Deep Learning for Acoustic Side-Channel Attacks on Keyboards. Sterile vs. Noisy Environments

Julia Przybytniowska*, Adam Żychowski*, and Jacek Mańdziuk*†
*Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
†Faculty of Computer Science, AGH University of Krakow, Krakow, Poland

***Abstract*—Acoustic side-channel attacks (ASCAs) on keyboards pose a significant security threat, yet existing research often overestimates their viability by focusing on sterile, noise-free environments. This paper presents a comprehensive study on the robustness and generalization of modern hybrid neural architectures (CoAtNet, MOAT, Swin Transformer) for this task. Under controlled conditions, our CoAtNet-based model establishes a new state-of-the-art accuracy of 95.23% on the most popular benchmark. Furthermore, to address the challenge of ASCA in noisy, realistic conditions, we propose a novel benchmark dataset recorded across four realistic noisy environments, and uncover a crucial phenomenon of asymmetric generalization: while models trained on noisy data generalize remarkably well to clean audio, models trained exclusively on clean data fail catastrophically in the presence of noise. Furthermore, we demonstrate that integrating Large Language Models (LLMs) as a post-processing step creates a robust information-recovery pipeline, effectively "denoising" corrupted sequences and reducing error rates to near-zero. Finally, we show the practical impact of the attack through a probabilistic guessing strategy, which recovers complex 12-character passwords in a mean of only 87.76 guesses. Our findings suggest that the integration of diverse training data and LLM-based semantic correction makes ASCAs a potent threat in real-world, unpredictable environments.**

***Index Terms*—Acoustic Side-Channel Attacks, Keystroke Recognition, Large Language Models**

I. Introduction

While modern digital security advances focus on robust encryption [1] and biometric authentication [2], side-channel attacks (SCAs) remain a persistent and often-overlooked vulnerability. Among these, acoustic side-channel attacks (ASCAs) targeting keyboards pose a significant threat. These attacks do not require breaching the system-level controls; instead, they exploit the subtle, unintentional acoustic emanations produced during typing. Each keystroke generates a unique acoustic signature [3], which, if captured and analyzed, can be used to reconstruct sensitive information, from passwords to private messages. The dynamic advancement of artificial intelligence has made the analysis of these subtle sounds, often indistinguishable to the human ear, both possible and highly effective. The proliferation of high-fidelity microphones in ubiquitous devices (e.g., smartphones, laptops, smart speakers) makes capturing these sounds trivial, even in seemingly public or noisy environments like video conferences [4].

Early approaches to ASCA relied on classical machine learning with hand-crafted features like Mel-Frequency Cepstral Coefficients (MFCCs) [4], [5]. More recently, deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated higher accuracy [6], [7]. However, existing research suffers from several key limitations that question its real-world viability. First, most studies evaluate models under sterile, noise-free conditions, failing to address the acoustic variability of real-world environments. Second, many proposed models exhibit poor generalization, often overfitting to a specific keyboard model or a single user's typing style [8]. Finally, the problem is often simplified to alphanumeric-only classification, ignoring the special characters, modifiers (e.g., Shift), and function keys that are critical for capturing passwords or system commands.

In this work, we bridge these gaps by presenting a comprehensive empirical study on the robustness and generalization of modern neural architectures for acoustic keystroke recognition. We move beyond simple CNNs and RNNs to evaluate the performance of state-of-the-art hybrid architectures, namely Swin Transformer [9], CoAtNet [10], and MOAT [11], which integrate the strengths of convolution and self-attention. To facilitate a realistic evaluation, we introduce a new, challenging dataset recorded under four distinct, real-world background noise conditions. Our analysis systematically investigates performance in both controlled (clean) and in-the-wild (noisy) scenarios, setting a new benchmark for the field.

The primary contributions of this paper are five fold:

- **Establishing a new state-of-the-art performance on the most popular ASCA benchmark.** We demonstrate that hybrid vision architectures (CoAtNet, MOAT), which integrate convolution and self-attention, outperform pure-transformer models for acoustic keystroke recognition. Our CoAtNet-based model achieves a new state-of-the-art (SOTA) accuracy of 95.23% on a primary public benchmark [12].
- **Introducing a novel ASCA benchmark.** We propose a novel benchmark *Realistic-ASCA* (R-ASCA) [13] recorded in four distinct, real-world, noisy environments (including traffic and household appliances).
- **Discovering Asymmetric Generalization.** Using R-ASCA we uncover an important phenomenon: models trained on diverse noisy data generalize well to clean audio, whereas models trained on clean data fail catastrophically in noisy conditions. This aspect was not considered in the previous ASCA papers where only clean data was considered.
- **Introducing a practical information-recovery pipeline using LLMs.** We demonstrate that integrating LLMs as

a post-processing step effectively "denoises" corrupted acoustic predictions. This semantic and syntactic correction transforms raw keystroke sequences into coherent information, reducing error rates to near-zero even in high-noise environments.
- **Assessing a probabilistic threat for password discovery.** We provide an evaluation of a probabilistic "best-first" guessing strategy, showing that model confidence can drastically reduce the search space for complex passwords. For example, a 10-character password can be recovered in a mean of only 48 guesses.

II. Related Work

Research in ASCAs has diversified into several distinct approaches. We categorize these into time-based, geometry-based, and frequency-based methods.

A. Classical ASCA Approaches

Time-based methods. Time-based analysis exploits the temporal patterns between keystrokes, such as the interval between key presses and hold durations. This has proven effective for short, fixed sequences like PINs [14], but struggles to scale to free-form text.

Geometry-based methods. Geometry-based approaches utilize the physical layout of the keyboard, often employing Time Difference of Arrival (TDoA) principles with multiple microphones to triangulate a key's position [15]–[17]. While accurate under specific conditions, the requirement for a multi-microphone array or precise device placement limits its practical feasibility in generalized attack scenarios.

Frequency-based methods. This is the most dominant and relevant line of research, as it relies on analyzing the unique spectral signature of each keystroke. Early works converted audio signals into feature representations like Fast Fourier Transform (FFT) coefficients or Mel-Frequency Cepstral Coefficients (MFCCs) and fed them into classical machine learning classifiers. Notably, [4] demonstrated the feasibility of keystroke eavesdropping over VoIP calls (like Zoom) using classifiers like Logistic Regression, achieving significant accuracy gains over random guessing. Similarly, [5] confirmed that MFCCs provide a more robust feature set than raw FFT coefficients for classifiers like SVMs and Random Forests.

B. Advanced ASCA Approaches

Deep Learning for ASCA. With the rise of deep learning (DL), the field shifted from manual feature engineering to end-to-end models. Initial works employed combinations of CNNs to extract spatial features from spectrograms [6] and RNNs like LSTMs or GRUs to model the temporal sequence of keystrokes [7]. These models demonstrated superior performance, especially in handling concurrent speech and typing. More recently, state-of-the-art vision architectures have been adapted for this task, treating spectrograms as images. This includes models like ConvMixer [18] (used to classify entire password recordings) and Vision Transformer (ViT) [19].

The SOTA benchmark and research gap. The work most pertinent to our own is [12], which introduced a high-quality dataset of keystroke recordings (which we use as a benchmark) and was the first to apply a hybrid convolutional and attention-based architecture, CoAtNet, to the problem. They achieved a high accuracy of 95% on phone-recorded data. This result set a strong benchmark for classification in clean, controlled environments.

However, a critical research gap persists, which this paper directly addresses:

1) **Robustness to real-world noise:** The vast majority of existing work, including the SOTA benchmark [12], evaluates models almost exclusively on clean, noise-free recordings. The practical utility of these models in everyday, noisy environments remains largely untested and unproven.
2) **Generalization:** While high accuracy is achieved, it is often on data from a single user or keyboard, and models show poor generalization when (or if) tested on new acoustic domains.

Our work significantly extends [12] by not only pushing the SOTA on their clean benchmark but, more importantly, by introducing a novel benchmark dataset and methodology to systematically address and evaluate robustness in realistic, noisy conditions.

III. Solution Method

Our methodology is designed to rigorously evaluate the performance of modern neural architectures for ASCA. This section details our data preparation pipeline, the datasets used for evaluation, the architectures under comparison, and our experimental setup. The overall end-to-end pipeline of our proposed approach is illustrated in Figure 1.

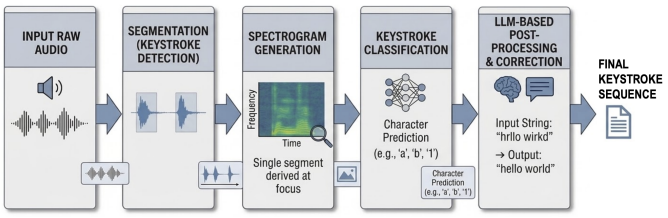

Fig. 1. High-level system flow architecture of the proposed ASCA pipeline.

A. Data Preparation Pipeline

To convert raw audio into a format suitable for DL models, we employ a two-stage process.

1. Keystroke segmentation: We first isolate individual keystroke events from continuous recordings. Raw audio files, each containing multiple consecutive keystrokes, are processed using an energy-based segmentation algorithm. The signal is transformed using a Short-Time Fourier Transform (STFT), and an energy profile is computed. A dynamic thresholding mechanism with peak clustering identifies the precise timestamp of each keystroke's acoustic peak. A fixed-size window

is extracted around each peak, and the resulting segment is trimmed of leading/trailing silence to produce a clean, isolated audio clip for each event.

2. Spectrogram generation and augmentation: Each segmented audio clip is converted into a 64×64 log-mel spectrogram. We use an n_{fft} of 512 or 1024 (depending on clip length) and 64 mel bands. This 2D representation, which visualizes frequency content over time , serves as the input to our models, transforming the task into an image classification problem. To enhance robustness and prevent overfitting, we apply online data augmentation. This includes time stretching (up to $\pm 20\%$ with 80% probability), frequency masking (40% probability), and time masking (40% probability).

B. Datasets

We evaluate performance across two distinct experimental settings: a controlled, noisy-free benchmark and a realistic, noisy simulation.

Controlled (noise-free) setting: For benchmarking and establishing the SOTA, we combine three publicly available datasets recorded on MacBook keyboards.

- `Practical` [12]: This is our primary benchmark dataset, consisting of 36 classes (a-z, 0-9) recorded via a smartphone and Zoom.
- `MKA` [20]: We utilize the raw MacBook recordings from this multi-keyboard collection.
- `Noiseless` [21]: A small, high-quality dataset of 42 classes, originally from the AI CTF challenge organized by the Cyber Security Agency of Singapore.

These datasets are segmented using our pipeline and randomly split into 70/15/15 train/validation/test sets for a robust, controlled evaluation.

Real-world noisy setting (our proposed dataset): To test model robustness and generalization, we created and segmented a new dataset. It was recorded using the built-in microphone of a MacBook Pro (14-inch, M3) across 42 independent sessions per environment, with an average duration of 30 seconds each. The dataset (3,547 samples) is structured into four distinct acoustic environments:

- `Clean`: Recorded in a quiet home environment (baseline, 872 samples).
- `Outdoor Noise`: Recorded with an open window, capturing street traffic and environmental sounds (846 samples).
- `Washing Machine`: Recorded with a nearby washing machine, introducing low-frequency, rhythmic mechanical noise (911 samples).
- `Dishwasher`: Recorded with a running dishwasher, adding intermittent water and mechanical sounds (918 samples).

C. Model Architectures

We conducted a comparative study of three state-of-the-art hybrid vision architectures, chosen for their ability to blend the local feature extraction of CNNs with the global context modeling of Transformers. To analyze the impact of model scale, we evaluated specific configurations for each architecture.

- **Swin Transformer [9]:** A hierarchical Vision Transformer that computes self-attention within non-overlapping, shifted windows (SW-MSA). This design achieves linear complexity relative to input size and is highly effective at capturing local and hierarchical visual (or, in our case, spectral) patterns. For Swin Transformer, we tested variants from *Swin-T (Small)* (5.5M params, 96 embed dim, [2,3,2] blocks) up to a large variant (*Swin-L*, 88.5M params, 128 embed dim, [2,4,18,2] blocks).
- **MOAT [11]:** A lightweight and efficient hybrid model. MOAT intertwines MBConv blocks and self-attention, but innovatively places the convolutional MBConv block *before* the attention layer within its core *MOAT block*. This allows it to learn effective downsampling kernels while retaining rich feature details. For MOAT, configurations ranged from a small model (*MOAT (Small)*, 14.3M params, [2,2,4,2] depths) to a large model (*MOAT (Large)*, 116.9M params, [3,5,9,3] depths, 256 embed dim).
- **CoAtNet [10]:** A hybrid architecture that explicitly combines convolution and attention. It typically uses MBConv [22] blocks in the early stages to efficiently extract spatial features and then transitions to Transformer blocks (using relative self-attention) in later stages to model global relationships. For CoAtNet, the evaluated models include *CoAtNet-1* (18.7M params, [2,2,2,2] blocks, [2,4,8,8] hidden dims), *CoAtNet-3* (76.4M params, [2,4,12,2] blocks, [4,8,16,16] hidden dims), and *CoAtNet-4* (133.6M params, [2,4,12,2] blocks, [4,8,16,32] hidden dims).

In contrast to the above SOTA CoAtNet configurations, our implementation, as illustrated in Figure 2, prioritizes network depth over width, deploying a significantly larger number of stacked processing blocks with narrower channel dimensions. Moreover, we avoid simplified designs based on decoupled pooling and standard dot-product attention; instead, we integrate learnable downsampling directly within the blocks and employ Relative Self-Attention to rigorously preserve positional information.

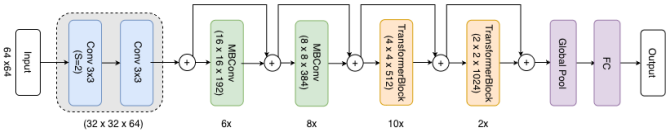

Fig. 2. A schematic architecture of the best-performing CoAtNet-3.

D. Experimental Setup

All models are trained using the PyTorch framework [23] for up to 1000 epochs, with early stopping (patience of 50 epochs) based on validation loss. We use the *CrossEntropyLoss* function. Optimization is performed using the AdamW

optimizer [24] paired with a *CosineAnnealingWarmRestarts* learning rate scheduler. The source code is available at https://github.com/przybytniowskaj/KeystrokeDetection [13].

We conduct experiments across two task definitions:

1) **Alphanumeric-only:** A 36-class problem (a-z, 0-9), consistent with the SOTA benchmark [12].
2) **Full-keyboard:** A more complex, real-world task including all keys (e.g., symbols, modifiers, spacebar) from both our custom and noiseless datasets.

Evaluation is primarily based on Top-1 and Top-k accuracy. For the real-world simulation (i.e., in noisy surroundings), we explicitly test asymmetric generalization by training on the `Clean` subset and testing on the three noisy subsets (Outdoor Noise, Washing Machine, and Dishwasher), and vice-versa. To assess practical utility, we measure the Levenshtein distance on reconstructed text and use LLM (Gemini 3.0 Pro) as a post-processing step to correct semantic and syntactic errors.

IV. Experimental Results

We present our results in four parts. First, we establish a new SOTA results on popular, noise-free benchmarks (Section IV-A). Second, we analyze model robustness and generalization using our novel real-world noisy dataset (Section IV-B). Third, we assess the strength of the probabilistic password guessing algorithm (Section IV-C). Finally, we assess the practical end-to-end performance using text reconstruction and LLM-based error correction (Section IV-D).

A. Controlled, Noise-free Settings

We evaluated all architectures on the three combined noise-free datasets (Practical, MKA, and Noiseless). Table I summarizes the Top-1 and Top-5 accuracy for both the alphanumeric-only (36 classes) and full-keyboard (all classes) tasks.

TABLE I
Model performance on the combined noise-free benchmarks(controlled, clean conditions). The best results in each metrics are **bolded**.

Model	Params (M)	Alphanumeric		Full-Keyboard	
		Top-1 Acc	Top-5 Acc	Top-1 Acc	Top-5 Acc
Swin-T (Small)	5.5	38.47	76.79	39.07	76.84
Swin-L (Large)	88.5	55.30	88.47	58.47	87.32
MOAT (Small)	14.3	80.06	97.66	87.71	98.84
MOAT (Large)	116.9	85.98	**98.91**	89.65	98.58
CoAtNet-1	18.7	84.27	98.44	83.31	98.19
CoAtNet-3	76.4	**92.21**	98.75	**90.94**	98.84
CoAtNet-4	133.6	90.65	98.91	90.82	**98.97**

The results show a clear hierarchy. CoAtNet and MOAT, which integrate both convolution and attention, significantly outperform the pure-transformer Swin architecture. This suggests that the local inductive bias of convolution is highly beneficial for processing spectrograms. The CoAtNet-3 architecture achieves the highest accuracy, reaching 92.21% on the 36-class task and 90.94% on the full-keyboard task.

Crucially, when evaluating our CoAtNet-3 model solely on the test split of the `Practical` dataset [12], it achieves 95.23% accuracy. This surpasses the 95.0% reported in the original paper, establishing a new state-of-the-art for this benchmark.

The Top-5 accuracies are very high (98-99%) for both CoAtNet and MOAT, indicating that even when the top prediction is wrong, the correct key is almost always within the top few guesses. Analysis of confusion matrix (see Fig. 3) confirms that the rare misclassifications are not random, but occur between acoustically similar or physically adjacent keys (e.g., `w` and `s`, `j` and `k`).

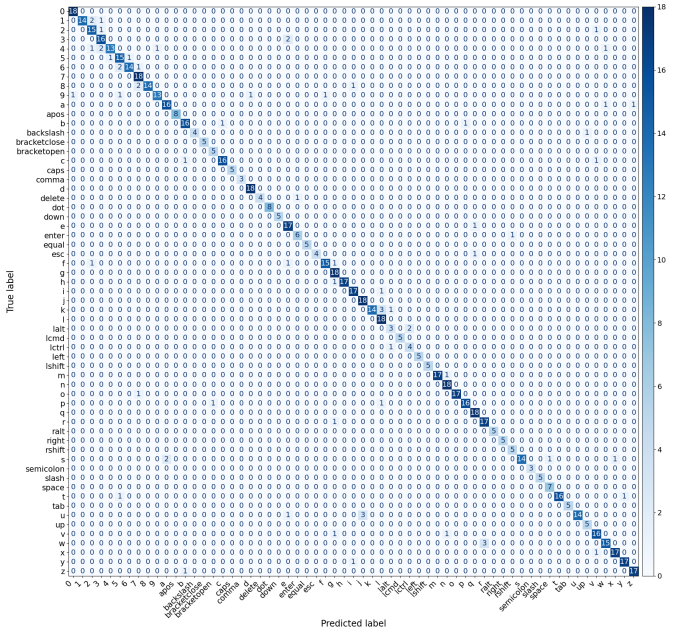

Fig. 3. CoAtNet-3 confusion matrix on the public dataset, showing strong diagonal concentration and minor adjacent-key confusion.

B. Real-World Simulation Generalization

While previous subsection established a new SOTA accuracy, those experiments were conducted under sterile, noise-free conditions. Such "perfect" scenarios rarely reflect real-world environments, which are typically polluted with background noise. This section transitions to a more realistic evaluation, designed specifically to test model *robustness*. We investigate not only performance in noisy settings but also the critical challenge of *knowledge transfer*, analyzing the asymmetric generalization that occurs when models are trained on clean audio and tested on noise, and vice-versa.

First, we evaluated the models on our more challenging custom dataset, which includes four distinct acoustic environments. Models were trained on the full dataset (`Clean` + all 3 noisy subsets).

Table II shows that while performance on the unseen `Clean` test subset remains high (95.86%), accuracy drops significantly in noisy conditions, particularly for the `Outdoor Noise` subset (76-78%). This confirms the difficulty of real-world ASCA.

TABLE II
PERFORMANCE OF BEST MODELS WHEN TRAINED ON THE **FULL** CUSTOM DATASET (CLEAN + NOISY) AND EVALUATED ON EACH ENVIRONMENT SEPARATELY (FULL-KEYBOARD TASK).

Model	Overall	Accuracy by Test Environment			
	Acc	Clean	Outdoor	Dishwasher	Washing Machine
MOAT (Large)	85.68	95.86	76.47	87.77	83.60
CoAtNet-4	84.72	95.27	74.87	87.77	82.01

To better understand generalization, we conducted two cross-domain experiments - see results in Table III.

TABLE III
THE DISCOVERY OF ASYMMETRIC GENERALIZATION (**BOLDED**). TRAINING ON CLEAN DATA AND TESTING ON NOISE FAILS, BUT TRAINING ON NOISE GENERALIZES WELL TO CLEAN DATA.

Model	Training Set	Test Set	Alphanumeric		Full-Keyboard	
			Top-1 Acc	Top-5 Acc	Top-1 Acc	Top-5 Acc
MOAT	Clean	Clean (Baseline)	70.00	94.00	87.57	98.22
		Noisy (Transfer)	**8.36**	**31.89**	**13.48**	**31.56**
CoAtNet	Noisy	Noisy (Baseline)	72.14	96.90	78.72	95.04
		Clean (Transfer)	**63.00**	**92.00**	**71.01**	**98.67**

1. Clean → Noisy Failure: Models trained *only* on the `Clean` dataset perform well on the clean test set (e.g., 87.57% for Full-Keyboard MOAT). However, when these models are tested on the unseen `Noisy` subsets, their performance collapses catastrophically, dropping to 13.48%.

2. Noisy → Clean Generalization: Conversely, models trained *only* on the Noisy subsets (`Outdoor`, `Washing Machine`, `Dishwasher`) not only perform well on unseen noisy data (e.g., 78.72% for Full-Keyboard CoAtNet) but also generalize well to the unseen `Clean` test set, achieving 71.01% accuracy.

This demonstrates a crucial asymmetric generalization: training on diverse, noisy data is essential for building robust models that can function in the real world. A model trained only on "perfect" data is useless outside a lab.

C. Probabilistic Password Guessing

To quantify the practical security threat posed by ASCAs, we simulated a targeted password recovery scenario using our best-performing CoAtNet-3 model. This experiment evaluates how effectively the model's probabilistic output can guide an intelligent guessing strategy.

For the evaluation, we generated a set of random passwords across four distinct lengths: 6, 8, 10, and 12 characters. For each length, 5 unique passwords were created and each of them was typed 5 times, leading to 25 samples per password length). These targets were composed of a highly diverse character set, including lowercase and uppercase letters, digits, and special characters. A representative examples of a password used in experiments are `!Wy%'}Ghe'+9` or `QIhH8B5JmRng`.

The complexity of the attack is determined by the number of individual acoustic events rather than the nominal character count of the password. Since uppercase letters and many special characters require the simultaneous use of a modifier key (e.g., the Shift key), the actual number of captured keystrokes was often higher than the character length of the password. Our model was tasked with identifying each discrete keystroke within these extended acoustic sequences.

The simulated attack utilizes a structured "best-first" guessing strategy, where the model produces a ranked probability distribution across the entire keyboard set for every captured keystroke in a sequence. The process initiates with a Top-1 guess—a sequence composed of the most probable character for each event. If this initial sequence is incorrect, the algorithm begins an iterative exploration of the search space based on the model's confidence levels. In each subsequent step, the system identifies and selects the sequence from the set of all candidates not yet tested that yields the highest sum of individual character probabilities.

The results presented in Table IV demonstrate the practical risk by showing how the model's high Top-k accuracy drastically reduces the effective search space for complex, real-world passwords. Even as the password length increases, the "best-first" strategy remains highly efficient. For a 12-character password, the mean number of guesses is only 87.76, while the median remains as low as 2. This indicates that in most cases, the correct sequence is identified almost immediately, making this attack significantly more potent than traditional brute-force methods.

TABLE IV
NUMBER OF GUESSES REQUIRED FOR SUCCESSFUL PASSWORD RECOVERY USING THE PROBABILISTIC ATTACK STRATEGY, SHOWN AS A FUNCTION OF PASSWORD LENGTH. FOR EACH LENGTH, 25 TRIALS WERE PERFORMED.

Password length	Min	Max	Mean	Median
6	1	27	4.00	1
8	1	164	9.84	1
10	1	668	48.04	1
12	1	807	87.76	2

D. Practical Reconstruction with LLM Correction

While the DL models described in previous sections achieve high character-level accuracy, the ultimate goal of the ASCA is the successful reconstruction of coherent information, such as commands or full sentences. In real-world scenarios, even a small percentage of misclassified keystrokes can render a captured sequence unintelligible. To address this, we explore the use of LLMs as a post-processing step.

Motivation for LLM Integration. Traditional error-correction methods, such as dictionary-based spell checkers, often fail in the context of keystroke recognition. They typically lack the semantic awareness required to distinguish between acoustically similar keys that result in valid but contextually incorrect words (e.g. "Can you *wend* the report?"). Furthermore, they are generally ineffective at correcting technical strings, such as code snippets or shell commands. LLMs, however, leverage vast contextual knowledge and can infer the intended meaning of a "noisy" or corrupted string by analyzing

the surrounding syntax and semantics [25]–[27]. This makes them uniquely suited for reconstructing both natural language and highly structured technical data.

Experimental Setup. To evaluate the effectiveness of LLM-based correction, we constructed a dataset consisting of 30 distinct sequences, divided into two categories:

- **Natural Language Sentences:** 15 short English sentences representing common communication (e.g. *"It was a bad time to leave"*, *"Did you see the news?"*).
- **Technical Commands:** 15 strings representing programming code, terminal commands, and URLs (e.g. *"ssh root@192.168.1.55"*, *"chmod 755 ./script.sh"*, *"pip install numpy pandas"*, *"for i in range(10): print(i * 2)"*).

Methodology. The experiment followed a multi-stage pipeline. Each of the 30 sequences was subjected to 5 independent trials to ensure statistical robustness. In each trial, different acoustic signal of the typing sequence was processed by our best-performing model (CoAtNet-3) to produce a raw predicted string. These raw outputs - often containing errors - were then passed to the Gemini 3.0 Pro model with a prompt instructing it to perform semantic and syntactic correction.

The evaluation was conducted across two acoustic environments: *Clean* (the baseline quiet environment) and *Noise* (representing realistic background interference). The results, discussed in Table V, compare the error rates before and after the LLM correction across both text categories and environmental conditions.

TABLE V
LEVENSHTEIN DISTANCE BEFORE AND AFTER GEMINI CORRECTION FOR NATURAL LANGUAGE AND TECHNICAL COMMANDS.

Environment	Type	Raw			Gemini		
		Min	Mean	Max	Min	Mean	Max
Clean	Natural Language	0.00	1.13	4.00	0.00	0.05	2.00
	Technical Commands	0.00	1.40	5.00	0.00	0.12	2.00
Noisy	Natural Language	3.00	8.41	13.00	0.00	0.37	9.00
	Technical Commands	3.00	8.49	22.00	0.00	0.85	13.00

The results demonstrate that LLM-based correction significantly enhances the practical utility of the attack, particularly in the Noisy environment. In clean conditions, the LLM reduces an already low mean error rate to near-zero (0.05 for Natural Language and 0.12 for Technical Commands). However, the most substantial advantage is observed in the Noisy environment, where raw acoustic predictions are heavily degraded by background interference, reaching mean Levenshtein distances of approximately 8.4. In such cases, the LLM acts as a powerful "denoising" filter, leveraging language patterns to reduce the mean error distance to 0.37 for Natural Language sentences and 0.85 for Technical Commands.

Statistical analysis using a Wilcoxon signed-rank test validates that this improvement is statistically significant with $p < 0.001$. This proves that even when the acoustic signal is heavily corrupted, the intended information can still be recovered with high reliability.

1) Reconstruction Examples: The following examples illustrate the reconstruction capability in noisy environments, showing the compressed token sequences.

Example 1: Natural Language. The LLM reconstructed the corrupted word perfectly.
Ground Truth: `Don't quit on this project.`
Raw Predicted: `[shift]con'[shift][space]qji [shift][space]on[space]thi1[space]prik4c[down]`
Reconstructed: `[shift]don't[space]quit[space]on [space]this[space]project`

Example 2: Technical Command (SQL). In this instance, the model confused several characters due to physical proximity and acoustic similarity (e.g., confusing 's' and 'e' with '1' and '4'). The LLM successfully identified the SQL syntax.
Ground Truth: `select * from users where registrationYear > 2024;`
Raw Predicted: `14lsct[space][shift]8[space]5rom [space]j14r1[space]eh4r4[space]rsyi1tration [shift]ysar[space][shift][down][space]2m2w;`
Reconstructed: `select[space][shift]8[space]from [space]users[space]where[space]registration [shift]year[space][shift].[space]2023;`

Presented results show that integration of an LLM as a post-processing step transforms ASCA from a character-recognition task into a robust information-recovery pipeline. By prioritizing semantic coherence, LLMs mitigate the physical limitations of acoustic classification, making the attack potent even in the presence of significant environmental noise.

V. CONCLUSIONS

This paper presents a comprehensive empirical study on the viability and robustness of modern neural architectures for acoustic keystroke recognition. Our findings yield four key insights that redefine the understanding of acoustic side-channel attacks in real-world scenarios.

First, we demonstrated that hybrid architectures integrating convolution and attention (CoAtNet, MOAT) significantly outperform pure-transformer models. By leveraging the local inductive bias of convolution and the global context of self-attention, our CoAtNet-based model established a new state-of-the-art accuracy of 95.23% on the most popular benchmark.

Second, we provided the first systematic evaluation of robustness against realistic background noise. This led to the discovery of asymmetric generalization: while models trained on clean audio fail catastrophically in noisy environments (dropping from 87% to 13% accuracy), models trained on diverse, noisy data generalize remarkably well to clean audio. This implies that many previous studies using only sterile data may have overestimated the threat in some areas while underestimating the efficacy of robust training.

Third, we showed that the integration of LLMs as a post-processing step transforms ASCA from a simple character-recognition task into a robust information-recovery pipeline. The LLM acts as a semantic filter that "denoises" corrupted

sequences, making the attack potent even when the acoustic signal is heavily degraded.

Finally, our probabilistic threat assessment revealed that even complex 12-character passwords can be recovered in a mean of only 87.76 guesses. This demonstrates that ASCAs are no longer a theoretical lab-based curiosity but a practical, high-impact threat.

Our discovery of asymmetric generalization provides a nuanced perspective on noise-based countermeasures. At first glance, environmental noise appears to be a highly effective defense for models trained in sterile conditions, where full-keyboard accuracy collapses from 87.57% to 13.48%. However, our results demonstrate that noise is a fragile barrier. A sophisticated attacker who incorporates diverse, noisy environments into their training set can bypass this defense entirely, as models trained on noise generalize remarkably well back to clean audio, maintaining 71.01% accuracy.

Consequently, simple software-based acoustic masking is insufficient against robustly trained architectures. More effective defenses must prioritize physical interventions, such as the use of silent mechanical switches or sound-dampening keyboard covers, to minimize the acoustic emanations at their source. On the digital side, introducing random timing jitters or low-pass filtering in VoIP audio transmission could disrupt the temporal patterns and spectral clarity required for accurate keystroke segmentation.

Limitations of this study include the focus on a single family of keyboards (MacBook) and data from a limited number of typists. Our future work will focus on cross-device and cross-user generalization, as well as exploration of adaptation techniques to efficiently fine-tune the models to new hardware. Additionally, we plan to explore adversarial defenses, mitigation strategies, and architectural innovations for inherent noise tolerance, such as training with clean data augmented by synthetic noise profiles.

REFERENCES

[1] P. Dixit, A. K. Gupta, M. C. Trivedi, and V. K. Yadav, "Traditional and hybrid encryption techniques: a survey," in *Networking Communication and Data Knowledge Engineering*. Springer, 2018, pp. 239–248.

[2] Z. Rui and Z. Yan, "A survey on biometric authentication: Toward secure and privacy-preserving identification," *IEEE Access*, vol. 7, pp. 5994–6009, 2019.

[3] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, p. 408280, 2013.

[4] A. Compagno, M. Conti, D. Lain, and G. Tsudik, "Don't Skype & Type! Acoustic Eavesdropping in Voice-Over-IP," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 703–715.

[5] S. A. Anand and N. Saxena, "Keyboard emanations in remote voice calls: Password leakage and noise (less) masking defenses," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 2018, pp. 103–110.

[6] T. Giallanza, T. Siems, E. Smith, E. Gabrielsen, I. Johnson, M. A. Thornton, and E. C. Larson, "Keyboard snooping from mobile phone arrays with mixed convolutional and recurrent neural networks," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–22, 2019.

[7] D. Slater, S. Novotney, J. Moore, S. Morgan, and S. Tenaglia, "Robust keystroke transcription from the acoustic side-channel," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 776–787.

[8] A. Taheritajar, Z. M. Harris, and R. Rahaeimehr, "A survey on acoustic side channel attacks on keyboards," in *International Conference on Information and Communications Security*, 2024, pp. 99–121.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[10] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Advances in neural information processing systems*, vol. 34, pp. 3965–3977, 2021.

[11] C. Yang, S. Qiao, Q. Yu, X. Yuan, Y. Zhu, A. Yuille, H. Adam, and L.-C. Chen, "MOAT: Alternating Mobile Convolution and Attention Brings Strong Vision Models," in *International Conference on Learning Representations*, 2023.

[12] J. Harrison, E. Toreini, and M. Mehrnezhad, "A practical deep learning-based acoustic side channel attack on keyboards," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2023, pp. 270–280.

[13] Przybytniowska, Julia. (2025) Realistic-ASCA: A realistic acoustic side-channel attack dataset. [Online]. Available: https://github.com/przybytniowskaj/KeystrokeDetection

[14] X. Liu, "When keystroke meets password: Attacks and defenses," PhD dissertation, Singapore Management University, Singapore, 2019.

[15] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 142–154.

[16] Y. Rosmansyah *et al.*, "The microphone array sensor attack on keyboard acoustic emanations: Side-channel attack," in *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*. IEEE, 2017, pp. 261–266.

[17] G. de Souza Faria and H. Y. Kim, "Differential audio analysis: a new side-channel attack on pin pads," *International Journal of Information Security*, vol. 18, pp. 73–84, 2019.

[18] A. Akinbi, E. Deniz, A. M. Ismael, Z. N. Rashid, and A. Sengur, "Password-sniffing acoustic keylogger using machine learning," *Available at SSRN 4431909*, 2023.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[20] K. M. H. Rawf, A. O. Abdulrahman, H. O. Kamel, L. M. Hassan, and A. O. Ali, "Multi-datasets for different keyboard key sound recognition," *Data in Brief*, vol. 57, p. 110949, 2024.

[21] N. C. Duy, "Keystroke Noiseless Final Data," https://www.kaggle.com/datasets/nguyncaoduy/keystroke-noiseless-final, accessed: 2026-01-19.

[22] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 model for image classification," in *2nd International Conference on Information Technology and Computer Application*. IEEE, 2020, pp. 476–480.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] S. Ruder, "An overview of gradient descent optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1609.04747

[25] Z. Min and J. Wang, "Exploring the integration of large language models into automatic speech recognition systems: An empirical study," in *International Conference on Neural Information Processing*. Springer, 2023, pp. 69–84.

[26] A. Thomas, R. Gaizauskas, and H. Lu, "Leveraging LLMs for post-OCR correction of historical newspapers," in *Proceedings of the 3rd Workshop on Language Technologies for Historical and Ancient Languages*, 2024, pp. 116–121.

[27] J. Kanerva, C. Ledins, S. Käpyaho, and F. Ginter, "OCR Error Post-Correction with LLMs in Historical Documents: No Free Lunches," in *Proceedings of the Third Workshop on Resources and Representations for Natural Language Processing (RESOURCEFUL 2025)*, 2025.