POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

TEXAS: probing workloads in shared GPU environments through TEXture and ASync Pipeline Side Channels

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Samuele Motta

Advisor: Prof. Luca Cassano

Co-advisor: Elia Lazzeri

Academic year: 2025-2026

---

1. Introduction

In modern computing paradigms, Graphics Processing Units (GPUs) have transitioned from specialized graphics hardware to powerful general-purpose accelerators. They underpin critical modern tasks, including deep learning inference and training, scientific simulations, and large-scale data processing. This shift has been vastly accelerated by modern software environments and hardware extensions [8].

Concurrently, the high monetary cost of cutting-edge hardware has led cloud service providers, multi-user research workstations, and shared clusters to adopt multi-tenant resource sharing models. In these standard bare-metal environments, multiple independent user applications run concurrently on the same physical device, relying on the hardware scheduler and driver layers to interleave execution contexts. However, this native co-location introduces an inherent trade-off between execution efficiency and strict tenant isolation. Co-scheduled applications inevitably contend for microarchitectural hardware pipelines, creating a substantial attack surface for microarchitectural side-channel information leakage [7].

While CPU-based side-channel analysis represents a thoroughly mapped field [10, 11], GPU security exploration is still evolving. Prior works on GPU side and covert channels often focused on on-chip networks [1], translation lookaside buffers under rigid isolation models [12], multi-GPU cache hierarchies [2], or uncore components [6]. Crucially, previous research overlooks how distinct signals from different hardware pipelines can be integrated to increase fingerprinting resolution.

This summary introduces **TEXAS** (**TEX**ture and **AS**ync Pipeline Side Channels), a novel contention-based side-channel framework. Operating entirely from unprivileged user-space code within a native bare-metal execution environment, TEXAS targets two independent, hardware-isolated GPU memory pathways: the Texture Units (TEX) pipeline and the modern asynchronous global-to-shared memory copy (Async) pipeline. By deploying a hierarchical dual-stream classification architecture, TEXAS fuses these complementary latency patterns to accurately fingerprint co-resident workloads, identify architectural deep learning structures, and track execution dynamics across different generations of NVIDIA hardware architectures.

2. The TEXAS Attack Framework

2.1. Threat Model

The framework operates under a non-privileged bare-metal co-location scenario. The attacker is assumed to run standard, unprivileged user-level CUDA applications on the target machine, sharing the physical GPU simultaneously with a victim process via the native driver runtime. The attacker lacks explicit permission to access the victim's memory footprint, execution instructions, or file structures. The attack remains completely passive, observing only the localized latency variations resulting from raw microarchitectural pipeline resource contention under standard concurrent execution scheduling.

2.2. Spy Kernels Architecture

To map the execution fingerprint of the target GPU, TEXAS deploys two specialized, independent measurement routines designed to target specific execution pipelines.

TEX Spy Kernel The TEX spy kernel targets the specialized texture units and their dedicated cache. The tracking algorithm leverages continuous, highly dense texture-fetching operations designed to stress the pipeline's maximum bandwidth. To prevent accidental caching hits from giving false timing results, the spy enforces pseudo-random memory indexing using a Linear Feedback Shift Register (LFSR). The memory allocation covers a 1 MB texture buffer space (2^{18} indexing entries), which intentionally exceeds typical streaming multiprocessor (SM) L1 texture cache sizes to measure higher-level contention variances across the shared GPU memory fabric.

Async Spy Kernel The Async spy kernel focuses specifically on the hardware asynchronous data copy engine (accessed via the CUDA `cp.async` instruction) introduced in modern architectures [9]. This engine bypasses the register file, copying blocks directly from global memory to shared memory. The spy kernel sets up a large global memory arena along with isolated shared memory blocks inside the execution threads. During active runtime, it issues a non-blocking sequence of asynchronous data transactions, followed immediately by hardware synchronization barriers. Competing global-to-shared memory transfers from a co-resident victim strain the underlying data buses, resulting in measurable execution latency fluctuations at the synchronization boundaries.

2.3. Sequential Execution and Feature Fusion

A core constraint of the TEXAS design is that the TEX and Async spy kernels are executed sequentially rather than simultaneously. If executed concurrently, the memory traffic from one spy kernel would inevitably distort the latency metrics of the other, introducing mutual interference and noise into the collected measurements. Consequently, the framework conducts two separate sampling phases alongside the victim's workload to capture isolated, clean time-series traces for each pathway.

Each raw trace is segmented into fixed-length windows of 256 measurements with a stride of 16 samples. Each window is independently standardized via Z-score normalization, making the model invariant to absolute latency baselines while preserving the structural signal shape. To prevent data leakage, the train and validation splits are partitioned at the CSV file level: windows from independent execution runs are reserved exclusively as a held-out inference set.

The two window sequences are then processed by a dual-stream BiLSTM classifier. Each stream consists of two stacked Bidirectional LSTM layers with a hidden size of 64 units per direction, producing a 128 dimensional feature vector from the last timestep. The two feature vectors are concatenated into a 256 dimensional joint representation and passed through a fusion block — comprising a fully connected layer, Batch Normalization, and Dropout ($p = 0.3$) — before the final classification head. The two branches share no weights, allowing each to specialize independently on its respective pipeline signal. The network is trained with the Adam optimizer ($\eta = 10^{-3}$) under Cross-Entropy loss, with early stopping based on validation F1 (patience of 15 epochs).

3. Experimental Evaluation and Results

3.1. Experimental Setup

The experimental validation was conducted in native, multi-process bare-metal environments across three distinct generations of NVIDIA hardware architectures: an NVIDIA RTX 3050 (Ampere Architecture), an NVIDIA RTX 4070 (Ada Lovelace Architecture), and an NVIDIA RTX 5070 (Blackwell Architecture). The test suite comprised 30 deep learning topologies and workloads across 5 core industrial categories: Convolutional Neural Networks (CNNs), Transformers, Recurrent Neural Networks (RNNs), Generative Diffusion Models and Cryptocurrency Mining Algorithms. For the profiling validation phase, high-resolution latency records were systematically collected across these target environments.

3.2. Hierarchical Classification Performance

The dual-stream network uses a Bidirectional Long Short-Term Memory (BiLSTM) backbone to handle the fused time-series sequences. Classification operates hierarchically, first identifying the coarse workload category before drilling down into model family variations and specific model instances.

Category-Level Fingerprinting At the root level, the dual-stream classifier effectively isolates macro-level execution footprints under cross-process resource contention. On the reference RTX 4070 platform, the classifier achieves a weighted F1 score of **0.94** across five core categories (CNN, Transformer, RNN, Diffusion, and Mining), as shown in Figure 1. This result outperforms both single-stream baselines: the TEX-only classifier reaches F1 $\approx$ 0.93 on four categories, while the Async-only classifier reaches only F1 $\approx$ 0.70, confirming that fusing complementary signals from two independent pipelines yields higher discriminative power than either source alone.

Model-Level Fingerprinting Once the general category is identified, dedicated dual-stream BiLSTM classifiers distinguish fine grained topologies within each domain. Table 1 summarizes the weighted F1 scores for each classifier on the RTX 4070.

Table 1: Model-level weighted F1 scores on the RTX 4070.

Category	Classifier	F1
CNN	DenseNet vs. MobileNet	0.93
CNN	DenseNet-121/169/201	0.63
Transformer	Family-level (BERT/DeepSeek/Gemma/GPT-2)	0.89
Transformer	DeepSeek model-level	0.74
Transformer	Gemma model-level	0.82
RNN	GRU vs. LSTM	0.96
Diffusion	SD v1.5 vs. tiny-sd	0.63
Mining	Category-level only (uniform saturation)	

The highest model-level F1 is achieved for RNN (0.96), where GRU and LSTM produce markedly different contention patterns. The lowest scores (0.63) appear for the DenseNet variant classifier and the Diffusion classifier: the three DenseNet variants differ only in depth, producing gradual rather than sharp contention differences, while Stable Diffusion and tiny-sd share the same transformer-based attention backbone, making their pipeline footprints difficult to separate. Mining workloads completely saturate GPU resources regardless of the specific hashing algorithm, producing a uniform contention footprint with insufficient intra-class variance for model-level discrimination; they are therefore retained exclusively for category-level classification.

3.3. Cross-GPU Architecture Generalization

To test the portability of the attack, the framework was evaluated across the hardware generations under native runtime execution. Since absolute latency values depend on the specific microarchitecture, classifiers were retrained from scratch on data collected from each target GPU. At the category level, both the RTX 3050 and RTX 5070 achieve a weighted F1 score of **0.91** on four categories (CNN, Transformer, RNN, and Diffusion), confirming architecture-independent discriminative power. The slight gap with respect to the RTX 4070 (0.94) is partly attributable to the exclusion of Mining, which is among the easiest categories to classify. At the model level, retrained classifiers maintain stable performance across hardware configurations, as summarized in Table 2.

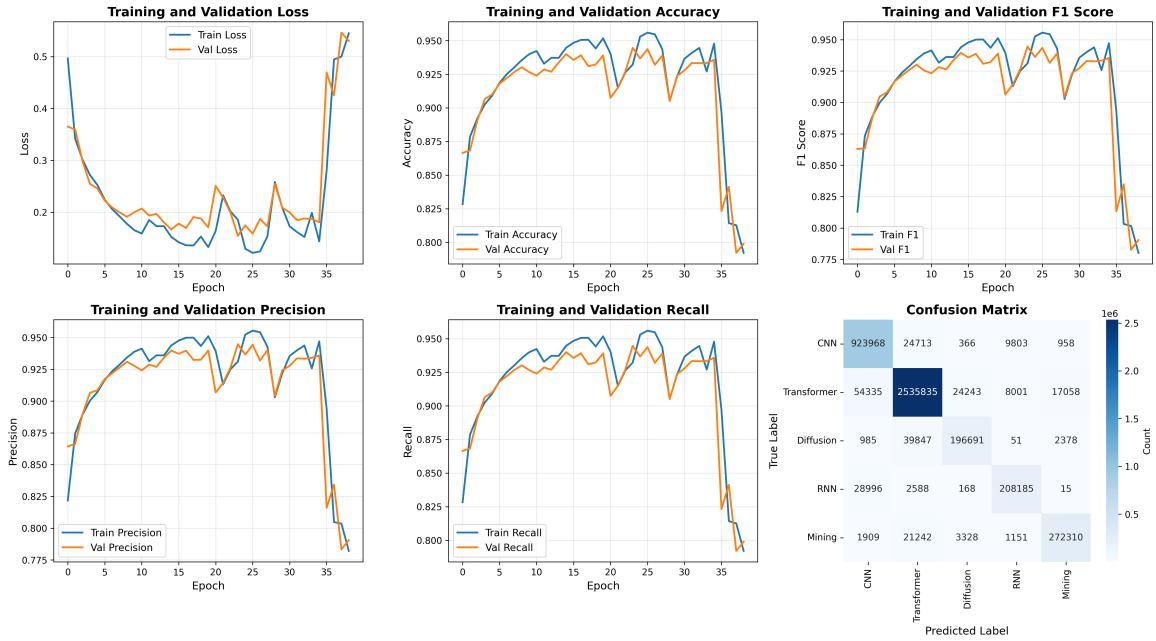

Figure 1: Training metrics and confusion matrix for the dual-stream BiLSTM classifier evaluated on all five workload categories on the RTX 4070.

Table 2: Model-Level Classification Across GPU Hardware Generations

Classifier Target	RTX 4070	RTX 3050	RTX 5070
Transformer Family	0.89	0.98	0.81
DenseNet vs. MobileNet	0.93	0.82	0.93
GRU vs. LSTM	0.96	0.93	0.89
SD v1.5 vs. tiny-sd	0.63	0.76	0.81

3.4. CNN Structural Parameter Regression

For the CNN domain, TEXAS utilizes a regression head over the collected timing traces to extract structural layer dimensions. The categorical Softmax layer is replaced with a linear activation head optimized via Mean Squared Error (MSE) loss. The network successfully predicts internal hyperparameters, specifically estimating the exact number of convolutional and pooling layers. On DenseNet-121 traces (whose true value is 120 convolutional layers), the mean prediction ranges between approximately 80 and 107 layers depending on the execution phase and dataset, yielding a mean absolute error of 27.9 layers (Figure 2). A similar experiment for pooling layer count achieves a mean absolute error of 1.6 pooling layers (where the true value is 5), proving that general architectural complexity leaks through resource contention.

4. Countermeasures

Mitigating the TEXAS framework requires evaluating software-level and hardware-enforced defense strategies inspired by traditional side-channel mitigation concepts:

1. **Timer Restriction:** Since the attack relies on microsecond-level timing differences, restricting or coarsening unprivileged user-space access to the high-resolution hardware cycle counter (`clock64()`) would significantly hinder information extraction [5].
2. **Resource Partitioning and QoS Enforcements:** Implementing explicit cache Quality of Service (QoS) guarantees or strict microarchitectural channel partitioning — similar to CPU cache allocation strategies [3, 4] — would isolate the texture caching layers and data transfer paths across concurrent processes.
3. **Noise Injection and Jitter:** Injecting randomized execution delays into the completion routines of the texture cache fabric or asynchronous copy pipelines breaks the correlation between victim execution states and observed spy latencies, lowering the side-channel signal-to-noise ratio.

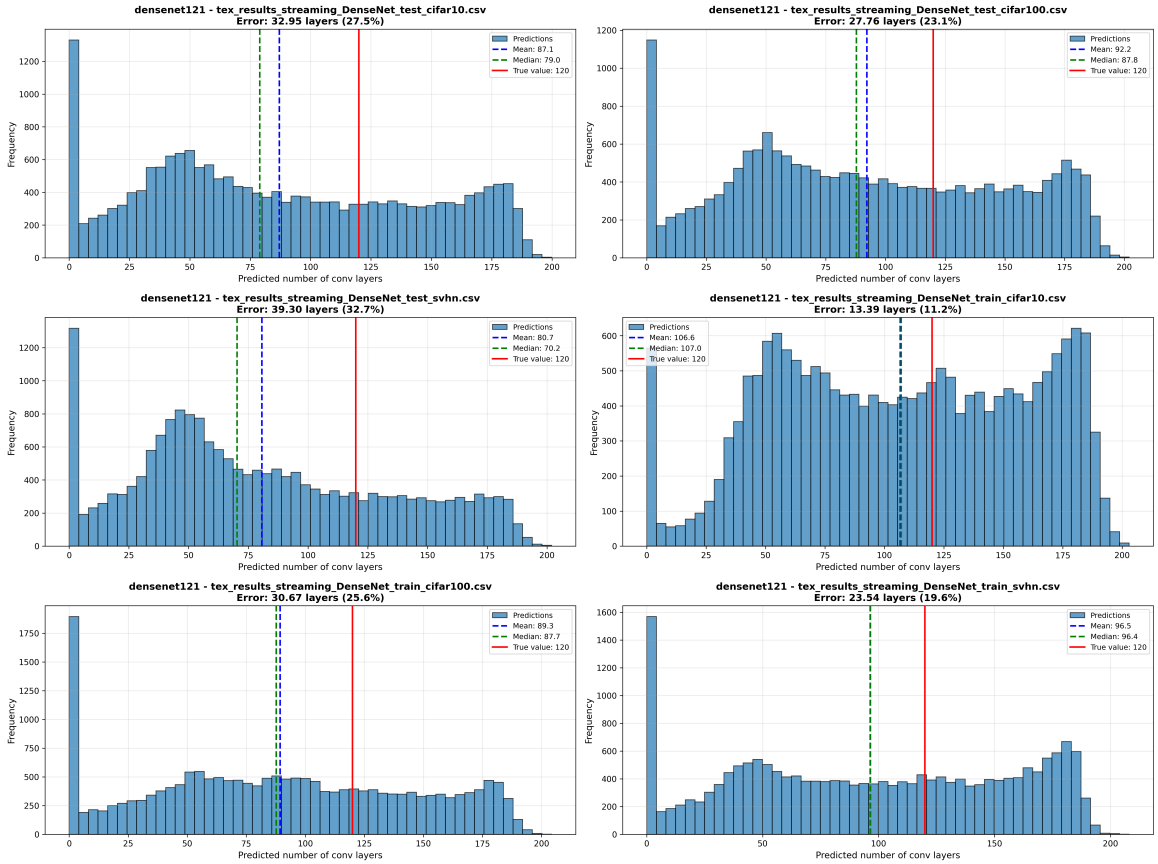

Figure 2: Distribution of predicted convolutional layer counts for DenseNet-121 across different datasets and execution phases.

5. Conclusions

This thesis has introduced TEXAS, a contention-based side-channel framework targeting the Texture and Asynchronous memory pipelines of modern NVIDIA GPUs. Operating entirely within unprivileged user spaces in bare-metal deployments, the work demonstrates that fusing independent microarchitectural signals provides substantially higher fingerprinting resolution than single-pipeline approaches.

The practical security implications are concrete: an unprivileged attacker sharing a research server or workstation can determine with high confidence which category of workload the victim is running, and in many cases identify the specific model family or architecture. This constitutes a realistic threat to the intellectual property of proprietary models deployed in shared environments, where model identity and structural complexity are typically considered confidential.

The hierarchical framework reliably identifies diverse execution workloads across multiple hardware generations, emphasizing the need for robust microarchitectural isolation mechanisms.

References

[1] Jaeguk Ahn, Jiho Kim, Hans Kasan, Leila Delshadtehrani, Wonjun Song, Ajay Joshi, and John Kim. Network-on-chip microarchitecture-based covert channel in gpus. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, page 565–577, New York, NY, USA, 2021. Association for Computing Machinery.

[2] Sankha Baran Dutta, Hoda Naghibijouybari, Arjun Gupta, Nael Abu-Ghazaleh, Andres Marquez, and Kevin Barker. Spy in the gpu-box: Covert and side channel attacks on multi-gpu systems. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ISCA '23, New York, NY, USA, 2023. Association for

Computing Machinery.

[3] Andrew Herdrich, Edwin Verplanke, Priya Autee, Ramesh Illikkal, Chris Gianos, Ronak Singhal, and Ravi Iyer. Cache qos: From concept to reality in the intel® xeon® processor e5-2600 v3 product family. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 657–668, 2016.

[4] Fangfei Liu, Qian Ge, Yuval Yarom, Frank Mckeen, Carlos Rozas, Gernot Heiser, and Ruby B. Lee. Catalyst: Defeating last-level cache side channel attacks in cloud computing. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 406–418, 2016.

[5] Weijie Liu, Debin Gao, and Michael K. Reiter. On-demand time blurring to support side-channel defense. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security – ESORICS 2017*, pages 210–228, Cham, 2017. Springer International Publishing.

[6] Yuanqing Miao, Yingtian Zhang, Dinghao Wu, Danfeng Zhang, Gang Tan, Rui Zhang, and Mahmut Taylan Kandemir. Veiled pathways: Investigating covert and side channels within gpu uncore. In *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1169–1183, 2024.

[7] Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh. Rendered insecure: Gpu side channel attacks are practical. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 2139–2153, New York, NY, USA, 2018. Association for Computing Machinery.

[8] NVIDIA Corporation. *CUDA C++ Programming Guide*, 2025. Accessed: March 2026.

[9] NVIDIA Corporation. *Parallel Thread Execution ISA Reference*, 2025. Accessed: March 2026.

[10] Dag Arne Osvik, Adi Shamir, and Eran Tromer. Cache attacks and countermeasures: The case of aes. In David Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006*, pages 1–20, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[11] Yuval Yarom and Katrina Falkner. FLUSH+RELOAD: A high resolution, low noise, l3 cache Side-Channel attack. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 719–732, San Diego, CA, August 2014. USENIX Association.

[12] Zhenkai Zhang, Tyler Allen, Fan Yao, Xing Gao, and Rong Ge. Tunnels for bootlegging: Fully reverse-engineering gpu tlbs for challenging isolation guarantees of nvidia mig. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 960–974, New York, NY, USA, 2023. Association for Computing Machinery.