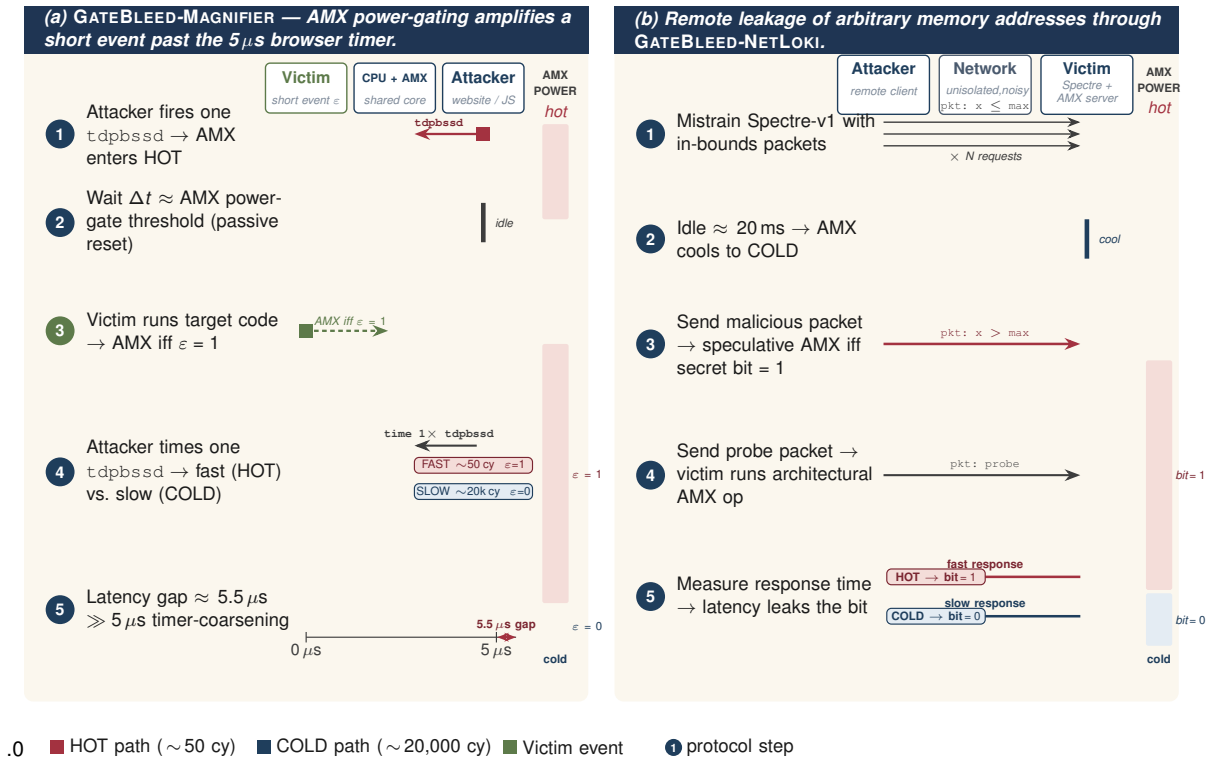


FIGURE 5: Two end-to-end protocols on the same AMX wake-up latency gap. **(a)** The in-browser magnifier converts a 200-cycle victim event into a ~5.5 μs latency gap, observable past Chrome's 5 μs timer coarsening. **(b)** The remote GATEBLEED-NETLOKI attack uses Spectre-v1 plus AMX power-gating to leak a memory bit per probe: the speculative AMX op fires only when the secret bit is 1, leaving AMX HOT; probe response time discriminates HOT (bit=1) from COLD (bit=0). Red: HOT path; navy: COLD path; green: victim event.

Threat Model

Depending on the variant of our attack, the adversary's goal is to infer (i) whether a given input was part of the training set similar to the goal of prior MIA attacks [5], (ii) which expert or routing branch was activated, or (iii) at which exit a depth-adaptive model terminated.

We assume confidence scores are masked as a defense against conventional confidence-based MIA attacks [5]. We also assume the neural network software is padded for constant end-to-end time, to show how AMX power gating introduces an observable state visible to other processes despite being independent of end-to-end inference time.

We consider an adversary who can issue inference queries to a target model and who shares a CPU core with the inference process. The adversary does not require privileged access: no shared memory mapping, no SMT sibling, no model-output logits, and no DVFS or RAPL access are needed. The single capability required is the ability to time an AMX-bearing instruction sequence and to observe its cold-start latency.

The three end-to-end attacks we evaluate are not the only places this channel can arise. Examining widely used production ML libraries, we identify a recurring *structural* pattern, code paths in which an input- or state-dependent decision selects whether, or how long, AMX is exercised, which we group into three classes and summarize visually in Figure 3.

Axis 1: New Microarchitectural Side Channel Target (MIA without Confidence)

GATEBLEED leaks targets that previous microarchitectural side channels could not reach: training-set membership, mixture-of-experts routing, and early-exit decisions. These are properties of the running model,

Family	Configuration	Acc.	TPR	FPR	TNR	FNR	Prec.
Routing *MoE expert inference*	Layer Gap ≥ 8	100%	100%	0%	100%	0%	1.00
	Gap = 7	98%	98%	2%	98%	2%	0.98
	Gap = 6	96%	95%	3%	97%	5%	0.97
	Gap = 5	90%	86%	6%	94%	14%	0.93
	Gap = 4	82%	74%	10%	90%	26%	0.88
	Gap = 3	69%	62%	25%	75%	38%	0.71
	Gap = 2	46%	40%	48%	52%	60%	0.45
Adaptive Inference *early-exit & membership*	Early-Exit CNN	99.72%	99.99%	0.54%	99.46%	0.01%	0.99
	Early-Exit Transformer	100%	100%	0%	100%	0%	1.00
	Transformer MIA (LLM)	81%	78%	16%	84%	22%	0.89

Best Result

Routing
100%
Layer Gap ≥ 8
MoE expert ID

Early-Exit
99.72%
CNN, six layers
layer disclosure

MIA
81%
Billion-param LLM
membership

TABLE 1: Evaluation metrics across verified end-to-end attacks using GATEBLEED. Rows are grouped by attack family; the rightmost panel highlights the strongest result per family.

not bytes in memory, so the entire literature of output-side privacy defenses—top-k truncation, output rounding [5], DP-SGD [6], MemGuard [18] —is structurally blind to them. We demonstrate this with two end-to-end attacks; Table 1 reports all metrics.

Threat model. The MIA attacker holds a sample x and wants to determine whether x was in the training set of an MLaaS-served model. The attacker is the only party in the system besides the model: they submit x as an inference query, the MLaaS process runs the model on x on a shared CPU core, and the attacker times AMX on the same core immediately afterwards. There is no third-party victim user; the "victim" of the timing measurement is the model's own inference of the attacker's chosen query.

Attack on Transformer

We trained a heterogeneous mixture-of-experts (HMoE) transformer with two experts of equal width but different depth: a *big* expert (24 layers) and a *small* expert (10–22 layers). The asymmetry is deliberate—training-set members route disproportionately to the small expert—and reproduces the routing imbalance reported for production MoE models. Training set: 784 English sentences. Test set: 300 held out.

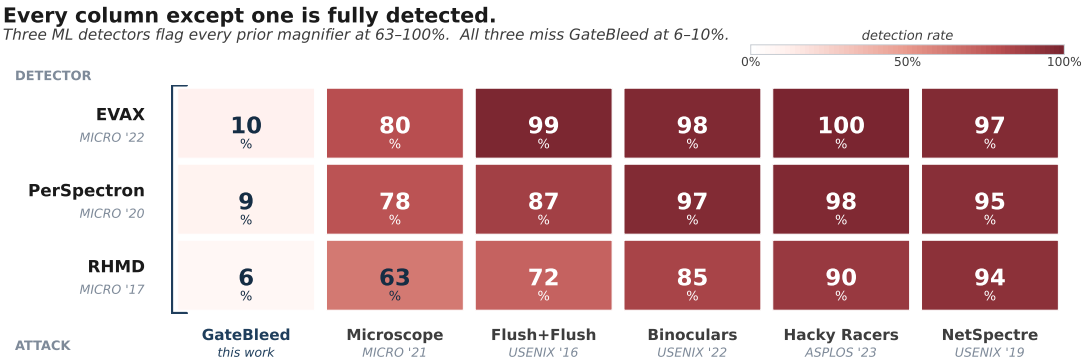
The attacker submits its candidate sample x as an inference query, then repeatedly times one AMX operation on the same core. Time slices that immediately follow the model's inference using AMX register as *fast*; the rest register as *slow*. Larger experts sustain AMX longer and produce more fast slices, so the fast/slow ratio reveals which expert the model ran on the attacker's query (Figure 4). From this single bit per probe we infer expert selection at 100% accuracy and training-set membership of x at 81%. Both are upper bounds under our routing function and noise floor.

Attacks on CNN

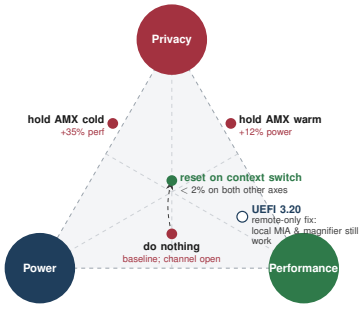
We trained a 6-layer CNN on MNIST that exits after layer 2 when sufficiently confident. The standard defense against timing leakage on adaptive networks is *nop* padding the shorter path so all inputs complete in equal time. GATEBLEED bypasses this defense: padding equalizes the model's instruction stream, but GATEBLEED does not measure that stream—it measures AMX wake-up latency on the attacker's own subsequent slice, which *nop* instructions do not exercise. The attack distinguishes real early-exit decisions from padded ones at 99.72% accuracy (0.54% false positive rate) and infers training-set membership of the queried sample at 99%.

Axis 2: From a Theoretical Microarchitectural Side Channel Attack to a Practical One

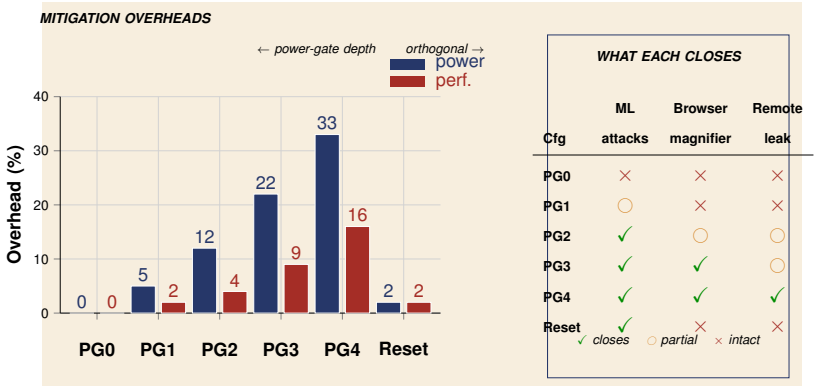
GATEBLEED runs on a real production network carrying tens of terabytes of daily traffic and achieves a leakage rate 70,000× higher than NETSPECTRE in identical conditions. The reason is the magnitude of the AMX wake-up latency gap: where NETSPECTRE



(a) Detection rate of three ML-based microarchitectural attack detectors on GATEBLEED versus five prior attacks.



(b) Geometric trade-off space.



(c) Measured overheads and channel coverage.

FIGURE 6: **Defenses against GATEBLEED: stealth bypass and the performance–power–privacy trilemma.** **(a)** Three state-of-the-art microarchitectural-attack detectors (EVAX, PERSPECTRON, RHMD) catch every prior attack we tested at 63–100% but miss GATEBLEED at 6–10%; GATEBLEED stays below the 70% operating threshold on all three. **(b)** Geometric view of GATEBLEED mitigations after Chord's identifier-ring layout (Stoica et al., SIGCOMM 2001). Closeness to a corner indicates the axis a mitigation favours; distance to the opposite edge is its cost. *Reset on context switch* (green) sits closest to the privacy corner at small cost on the other two axes. *UEFI 3.20* eliminates only the deepest two power-gate stages, partially mitigating the remote NETSPECTRE-style variant but leaving the first power-gate transition—and so the local magnifier and MIA channels—intact. **(c)** The same configurations measured directly: deeper power-gate stages close more channels but cost more power and runtime; the rightmost *Reset* bars show the context-switch reset variant, which closes ML attacks at 2%/2% overhead but leaves the magnifier and remote leak intact.

amplified small cache-timing differences, GATEBLEED amplifies a gap reaching $\sim$ 20,000 cycles between the warm state and the deepest power-gating stage—200$\times$ larger—and a signal of this size survives the jitter that buries cache-scale timing.

GATEBLEED-NETLOKI: Magnification & Remote Arbitrary Address Leakage

The same wake-up latency gap supports two attack shapes (Figure 5). GATEBLEED-MAGNIFIER runs in-process—e.g., from JavaScript in a browser—and amplifies a 200-cycle victim event into a 5.5 μs signal observable past Chrome's 5 μs timer-coarsening defense (and the implicit microsecond-scale timer cap in WebAssembly), using a single AMX instruction. Unlike

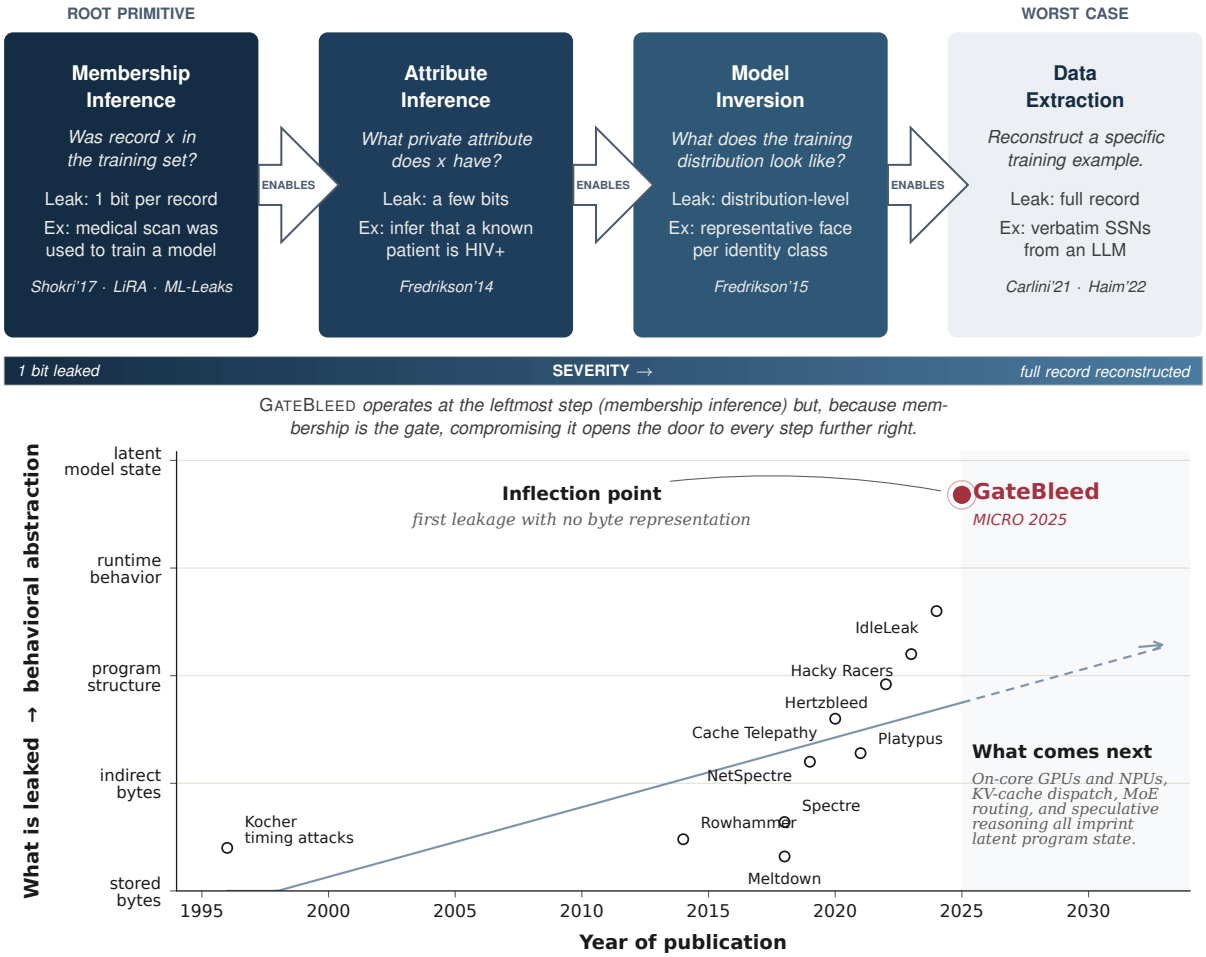


FIGURE 7: Membership inference is the root primitive; every more invasive attack builds on that single bit. GATEBLEED operates at the leftmost step, so compromising it opens the door to every step further right. Microarchitectural attacks are migrating from bytes to behavior; GATEBLEED is the first to leak target information that has no byte representation in memory. The dashed projection extends the trajectory into on-core GPUs and NPUs, KV-cache dispatch, MoE routing, and speculative reasoning—all of which imprint latent program state into timing.

prior magnifiers such as Hacky Racers [19]—winner of the ASPLOS 2023 Distinguished Paper Award and a MICRO Top Picks 2023 Honorable Mention—GATEBLEED does not require long instruction streams; the magnification gadget is a single tile-multiply instruction whose latency is determined by accelerator residency rather than by instruction-level parallelism pressure.

As ML inference moves to the edge and browsers grow paths to on-core accelerators, a magnification primitive that survives microsecond-scale timer coarsening with a single instruction opens a threat model that today's browsers are not built to mitigate. The *NetLoki* variant repurposes the same gap as the transmission channel of a remote Spectre-v1 attack, replacing the cache- or port-contention channel used by prior network-side variants: the secret is loaded speculatively into an AMX-bearing path; the wake-up latency observed by a remote client is the side-channel readout.

Axis 3: Stealth

Three state-of-the-art ML-based microarchitectural attack detectors—EVAX [15], PERSPECTRON [16], and RHMD [17]—catch every prior attack we tested including NetSpectre [14] and Hacky Racers [19] at 63–100% accuracy. However, they catch GATEBLEED at only 6–10% detection rate (Figure 6).

Four mechanisms explain the gap: (i) those detectors were trained on *active* resets—loops, contention, instruction-level stress—whereas GATEBLEED resets *passively* by waiting for AMX to cool; (ii) one AMX instruction is enough to register the wake-up latency, so the per-probe footprint is too small to trigger repetition-count thresholds; (iii) they treat hardware *power* optimizations as out of scope and so do not monitor them; and (iv) the perf-counter set on Sapphire Rapids exposes only one AMX-related event (cycles spent in AMX), which we found insufficient to discriminate attacker probes from legitimate workloads.

AVAILABLE TIMER COARSENING INEFFECTIVE

20,000× resolution loss required. To suppress the magnifier channel, timer resolution must drop from 0.5 ns (Sapphire Rapids TSC) to 10 μs—a 20,000× degradation. Chrome's aggressive coarsening is 5 μs, leaving a 200× margin still exploitable. No deployed system can absorb the latency cost of 10 μs timer granularity for legitimate workloads.

Axis 4: The Performance–Power–Privacy Trilemma

Mitigating GATEBLEED returns the AMX accelerator toward a single-power-stage operating regime. Two principled options collapse the wake-up latency gap: a *warm-stage lock* that keeps AMX powered between operations, and a *cold-stage lock* that powers AMX off and forces a uniform wake-up on every operation. Each option closes the channel at a cost on a different axis. Figure 6 summarizes the trade-off as we measured it.

Four operating positions

Default AMX. The factory operating regime. Fast and energy-efficient, but exposes the wake-up latency gap that GATEBLEED exploits.

Warm-stage lock. A microcode or firmware change that keeps AMX in the warm power stage at all times. Closes the channel in our measurements; the cost is a power overhead in the range of 2 to 12%, depending on the context-switch rate of the workload.

Cold-stage lock. A microcode change that holds AMX fully power-gated and pays the full wake-up on every use, removing the time-since-last-use signal. Closes the channel in our measurements; the cost we observed is 35% in additional runtime.

Reset on context switch. A lighter-weight mitigation resets AMX state at context-switch boundaries. In our measurements this reduces but does not eliminate the magnifier variant of the channel, because the magnifier does not require a context-switch boundary to be measurable.

PRACTICAL RECOMMENDATION

Context-switch AMX reset is affordable and effective. For typical context-switch rates (10–500 switches/sec), the power overhead lands between 2% and 9%, well below typical security-sensitive workload tolerances. The defense is fundamentally a hardware/OS co-design problem: pure software cannot reach the AMX power-state machine. Lenovo's UEFI 3.20 update implements a firmware-level variant.

Software-level defenses do not address this channel

Defenses developed for inference-time attacks operate on the model-output channel. This is insufficient because the AMX wake-timing channel sits below that defense layer.

DEFENSES THAT NO LONGER APPLY

API and training defenses target the wrong window.
Top-k truncation: hides confidences. GATEBLEED never sees them.
Output rounding: blunts logit signal. GATEBLEED observes timing.
DP-SGD: bounds influence on output distribution. GATEBLEED bypasses the output entirely.
MemGuard: noises the confidence distribution. Same.

> *Privacy guarantees that depend on hiding outputs assume the attacker only sees outputs.* GATEBLEED *shows that on modern accelerator-equipped CPUs, the attacker also sees the silicon thinking.*

Future Impact

Membership is a fundamental privacy primitive: a model can behave measurably differently on training-set members vs non-members [5]. GATEBLEED shows that microarchitectural power optimizations themselves carry that distinction, bypassing software defenses operating on outputs. Figure 7 situates the evidence supporting this prediction for the future trajectory: membership inference is the entry point of an escalation pipeline through attribute inference, model inversion, and—at worst—reconstruction of training records.

Adaptive neural networks are here to stay. The recent DeepSeekMoE [20] design reaches 671 billion

parameters without requiring a top-end GPU by routing each token through only a small subset of experts. As production AI continues to rely on input-dependent architectures for efficiency—MoE routing, KV-cache reuse, quantization dispatch, agentic tool use—the population of GATEBLEED-style vulnerable applications will only grow.

Figure 7 places GATEBLEED on this longer arc: the field has been moving from leaking stored bytes toward leaking what computation does, and GATEBLEED marks the point at which that trajectory crosses into targets that never existed in memory at all. GATEBLEED therefore opens a new attack landscape, similar to the post-Spectre era, in which future work will explore how microarchitectural features enable AI privacy attacks beyond MIA—attribute inference, model inversion, and reconstruction—across diverse models and deployment settings.

None of the mitigations we measured (Figure 6) is cost-free—each closes the channel only by paying on power, runtime, or coverage—and this is exactly the kind of nontrivial trade-off that produced years of variants and defenses after Spectre [2] and Meltdown [3]. We therefore expect GATEBLEED to catalyze a stream of defense proposals with real industry consequences: AI chip vendors will need to redesign on-core accelerators to close emerging behavioral channels, while AI model developers and MLaaS providers may need new software-level mitigations to protect privacy on hardware already in deployment.

Future attacks are likely to resemble stealthy, high-performance workloads rather than suspicious activity, slipping past today's malware detectors and antivirus systems and forcing microarchitecture design from the ground up with AI privacy in mind—including detection mechanisms and performance counters explicitly designed to expose AI-privacy-relevant behavior. Like Spectre [2] and Meltdown [3], GATEBLEED may become a canonical teaching example in computer architecture and AI-systems courses, illustrating why performance, power, and AI privacy must be studied jointly rather than in isolation.

Whether other on-core accelerators admit channels of the same shape, and whether the trilemma can be closed in hardware without paying its costs in software, remain open questions.

Acknowledgments

The authors thank anonymous reviewers for their helpful comments and feedback. This work was supported by Semiconductor Research Corporation (SRC) contract #2025-HW-3306 and Intel Labs.

REFERENCES

1. Y. Xue and J. Huang, "Regate: Enabling power gating in neural processing units," in *Proceedings of the 58th IEEE/ACM International Symposium on Microarchitecture*, pp. 1160–1177, 2025.

2. P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, "Spectre attacks: Exploiting speculative execution," in *IEEE Symposium on Security and Privacy (S&P)*, 2019.

3. M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, *et al.*, "Meltdown: Reading kernel memory from user space," in *27th USENIX Security Symposium (USENIX Security 18)*, pp. 973–990, 2018.

4. M. Yan, C. W. Fletcher, and J. Torrellas, "Cache telepathy: Leveraging shared resource attacks to learn DNN architectures," in *29th USENIX Security Symposium (USENIX Security '20)*, pp. 2003–2020, 2020.

5. R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, IEEE, 2017.

6. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

7. Y. Wang, R. Paccagnella, E. T. He, H. Shacham, C. W. Fletcher, and D. Kohlbrenner, "Hertzbleed: Turning power side-channel attacks into remote timing attacks on x86," in *31st USENIX Security Symposium (USENIX Security '22)*, pp. 679–697, 2022.

8. M. Lipp, A. Kogler, D. Oswald, M. Schwarz, C. Easdon, C. Canella, and D. Gruss, "PLATYPUS: Software-based power side-channel attacks on x86," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 355–371, 2021.

9. F. Rauscher, A. Kogler, J. Juffinger, and D. Gruss, "Idleleak: Exploiting idle state side effects for information leakage," in *Network and Distributed System Security Symposium (NDSS) 2024*, Feb. 2024. Network and Distributed System Security Symposium 2024 : NDSS 2024, NDSS 2024 ; Conference date: 26-02-2024 Through 01-03-2024.

10. M. Coulter, "Aiming for fairness: an exploration into getty images v. stability ai and its importance in the landscape of modern copyright law," *DePaul J. Art Tech. & Intell. Prop. L*, vol. 34, p. 124, 2024.

11. M. Scarcella, "Apple sued by authors over use of books in AI training." https://www.reuters.com/sustainability/boards-policy-regulation/apple-sued-by-authors-over-use-books-ai-training-2025-09-05/, Sept. 2025. Accessed: May 22, 2026.

12. A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

13. N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914, IEEE, 2022.

14. M. Schwarz, M. Schwarzl, M. Lipp, J. Masters, and D. Gruss, "Netspectre: Read arbitrary memory over network," in *Computer Security–ESORICS 2019: 24th European Symposium on Research in Computer Security, Luxembourg, September 23–27, 2019, Proceedings, Part I 24*, pp. 279–299, Springer, 2019.

15. S. M. Ajorpaz, D. Moghimi, J. N. Collins, G. Pokam, N. Abu-Ghazaleh, and D. Tullsen, "EVAX: Towards a practical, pro-active and adaptive architecture for high performance and security," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1218–1236, 2022.

16. S. Mirbagher-Ajorpaz, G. Pokam, E. Mohammadian-Koruyeh, E. Garza, N. Abu-Ghazaleh, and D. A. Jiménez, "Perspectron: Detecting invariant footprints of microarchitectural attacks with perceptron," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1124–1137, IEEE, 2020.

17. K. N. Khasawneh, N. Abu-Ghazaleh, D. Ponomarev, and L. Yu, "Rhmd: Evasion-resilient hardware malware detectors," in *Proceedings of the 50th Annual IEEE/ACM international symposium on microarchitecture*, pp. 315–327, 2017.

18. J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 259–274, 2019.

19. H. Katzman, Z. N. Zhao, A. Morrison, C. W. Fletcher, and J. Torrellas, "Hacky Racers: Exploiting instruction-level parallelism to generate stealthy fine-grained timers," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '23)*, 2023.

20. D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024.

Joshua Kalyanapu is a Ph.D. candidate at North Carolina State University, Raleigh, North Carolina, USA. He received his MS in Electrical Engineering from North Carolina State University. His research interests include hardware security and side-channel attacks. Contact him at jkalyan@ncsu.edu.

Darsh Asher is a Ph.D. candidate at North Carolina State University, Raleigh, North Carolina, USA. He received his MS in Electrical Engineering from North Carolina State University. His research interests include learning about new microarchitectural optimizations in modern CPU and accelerator designs and finding vulnerabilities in them for security and privacy attacks. Contact him at dkasher@ncsu.edu.

Farshad Dizani is a Ph.D. candidate at North Carolina State University. He received his MS in Electrical Engineering from the University of Tehran. His research interests include computer architecture, microarchitectural security, and signal processing. Contact him at fdizani@ncsu.edu.

Azam Ghanbari is a Ph.D. candidate at North Carolina State University. She received her MS in Computer Engineering from the University of Tehran. Her research interests include computer architecture and applications of AI for high-performance and secure

CPU design. Contact her at aghanba2@ncsu.edu.

Aydin Aysu is an Associate Professor and University Faculty Scholar in the Department of Electrical and Computer Engineering at North Carolina State University. His research interests include hardware security. He received his Ph.D. degree from Virginia Tech. Contact him at aaysu@ncsu.edu.

Rosario Cammarota is an Associate Professor of Computer Science at UC Irvine whose work focuses on secure systems and technologies for computing on encrypted data. He received his Ph.D. from UC Irvine. Contact him at rcammaro@uci.edu.

Samira Mirbagher Ajorpaz is an Assistant Professor in the Department of Electrical and Computer Engineering at North Carolina State University. She received her Ph.D. in Computer Science from Texas A&M University. Her research interests lie at the intersection of microarchitecture, security, and AI privacy. Contact her at smirbag@ncsu.edu.