"Energon": Unveiling Transformers from GPU Power and Thermal Side-Channels

Arunava Chaudhuri*, Shubhi Shukla†, Sarani Bhattacharya*, and Debdeep Mukhopadhyay*
*Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, Kharagpur, India
†Centre for Computational and Data Sciences, Indian Institute of Technology, Kharagpur, Kharagpur, India
{arunavachaudhuri392, shuklashubhi6, bhattacharya.sarani.iitkgp, debdeep.mukhopadhyay}@gmail.com
Accepted at ICCAD 2025

***Abstract*—Transformers have become the backbone of many Machine Learning (ML) applications, including language translation, summarization, and computer vision. As these models are increasingly deployed in shared Graphics Processing Unit (GPU) environments via Machine Learning as a Service (MLaaS), concerns around their security grow. In particular, the risk of side-channel attacks that reveal architectural details without physical access remains underexplored, despite the high value of the proprietary models they target. This work to the best of our knowledge is the first to investigate GPU power and thermal fluctuations as side-channels and further exploit them to extract information from pre-trained transformer models. The proposed analysis shows how these side channels can be exploited at user-privilege to reveal critical architectural details such as encoder/decoder layer and attention head for both language and vision transformers. We demonstrate the practical impact by evaluating multiple language and vision pre-trained transformers which are publicly available. Through extensive experimental evaluations, we demonstrate that the attack model achieves a high accuracy of over 89% on average for model family identification and 100% for hyperparameter classification, in both single-process as well as noisy multi-process scenarios. Moreover, by leveraging the extracted architectural information, we demonstrate highly effective black-box transfer adversarial attacks with an average success rate exceeding 93%, underscoring the security risks posed by GPU side-channel leakage in deployed transformer models.**

***Index Terms*—side-channel, transformer, model stealing, GPU.**

I. Introduction

Transformer-based models like Bidirectional Encoder Representations from Transformers (BERT), Large Language Models (LLMs), and Vision Transformers (ViTs) have significantly advanced natural language processing (NLP) and computer vision by capturing complex patterns in large datasets.Their high computational demands have driven companies like NVIDIA to develop specialized GPUs, with the growing reliance on such hardware reflected in NVIDIA's rising stock prices. Despite this growing dependence on transformers and GPUs, the security of both against side-channel attacks remains limited and largely unexplored. In particular, there is currently no work examining the potential for model extraction or model-stealing attacks on transformers executed on GPUs through side-channel vulnerabilities, highlighting a critical gap in security research as these models continue to proliferate.

Model extraction or *model-stealing* attacks aim to replicate or gain insight into a target deep learning model's architecture or parameters without having direct access to it. Broadly, two types of model-stealing attacks have been studied. The first is the *query-based model-stealing attack*, where attackers exploit a model's prediction Application Programming Interface (API) to clone or replicate it, without accessing its parameters or training data. Query-based attacks have exposed vulnerabilities in models like DNNs and CNNs [1]–[4], and recent work has even extracted embedding layers from black-box transformer models such as OpenAI's ChatGPT using API access alone [5].

The second type of attack, *side-channel-based model-stealing*, exploits leakages such as power, thermal, electromagnetic emissions, and microarchitectural behaviors (e.g., cache accesses, branch misses) to infer model architecture. Unlike query-based methods, these attacks leverage shared hardware resources without directly interacting with the model API. Side-channel attacks use techniques which include cache-based channels [6]–[8], and physical leakages like power [9], thermal [10], electromagnetic emanations [11]–[13], and off-chip memory access [14]. Prior GPU-based efforts have focused on extracting DNN/CNN architectures using CUPTI counters [15] (currently not accessible) and resource-tracking APIs via CUDA-based spy applications [16]. *However, no prior work has explored GPU side-channels specifically for transformer model computations.*

In this work, we address this gap by exploring GPU side-channels for transformer models, focusing specifically on power and thermal channels that remain accessible without special privileges and cannot be virtualized, even within virtualized GPU environments. Our work specifically targets *NVIDIA GPUs*, which currently holds 88% of the GPU market share, though the approach can readily be extended to other GPUs. In

the literature several side-channel attacks are performed on older generation of NVIDIA GPUs, including power and timing side-channel attack on Kepler architecture [17], [18], as well as timing side-channel attack on Volta [19], [20] and Maxwell architecture [21]. However, post-Pascal generations of NVIDIA GPUs have undergone significant architectural changes, most importantly with the introduction of Multi-Instance GPU (MIG) (discussed in Section IV) technology in the Ampere generation for production-grade GPUs, enables better resource distribution and improved utilization in cloud environments. Furthermore, there are not much documented side-channel experiments on post-Pascal generations of NVIDIA GPUs in the literature. *In our experiment, we focus on newer generation of NVIDIA GPUs, specifically those based on the Turing and Ampere architectures.*

Despite advancements in underlying hardware of newer generation of NVIDIA GPUs (Turing, Ampere, Ada Lovelace), to the best of our knowledge, no GPU manufacturer offers protections against these side-channels. Our initial experiments show that transformers produce a distinctive *staircase-like* pattern in side-channel traces, far more distinctive than those in earlier DNNs like CNNs. This pattern enables accurate inference of encoder-decoder layers from a single trace. We train CNNs on these side-channels to predict transformer's architectural details, achieving 95.00% and 91.25% accuracy in classifying language and vision model families, respectively. In summary, the contributions of this work are as follows:

- We identify power and thermal traces as potential side-channels that reveal architectural details of transformer models and develop CNN-based predictors to infer attributes like encoder-decoder layers and attention heads.
- Moreover, to showcase the broad applicability of the observed GPU side-channels, we develop predictive models to identify the architectural families of 8 publicly available pre-trained language models and 8 vision models.
- We evaluate our attack against a noisy, multi-process environment and still achieve over 89% accuracy in predicting black-box transformer model's parameters.
- We demonstrate that partial architectural leakage enables building a substitute language transformer model with a BERT Score over 93, and achieves 93% success in black-box transfer attacks on Vision Transformers.

Responsible Disclosure: We have responsibly disclosed the power and thermal side-channel vulnerability to NVIDIA. NVIDIA offered to acknowledge us on product security page stating that telemetry should be disabled for more security-sensitive use cases such as Confidential Compute (CC) and MIG, and mentioned they are cautiously evaluating how much more telemetry can be safely exposed. They also approved public disclosure of our findings.

This paper is organized as follows: Section II presents background on transformer models. Section III analyzes GPU power and thermal side-channels during transformer inference. Section IV outlines our methodology for extracting architectural information and presents experimental results. Sections V and VI cover a black-box transfer adversarial attack and discuss about some of the related artifacts respectively. Finally, Section VII concludes with possible mitigations.

II. Background on Transformers

Introduced by Google Brain in their paper *Attention Is All You Need* [22], the transformer has become a state-of-the-art approach in natural language processing, surpassing earlier models like recurrent neural networks (RNNs) and long short-term memory networks (LSTMs). Unlike these models, which struggle with long sequences and slow training speeds, the transformer replaces recurrence with a fully attention-based mechanism, enabling efficient parallel processing of entire sequences. In the following, we briefly discuss about its core architectural modules.

Transformer Encoder: In a transformer, the encoder processes the input sequence to generate a representation that captures contextual relationships among tokens. It consists of multiple layers stacked together, each receiving embedded input plus positional embeddings to retain token positions. Each layer includes a multi-head attention mechanism and a feed-forward network, both with residual connections and followed by layer normalization. In the multi-head attention layer, attention scores are calculated using query, key, and value vectors from input embeddings. The query-key dot product, passed through a softmax, yields attention outputs multiplied by the value vector and sent to the feed-forward network. The final encoder output is passed to the decoder layers.

Transformer Decoder: The decoder generates the output sequence using the encoder's representation and previously generated tokens. Each decoder layer mirrors the encoder's structure but includes masked self-attention to prevent future tokens from influencing the current token. The final layer outputs a translated word, which is added to the decoder input to continue the process. This cycle repeats until an end-of-sequence token is predicted, enabling it to handle sequence-to-sequence tasks.

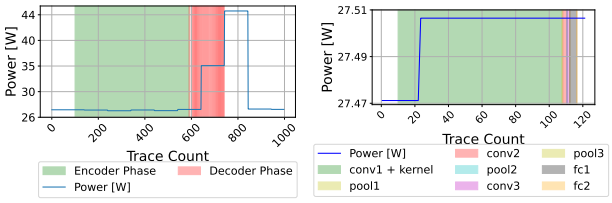
Self-Attention: In addition to the encoder and decoder, the transformer model uses self-attention, allowing each token in a sequence to focus on others, capturing contextual relationships regardless of position. Multi-head attention achieves this, with each head learning

different aspects of token relationships. In the encoder, self-attention gives each token access to all tokens in the input, while in the decoder, it is masked to prevent future tokens from influencing the current token.

Each of the transformer's components impose distinct computational demands, shaping unique power and thermal profiles. We aim to exploit these patterns to uncover architectural insights via side-channel analysis.

III. Power and Thermal Foot prints of Transformer

Power and thermal side-channels have been extensively used to extract secure information such as cryptographic keys [23], plaintext data [24], confidential algorithms [25], and more from various computational systems. Recently, these side-channels have also been applied to deep learning models, revealing private information including architectural details [26], model weights [27], and training hyperparameters [28] when executed on CPUs and edge devices. Traditionally, side-channel attacks and model fingerprinting techniques have focused primarily on simpler DNN architectures like CNNs. However, as complex and widely-used transformer-based architectures emerge, they remain relatively unexplored in terms of security and side-channel vulnerabilities. This work aims to address this gap by investigating side-channel-based attacks specifically targeting transformer models, highlighting potential security risks. An important aspect of our work is the application of side-channel analysis on GPUs, as transformers are frequently hosted on large data center GPUs to enable faster computation and support larger model sizes.



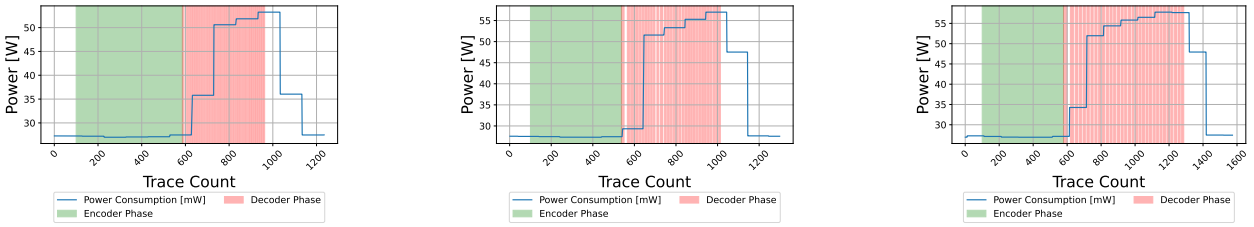
(a) Transformer Model (b) CNN Model

Fig. 1: Power Trace of a custom CNN and transformer model with one encoder/decoder layer and eight attention heads during single inference.

A. *Power Gradient Analysis for Transformer Models*

We aim to explore the impact of transformer model execution on GPU power consumption and as an initial step, we built a custom language transformer model. Our first model was trained with a configuration of one encoder and one decoder with eight attention heads. We set the embedding vector size to 512, and the maximum sentence length was limited to 200. The power trace was collected with user privileges using `nvidia-smi` query with GPU temperature stabilized at 28°C. Our goal was to collect data at a sampling rate of 10 Hz or 100 Hz, but the recorded trace length was extremely short due to faster model inference, and no distinguishable trends were observed for transformer execution. To address this, we aimed to set the sampling rate at 1 Hz to increase the trace length but had to settle for 7 Hz, as it was the minimum achievable rate. Additionally, at high temperatures, the GPU consumes more power, leading to erroneous information for transformer model inference. To mitigate this, we collected all traces while keeping the GPU temperature stable and allowed the GPU to cool down before the next execution cycle begins, ensuring that previously collected traces do not influence subsequent ones. Consequently, the overall trace collection time was increased due to this additional waiting time required for the GPU to cool down. This approach is effective as long as the GPU caches and memory are cleared of the prior process's data elements. Such a methodology is commonly employed for reliable data collection in side channel analysis. In real-world scenarios for similar analyses, additional GPU cooling would not be necessary, as traces are typically gathered from the same process running continuously for inference workloads.

Using the trained model, we performed an inference on a consumer-grade GPU (GTX 1660 Ti Mobile GPU) and plotted its power trace data, as shown in Figure 1a. The first green-shaded zone in the figure represents power usage during encoder execution, while the consecutive red-colored slender areas indicate the decoder execution as it generates the translated output for each word in the input phrase. Here, we observe a staircase-like pattern emerging during transformer model inference. Notably, the power consumption pattern differs significantly between the encoder and decoder phases: during encoder execution, power consumption rises to 6W, whereas, during decoder execution, power demand increases to 20W. *This happens because, during the execution of the encoder, it processes a fixed-length input sequence at a time to create the final attention matrix, which is then used as input to the decoder. This explains the initial small increase in the power graph before it stabilizes. In contrast, the decoder operates on a gradually increasing input sequence to generate each translated word. As the input size grows, the computation becomes more complex, leading to a steady step-wise increase in power consumption until the final translated word is generated.* For comparison, we also present power readings for a custom CNN model in Figure 1b and observe no distinctive pattern across its layers, unlike the unique patterns seen in transformers in Figure 1a. Instead, the CNN exhibits a steady power reading of approximately 27.5 W throughout execution. Consequently, in this work, we specifically target transformer



(a) Encoder/ Decoder: 3 (b) Encoder/ Decoder: 5 (c) Encoder/ Decoder: 8

Fig. 2: Power Trace of single transformer Model inference with varying encoder/decoder and eight attention heads.

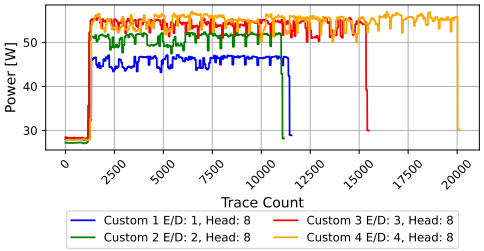


Fig. 3: GPU Power Trace of Transformer Model during 100 Inferences on NVIDIA GTX 1660 Ti (E/D: Encoder/Decoder).

architectures due to their uniquely observable patterns.

Following the observed results, we extended our analysis to larger transformer models to see if similar behavior could be replicated. We created and trained four additional transformer models with varying encoder-decoder configurations. Next, we collected power trace data for each model running on the same GPU, as shown in Figure 2. The power trace data reveals a clear relationship between power consumption and transformer size (number of encoder/decoder blocks). As the number of encoder/decoder blocks increases, so does the power consumption during execution. When the transformer's decoder becomes active during inference, power consumption increases to as much as 30 W across nearly all model inferences. Additionally, we observed that the number of staircase steps in the power trace grows proportionally with the number of encoder/decoder blocks in the transformer model.

B. *Power Side-channel on Data Center GPUs*

With above knowledge and understanding of transformer power trace characteristics, we extend our experiments on a larger scale with practical use cases using server-grade GPUs. For this, we replicate the scenario of transformer models running on a data center GPU, where multiple batch inferences are typically executed concurrently. To simulate this, we collect power trace data from our custom transformer models by performing 100 continuous inferences on each model. In Figure 3, we observe a clear difference in power consumption across each configuration of the transformer model.

To further demonstrate the impact of this side-channel, we built and trained four custom models with configurations similar to those of popular pre-trained language models on Hugging Face, as shown in Table I. This time, we extended our observations to data center GPUs commonly used for running large transformer models in various AI tasks. Therefore, all subsequent experiments are conducted on the NVIDIA A40 GPU. We collected power trace data for 120 seconds while running inference with our previously trained custom models. In this setup, we observed a similar pattern in Figure 5a (Custom Transformer Power Trace) as in our earlier experiment (Figure 3), though power consumption differences are significantly higher, while differences between models remain apparent. Additionally, to gain insights into practical applications, we collected GPU power trace data while running inference on popular pre-trained models from Hugging Face. In Figure 5a (Pre-trained Transformer Power Trace), we observe distinct power traces for each transformer model, depending on model size (number of encoder/decoder layers and attention heads). Each model exhibits a unique power consumption pattern that can aid in its identification when executing on GPUs.

C. *Thermal Side-channel on Data Center GPUs*

Given the observed variations in power consumption based on the size and complexity of transformer models on GPUs, we anticipate similar effects on other side-channel metrics, such as thermal data. To verify this, we conducted an experiment similar to the previous one, collecting temperature data over 120 seconds during transformer model inference. In the resulting temperature trace graph (Figure 5b), we observe a stepwise, gradual increase in temperature throughout the inference duration. This indicates that the rate of temperature rise correlates with the transformer model's size on the GPU, similar to our findings from the power traces.

Takeaway *: Based on the above observations, we conclude that each transformer model exhibits distinct temperature and power consumption patterns, which can uniquely differentiate them from other models running on consumer or production-grade GPUs.*

TABLE I: Popular Pre-Trained Language Transformer Configurations

Family	Transformer Name	#E/D	#Attention Heads	Embedding Dimension
T5	t5-small	6 / 6	8	512
T5	t5-base	12 / 12	12	768
T5	t5-large	24 / 24	16	1024
T5	t5-3b	24 / 24	32	1024
MarianMT	Helsinki-NLP/opus-mt-en-fr	6 / 6	8	512
META	facebook/nllb-200-distilled-600M	12 / 12	16	1024
META	facebook/nllb-200-distilled-1.3B	24 / 24	16	1024
Google	madlad400-3b-mt	32 / 32	16	1024

```
+-----------------------------------------+
| GPU  Name                 Persistence-M |
| Fan  Temp   Perf          Pwr:Usage/Cap |
|                                         |
|=========================================+
|   0  NVIDIA A40                     Off |
|  0%   42C    P0             79W / 300W |
|                                         |
+-----------------------------------------+
```

Fig. 4: `nvidia-smi` query output displaying power and thermal data for NVIDIA A40 GPU.

IV. Extracting Transformer Architecture Using GPU Side-channels

Building on observations from the previous section, we now aim to leverage power and thermal side-channels to extract architectural details of well-known transformer models, including both language and vision models. To proceed, we first define our threat model.

A. *Threat Model:*

We consider a scenario with multiple users on a remote cloud server. To provide single-GPU access to multiple clients, cloud providers use GPU virtualization technologies like NVIDIA vCS combined with Multi-Instance GPU (MIG) backend technology, in contrast to traditional time-shared setups. Modern cloud providers like AWS and Google Cloud offer dedicated GPU instances per VM, allowing multiple users to run tasks independently. According to NVIDIA's MIG documentation, each GPU instance has isolated compute, memory, and bandwidth resources, minimizing interference and context-switching issues. In addition, users can access power and thermal metrics via tools like *nvidia-smi* and *pynvml* without root privilege. These metrics are directly obtained from physical GPU sensors and cannot be virtualized, even with MIG.

In this setup, a victim runs an unknown model on one virtualized GPU instance, while an adversary in a separate VM tries to infer victim model's architecture through other GPU instances. Below, we outline the adversary's capabilities and objectives in detail:

Adversary's capabilities : The adversary can continuously monitor thermal and power data for its assigned virtual GPU using tools like `nvidia-smi` (Figure 4) or the Python library `pynvml`. *Notably, the adversary does not need sudo/root privileges to access this information.*

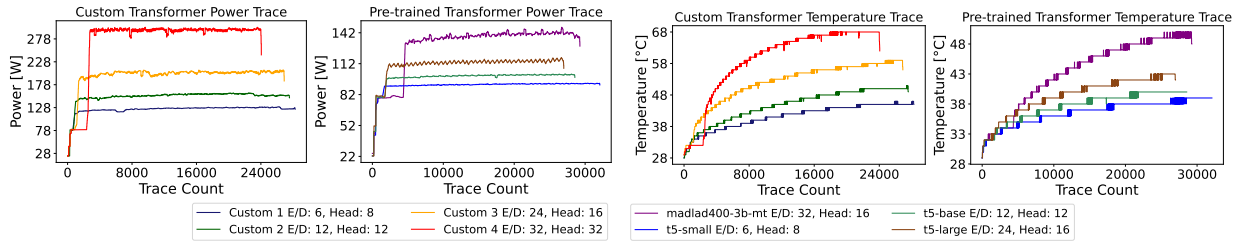
Adversary's Objective : The adversary's goal is to extract architectural information about the transformer model running on the victim's VM by monitoring power and thermal data from their own VM using the tools mentioned above.

B. *Transformer Architecture Extraction*

We now aim to demonstrate the process of building a prediction model to extract key architectural parameters of transformer models based on our defined threat model. We present results using our custom models, pre-trained language and vision transformers from Hugging Face, aiming to extract key architectural details such as the number of attention heads, and encoder/decoder layers. Both power and thermal traces are utilized as inputs to the prediction model. These traces were collected over 120 seconds at a sampling rate of 7 Hz, with an initial GPU base temperature of 28°C. In total, we gathered 100 traces for each model in our work, splitting the data into an 80:20 ratio for training and testing the prediction model, and utilized stratified *K-fold* cross-validation for building and validating the model. For clarity, transformer configurations are denoted as X/Y, where X represents the number of encoder-decoder layers and Y the number of attention heads.

*1) **Custom Language Transformer**:* We created and trained a total of nine custom transformer models with the following configurations: 6/8, 12/8, 12/12, 12/16, 24/16, 32/16, 48/16, 24/32 and 32/32. Our objective is to build machine learning models capable of predicting hyperparameter of an unknown transformer model, using power and thermal traces as inputs. This is done in two steps: first classifying number of attention heads, followed by predicting encoder/decoder counts. To achieve this, we built a prediction model consisting of three convolutional layers with sizes 32, 16, and 8, each followed by batch normalization and *ReLU* activation. The model also includes two max-pooling layers and two fully connected layers with a *softmax* activation function. The *Adam* optimizer with a learning rate of 0.00001 was used for training. The first model was trained for attention head classification, achieving 100% accuracy (Figure 6a) on test sets using both thermal and power traces. Subsequently, the prediction model further sub-categorized each attention head family based on the number of layers. The attention head count of 8 (two sub-classes) and 32 (two sub-classes) were classified with 100% accuracy (Figure 6b and 6d), while the 16 attention head model (four sub-classes) achieved 96.25% accuracy (Figure 6c).

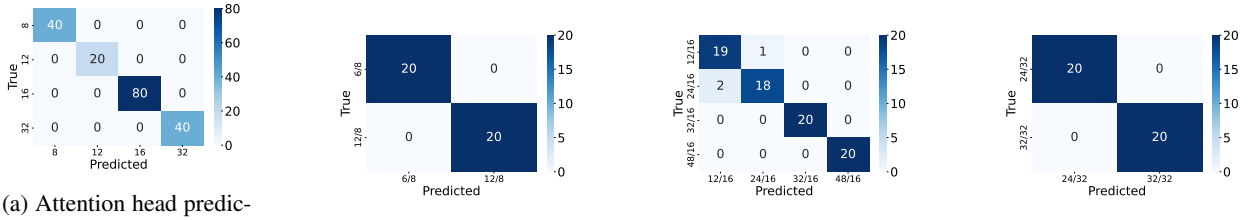
*2) **Pre-trained Language Model**:* Additionally, we present power/thermal-based prediction results for four pre-trained language transformer families: *T5, Google, MarianMT*, and *META*. As shown in Table I, *T5* family



(a) Power Traces

(b) Thermal Traces

Fig. 5: Power and thermal traces of transformers on NVIDIA A40 during 120s inference (E/D: Encoder/Decoder)



(a) Attention head prediction

(b) Attention head 8

(c) Attention head 16

(d) Attention head 32

Fig. 6: Confusion matrices for attention head and encoder/decoder layer prediction for attention head families.

TABLE II: Popular Pre-Trained Vision Transformer Basic Configuration (E/D: Encoder/Decoder)

Family	Transformer Name	#E/D	#Attention Heads	Embedding Dimension	Input Image Size
Google	vit-base-patch16-224	12	12	768	224×224
Google	vit-large-patch16-225	24	16	1024	224×224
Apple	mobilevit-small	12	4	384	256×256
META	deit-tiny-distilled-patch16-224	12	3	192	224×224
META	deit-small-distilled-patch16-224	12	6	384	224×224
META	deit-base-distilled-patch16-224	12	12	768	224×224
Microsoft	swin-tiny-patch4-window7-224	12	3	96	224×224
Microsoft	swin-base-patch4-window7-224	12	12	768	224×224

includes four models, while *Meta* has two, and *Google* and *MarianMT* each contributes one model. Unlike custom transformers, we consider eight pre-trained models across four research labs. Consequently, our first objective is to identify the target model's family, followed by classification of architectural details such as attention heads and encoder/decoder layers. This is feasible as models within the same family exhibit similar power and thermal patterns due to shared architectures and training methodologies. Following this plan, we constructed our first prediction model to classify the families of pre-trained models using the same CNN architecture as before, achieving up to 95% accuracy and an average of 89% across five folds. To identify additional architectural details, we focus on two families: *T5* and *Meta*, developing separate prediction models for each of them.

TABLE III: Pre-Trained Language Transformer Architecture Prediction Accuracy (Thermal & Power Traces)

Transformer Family	Prediction Criteria	Max Acc.(%)	Avg. Acc.(%)
All Models	Root Family	95%	89%
META	Encoder/Decoder	100%	100%
T5	Attention Head	100%	100%

As shown in Table I, the *T5* family models differ only in attention heads and layers, so we train a model to distinguish between the four variants based on attention heads. In contrast, the models from *Meta* family share the same number of attention heads but vary in the number of encoder/decoder and total layers. Therefore, we build a predictive model for encoder/decoder layers to differentiate between the two possible models in this family. As shown in Table III, both the *T5* and *Meta* prediction models achieve 100% accuracy in identifying attention heads and encoder/decoder layers, respectively, from the test samples.

*3) **Pre-trained ViT Model**:* We further investigate popular ViT models from several renowned research labs on Hugging Face (refer to Table II). Unlike language transformers, which feature an encoder-decoder structure, ViT models utilize a simpler encoder-only architecture. These models are also smaller and less complex than other transformer variants, allowing efficient execution on lower-end consumer GPUs. Most ViT models consist of 12 layers, with 3 to 16 attention heads, and variable embedding dimensions depending on the model's size. Nearly all the models studied accept input images of size 224×224. For our experiments, the input to these transformers is derived from the CIFAR-10 dataset [29], which includes 60,000 color images of size 32×32, distributed across 10 classes.

TABLE IV: Pre-Trained Vision Transformer's Architecture Prediction Accuracy

Transformer Family	Prediction Criteria	Max Acc.(%)	Avg. Acc.(%)
All Models	Root Family	91.25%	84.75%
META	Attention Head	100%	100%
Google	Attention Head	100%	100%
Microsoft	Attention Head	100%	100%

Architectural differences in ViT models are shaped by attention heads, embedding dimensions, optimization methods, and tokenization strategies. We focus on four model families: *Apple, Google, Facebook*, and *Microsoft*. ViT models from *Google* vary in the number of attention heads, encoders/decoders, and embedding dimensions, while those from *Facebook* and *Microsoft* differ mainly in attention heads and embedding sizes. Based on these distinctions, our initial prediction model classifies the models by their respective families followed by family-specific models that predict attention head counts. Following the same approach, our first prediction model (using the same CNN architecture as for custom transformer model prediction) for ViT models achieved a maximum accuracy of 91.25% with an average of 84.75% across five folds. Additionally, as shown in Table IV, prediction models for remaining three families achieve 100% accuracy in predicting attention head counts using both thermal and power traces.

C. *Model Extraction in Noisy Environment:*

We have further evaluated the practicality of our approach for real-world scenarios where multiple processes run concurrently on the GPU.

*1) **Model Extraction Against Variable Number of Background Process**:* In server-grade GPUs, we focus on scenarios where a cloud GPU is partitioned into more than one independent GPU instances, with multiple processes are running independently within each instances. Under this setup, we executed transformer model inference and matrix multiplication in parallel and collected corresponding power and thermal traces. As shown in Figure 7, despite the added workload noise, our classification model achieved an average accuracy of 89.25% for identifying the transformer model family, and 92% and 87% accuracy in distinguishing between *META* and *T5* architectures—closely aligning with results from the ideal scenario. We further extended our experiments to include three and four concurrent processes, observing only a minimal drop in performance across tasks such as model family and attention head identification. Still, these results confirm that even in complex, noisy environments with multiple simultaneous processes, our attack model can reliably extract architectural details of black-box transformer models.

*2) **Model Extraction Against Different Categories of Background Process**:* Additionally, we also want

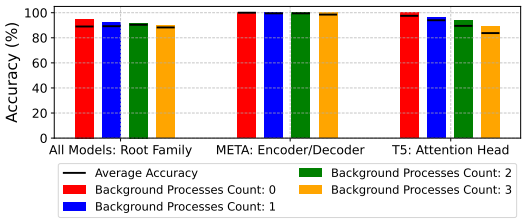


Fig. 7: Prediction Model performance with increasing number of background process: Maximum Accuracy with Average as Dashed Line.

to verify our attack model against different noisy environments. In a realistic scenario, MIG instances often handle various categories of machine learning tasks simultaneously for specific users. To replicate this environment, we ran a transformer inference workload in parallel with large matrix multiplication, CNN-based image classification, and vision transformer inference, in three independent scenarios. The inclusion of these kind of resource hungry tasks allowed us to assess how well our generalized prediction model performs in a complex, high-noise setting where GPU utilization is consistently high across multiple instances on a single physical device. Under this setting, we observe (Figure 8) a 4% drop in root family identification accuracy when large matrix multiplication runs in parallel. However, this scenario outperformed the others in detecting *META's* encoder/decoder and *T5's* attention heads. Although performance varied across tasks in different scenarios, our prediction model consistently identified key architectural parameters of transformer models with high accuracy, even in noisy, MIG-enabled GPU environments.

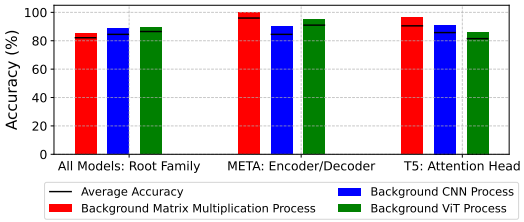


Fig. 8: Prediction Model performance with different types of background process: Maximum Accuracy with Average as Dashed Line.

V. Exploitation of Extracted Transformer Architecture

In this section, we show how an adversary can exploit leaked architectural information—obtained through thermal and power side-channel analysis—to compromise a black-box transformer model. We highlight two possible consequences of such leakage. First, an adversary could build substitute models that closely mimic the behavior of the original model and use them for their own purposes. Second, the leaked information could enable more

effective black-box transfer adversarial attacks, where adversarial examples crafted using a similar surrogate model are used to mislead the target model into making incorrect predictions.

A. *Creation of Substitute Model*

In our first experiment, we built a substitute language transformer model replicating the MarianMT Helsinki architecture using side-channel-extracted details such as the number of encoders/decoders and attention heads. This model achieved a BLEU[1] score of 44.12 on the test set, indicating high-quality translations. As shown in Table V, it also attained a BERT[2] score of 93.53, closely matching the original model's outputs when architectures aligned. These findings highlight that even partial architectural leakage can enable the creation of a shadow model that closely replicates the performance and outputs of the target model.

TABLE V: Substitute Language Transformer's Accuracy on Different Unknown Parameter Settings

Target Model	Substitute Model	Extracted Parameters	Unknown Parameters		BERT Score
			Embedding Dimension	Feed-forward Network	
Helsinki-NLP/ opus-mt-de-en	Helsinki-NLP/ opus-mt-de-en	Encoder/Decoder: 6 Attention Head: 8	512	512	93.5
			512	1024	93.08
			512	2048	**93.53**

B. *Black Box Transfer Adversarial Attack*

We further evaluated the implications of architectural leakage by conducting black-box transfer adversarial attacks on two target Vision Transformer (ViT) models from *Google* family. For these models, we constructed substitute models using architectural information extracted via side channels and generated adversarial examples using the FGSM [30] and PGD [31] methods, evaluating them against the corresponding target models. As shown in Table VI, both FGSM and PGD achieved an average attack success rate of 93% for both substitute model architectures, indicating strong similarity with the target models. This highlights how leaked architectural information can enable highly effective adversarial attacks in black-box settings.

TABLE VI: Pre-Trained Vision Transformer's Classification Accuracy in Adversarial Attack Scenarios

Model	Attack	Success Rate (%)
Google/vit-base-patch16-224	FGSM	83.38
	PGD	94.87
Google/vit-large-patch16-224	FGSM	98.63
	PGD	98.63

[1]BLEU score is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another.

[2]BERT score is an evaluation metric that compares candidate and reference sentences based on cosine similarity of contextual embeddings.

VI. Discussion

Our work goes beyond previous methods that mainly focused on CNNs using CPU side-channels or CUPTI-based GPU profiling. Transformers, unlike CNNs, have a more modular design with components like encoder-decoder blocks and attention heads, which create more visible patterns in power and thermal traces during inference. Moreover, while Transformers are typically larger than CNNs and may introduce more computational noise, their modular structure—such as repeated encoder/decoder blocks—creates stronger and more consistent patterns. This makes power and thermal side-channels more effective for detecting their behavior on GPUs even in noisy environments. Moreover, as CUPTI counters are often disabled or require root access on newer-generation GPUs, prior GPU-based techniques are becoming less practical. Power and thermal metrics, however, remain accessible through simple queries without elevated privileges. This makes our approach the first to successfully extract architectural information from transformer models on modern NVIDIA GPUs using side-channel signals alone.

We also observe that the most commonly downloaded transformers on platforms like Hugging Face share a core set of architectural parameters—such as the number of layers, attention heads, and intermediate sizes—which we summarize in Table I and II of our paper. While differences may exist in datasets, embedding dimensions, or optimization methods, the core structure remains consistent. Therefore, our classification models are trained specifically to detect these known architectural patterns.

VII. Mitigation and Conclusion

Power and thermal metrics are essential for GPU stability but can also leak sensitive model details. To mitigate this, we propose strategies on two fronts. On the *application side*, combining model *model pruning* [32] and *knowledge distillation* [33] can generate obfuscated transformer variants with similar functionality, allowing random selection during inference to conceal architectural details. On the *hardware or system side*, GPU vendors can restrict sensor access to admin users, exposing only key metrics like utilization and memory usage. In cases of high power or temperature, instance users can receive warnings, with minor throttling to maintain stability while limiting side-channel exposure.

In conclusion, this work investigates power and thermal side-channels as a novel approach to extract architectural details of transformer models on NVIDIA GPUs. These side-channels remains accessible without special privileges and cannot be virtualized, posing serious security risks. Using custom prediction models, we achieved close to 100% accuracy in identifying key parameters of both custom and pre-trained transformers.

For language models like *T5* and *Meta*, we accurately classified encoder/decoder layers and attention heads, while for vision transformers from *Google*, *Facebook*, and *Microsoft*, we achieved 100% accuracy in predicting attention head counts. Building on this extracted information, we successfully created substitute language transformer model with a BERT Score over 93, and achieves 93% success in black-box transfer attacks on Vision Transformers, demonstrating the practical implications of such side-channel vulnerabilities. This highlights the urgent need to address side-channel threats to safeguard model architectures on GPUs.

REFERENCES

[1] D. DeFazio and A. Ramesh, "Adversarial model extraction on graph neural networks," *CoRR*, vol. abs/1912.07721, 2019.

[2] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, "Stealing links from graph neural networks," in *30th USENIX Security Symposium*, pp. 2669–2686, USENIX Association, Aug. 2021.

[3] S. Kariyappa, A. Prakash, and M. K. Qureshi, "Maze: Data-free model stealing attack using zeroth-order gradient estimation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13809–13818, 2021.

[4] K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, and M. Iyyer, "Thieves on sesame street! model extraction of bert-based apis," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[5] N. Carlini, D. Paleka, K. D. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy, E. Wallace, D. Rolnick, and F. Tramèr, "Stealing part of a production language model," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, OpenReview.net, 2024.

[6] M. Yan, C. W. Fletcher, and J. Torrellas, "Cache telepathy: Leveraging shared resource attacks to learn DNN architectures," in *29th USENIX Security Symposium*, pp. 2003–2020, USENIX Association, Aug. 2020.

[7] S. Hong, M. Davinroy, Y. Kaya, S. N. Locke, I. Rackow, K. Kulda, D. Dachman-Soled, and T. Dumitraş, "Security analysis of deep neural networks operating in the presence of cache side-channel attacks," 2020.

[8] Y. Liu and A. Srivastava, "Ganred: Gan-based reverse engineering of dnns via cache side-channel," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, CCSW'20, (New York, NY, USA), p. 41–52, Association for Computing Machinery, 2020.

[9] K. Yoshida, T. Kubota, S. Okura, M. Shiozaki, and T. Fujino, "Model reverse-engineering attack using correlation power analysis against systolic array based neural network accelerator," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2020.

[10] N. Mishra, T. L. Dutta, S. Shukla, A. Chakraborty, and D. Mukhopadhyay, "Too hot to handle: Novel thermal side-channel in power attack-protected intel processors," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 378–382, 2024.

[11] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel," in *28th USENIX Security Symposium*, (Santa Clara, CA), pp. 515–532, USENIX Association, Aug. 2019.

[12] L. Chmielewski and L. Weissbart, "On reverse engineering neural network implementation on GPU," in *Applied Cryptography and Network Security Workshops - ACNS, Kamakura, Japan, June 21-24, 2021, Proceedings*, vol. 12809, pp. 96–113, Springer, 2021.

[13] H. Yu, H. Ma, K. Yang, Y. Zhao, and Y. Jin, "Deepem: Deep neural networks model recovery through em side-channel information leakage," in *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 209–218, 2020.

[14] W. Hua, Z. Zhang, and G. E. Suh, "Reverse engineering convolutional neural networks through side-channel information leaks," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2018.

[15] J. Wei, Y. Zhang, Z. Zhou, Z. Li, and M. A. A. Faruque, "Leaky DNN: stealing deep-learning model secret with GPU context-switching side-channel," in *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain*, pp. 125–137, IEEE, 2020.

[16] H. Naghibijouybari, A. Neupane, Z. Qian, and N. 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, (New York, NY, USA), p. 2139–2153, Association for Computing Machinery, 2018.

[17] C. Luo, Y. Fei, P. Luo, S. Mukherjee, and D. Kaeli, "Side-channel power analysis of a gpu aes implementation," in *33rd IEEE International Conference on Computer Design (ICCD)*, pp. 281–288, 2015.

[18] Z. H. Jiang, Y. Fei, and D. Kaeli, "A novel side-channel timing attack on gpus," in *Proceedings of the Great Lakes Symposium on VLSI 2017*, GLSVLSI '17, (New York, NY, USA), p. 167–172, Association for Computing Machinery, 2017.

[19] H. Naghibijouybari, A. Neupane, Z. Qian, and N. Abu-Ghazaleh, "Beyond the cpu: Side–channel attacks on gpus," *IEEE Design & Test*, vol. 38, no. 3, pp. 15–21, 2021.

[20] H. Naghibijouybari, A. Neupane, Z. Qian, and N. Abu-Ghazaleh, "Side channel attacks on gpus," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1950–1961, 2021.

[21] H. Naghibijouybari, A. Neupane, Z. Qian, and N. 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, (New York, NY, USA), p. 2139–2153, Association for Computing Machinery, 2018.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30, Long Beach, CA, USA*, pp. 5998–6008, 2017.

[23] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO* (M. Wiener, ed.), (Berlin, Heidelberg), pp. 388–397, Springer Berlin Heidelberg, 1999.

[24] W. J. B. Owen Lo and D. Carson, "Power analysis attacks on the aes-128 s-box using differential power analysis (dpa) and correlation power analysis (cpa)," *Journal of Cyber Security Technology*, vol. 1, no. 2, pp. 88–107, 2017.

[25] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side—channel(s)," in *Cryptographic Hardware and Embedded Systems* (B. S. Kaliski, ç. K. Koç, and C. Paar, eds.), (Berlin, Heidelberg), pp. 29–45, Springer Berlin Heidelberg, 2003.

[26] N. Mishra, T. L. Dutta, S. Shukla, A. Chakraborty, and D. Mukhopadhyay, "Too hot to handle: Novel thermal side-channel in power attack-protected intel processors," in *IEEE International Symposium on Hardware Oriented Security and Trust, HOST*, pp. 378–382, 2024.

[27] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel," in *28th USENIX Security Symposium*, (Santa Clara, CA), pp. 515–532, USENIX Association, Aug. 2019.

[28] Y. Xiang, Z. Chen, Z. Chen, Z. Fang, H. Hao, J. Chen, Y. Liu, Z. Wu, Q. Xuan, and X. Yang, "Open dnn box by power side-channel attack," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2717–2721, 2020.

[29] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA,*

USA, May 7-9, 2015, Conference Track Proceedings (Y. Bengio and Y. LeCun, eds.), 2015.

[31] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[32] W. Kwon, S. Kim, M. W. Mahoney, J. Hassoun, K. Keutzer, and A. Gholami, "A fast post-training pruning framework for transformers," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.

[33] X. Chen, Q. Cao, Y. Zhong, J. Zhang, S. Gao, and D. Tao, "Dearkd: Data-efficient early knowledge distillation for vision transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 12042–12052, IEEE, 2022.