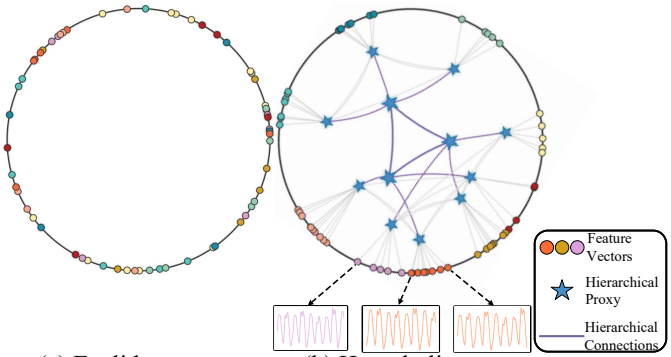
HypSCA: A Hyperbolic Embedding Method for Enhanced Side-channel Attack

Kaibin Li, Yihuai Liang, Zhengchun Zhou, *Member, IEEE*, Shui Yu, *Fellow, IEEE*

***Abstract*—Deep learning-based side-channel attack (DLSCA) has become the dominant paradigm for extracting sensitive information from hardware implementations due to its ability to learn discriminative features directly from raw side-channel traces. A common design choice in DLSCA involves embedding traces in Euclidean space, where the underlying geometry supports conventional objectives such as classification or contrastive learning. However, Euclidean space is fundamentally limited in capturing the multi-level hierarchical structure of side-channel traces, which often exhibit both coarse-grained clustering patterns (e.g., Hamming weight similarities) and fine-grained distinctions (e.g., instruction-level variations). These limitations adversely affect the discriminability and generalization of learned representations, particularly across diverse datasets and leakage models. In this work, we propose HypSCA, a dual-space representation learning method that embeds traces in hyperbolic space to exploit its natural ability to model hierarchical relationships through exponential volume growth. In contrast to existing approaches, HypSCA jointly combines hyperbolic structure modeling with local discriminative learning in Euclidean space, enabling the preservation of global hierarchies while enhancing fine-grained feature separation. Extensive experiments on multiple public datasets demonstrate that HypSCA achieves up to 51.6% improvement in attack performance over state-of-the-art DLSCA methods, consistently enhancing generalization across diverse datasets and leakage models.**

***Index Terms*—Side-channel attack, deep learning, hierarchical regularization, triplet loss, profiled attacks.**



(a) Euclidean space (b) Hyperbolic space

Fig. 1: Comparison of trace embeddings in Euclidean (a) and hyperbolic (b) spaces. Each point represents a trace embedding from electromagnetic measurements, with each color denoting a different class based on the Hamming weight of the intermediate cryptographic values (ranging from 0 to 8 bits set). Sample waveforms for two representative classes (orange and purple) are shown in the insets. In Euclidean space, traces of the same class cluster together but the layout remains scattered. In hyperbolic space, traces are organized around class-specific proxy nodes (blue stars ★) connected through hierarchical relationships (purple lines), forming a tree-like structure.

I. Introduction

DEEP learning-based side-channel attack (DLSCA) has emerged as a powerful cryptanalytic technique that leverages neural networks to automatically extract discriminative features from raw side-channel traces, eliminating the need for manual preprocessing and significantly improving attack effectiveness [1], [2]. Building on this foundation, researchers have proposed numerous enhancements, including architectural innovations [3]–[6], loss function refinements [7]–[9], and hyperparameter optimization techniques [10], [11]. Complementary to these efforts, Wu et al. [12] introduced a label correlation approach to enhance model generalization while reducing reliance on large training sets. Despite these advances, DLSCA models still struggle to generalize across datasets and scenarios, particularly under varying noise levels [13], input distributions [14], [15], or environmental conditions [16]. To improve both generalization and attack performance, it is essential to move beyond conventional classification-based DLSCA approaches toward more expressive, structure-aware embedding learning strategies.

To address these limitations, researchers have begun exploring advanced representation learning approaches that aim to learn discriminative embedding spaces by optimizing feature separability. Methods have been proposed to explicitly improve trace distinguishability, such as Wu et al. [17] who proposed a triplet loss-based approach for enhancing local feature discriminability, and Li et al. [18] who employed normalized lifted structured loss to improve both attack performance and training efficiency. However, existing approaches overlook a critical aspect: the inherent multi-level hierarchical structure [19] of side-channel traces, which encompasses both coarse-grained semantic patterns (such as algorithm-level operations) and fine-grained details (such as instruction-level leakages). If captured explicitly, such hierarchical relationships could better reflect the latent organization of trace data and enhance representation learning. Indeed, embedding hierarchical relations has proven beneficial in other domains such as document parsing [20] and structured clustering [21], where whole-part

This work was supported by the National Natural Science Foundation of China (62131016); (*Corresponding author: Yihuai Liang.*)

K. Li, Y. Liang and Z. Zhou are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611730, China. E-mail: lkb@my.swjtu.edu.cn, {liangyh, zzc}@swjtu.edu.cn.

S. Yu is with School of Computer Science, University of Technology Sydney, Australia. E-mail: shui.yu@uts.edu.au.

relations help structure the feature space. Intuitively, the structured nature of cryptographic execution suggests that embeddings capturing hierarchical relationships among traces may yield more generalizable and robust representations. Fig. 1 illustrates this fundamental geometric limitation by comparing Euclidean and hyperbolic embeddings of the same set of side-channel traces, where each point represents a trace from electromagnetic waveforms and each color denotes a different class. In Fig. 1(a), Euclidean space fails to capture any higher-order structure. Although traces of the same class (same color) tend to group together, the layout remains scattered, and inter-class relationships are not explicitly modeled. In contrast, Fig. 1(b) depicts the same traces embedded in hyperbolic space, where the geometry induces a tree-like arrangement. Traces from the same class are grouped around a shared parent node, represented by a blue star (★). These pentagrams serve as hierarchical proxies, summarizing the collective features of their associated child traces. For instance, all purple points are linked to their class-specific proxy node, and the same pattern holds for other classes. This structured layout uncovers both intra-class coherence and inter-class separation, reflecting the nested nature of cryptographic operations. The contrast highlights a fundamental shortcoming of current DLSCA methods: the use of Euclidean geometry hinders the modeling of the latent trace hierarchy intrinsic to side-channel data.

In this work, we explore a previously underexplored perspective in DLSCA: leveraging hyperbolic geometry to capture the hierarchical structure intrinsic to side-channel traces. Specifically, hyperbolic space [22] possesses the distinctive property of exponential volume growth, making it naturally suited for embedding tree-like cryptographic operation structures. This characteristic is particularly relevant for side-channel traces, where the Hamming weight leakage model induces hierarchical organization of trace clusters based on bit-level operations, creating implicit tree-like relationships that are better captured in hyperbolic space. Learning representations in hyperbolic space has demonstrated remarkable effectiveness across diverse machine learning domains, including category discovery [23], [24], graph embedding [25], and computer vision applications [26], [27]. Building on these successes, we aim to learn hierarchy-aware representations that yield more discriminative features for SCA, thereby advancing both attack effectiveness and cross-device generalization capabilities.

To achieve this goal, we propose a novel method named *Hyperbolic Embedding method for SCA (HypSCA)*, which leverages hyperbolic space embedding to significantly improve attack performance. HypSCA integrates two synergistic learning mechanisms: (1) a hierarchical regularization (HIER) module that explicitly models global trace hierarchies in hyperbolic space by capturing the tree-like structure of cryptographic operations; and (2) a triplet loss component in Euclidean space that sharpens local discriminability among traces corresponding to different key hypotheses. This dual-space optimization strategy addresses a core limitation of existing DLSCA methods, which predominantly focus on local relationships while failing to account for the global hierarchical organization inherent in side-channel traces. By jointly modeling global structure and local distinctions, HypSCA produces more expressive and generalizable trace representations. Experimental results show that HypSCA substantially enhances the separability of trace classes. This complementary geometric perspective facilitates more robust key recovery, even in scenarios with limited training data under noisy conditions.

The main contributions of this paper are summarized as follows:

- We introduce hyperbolic embeddings as a novel representation space for SCA, enabling semantically structured representations that effectively capture the inherent hierarchical structures in side-channel traces. This geometric approach facilitates clearer separation between trace classes and substantially improves model generalization across diverse attack scenarios.
- We develop a dual-space optimization strategy that synergistically combines a hyperbolic HIER module for preserving global trace hierarchies with a discriminative triplet loss in Euclidean space for enhancing local feature distinction. This integrated approach simultaneously addresses the hierarchical nature of trace data and specific SCA challenges such as desynchronization effects.
- We demonstrate through extensive experiments on three public datasets under two different leakage models. Our approach consistently outperforms state-of-the-art DLSCA approaches, achieving up to a 51.6% improvement in attack performance and an average improvement of 26.2% over existing methods.

The remainder of this paper is organized as follows. Section II provides an overview of related work. Section III introduces the notation used throughout the paper and presents background on profiled SCA, TA, and DLSCA. Section IV details the proposed HypSCA. Section V reports the experimental results. Finally, Section VI concludes the paper and outlines potential directions for future research.

II. Related Work

SCA exploits physical leakages such as power consumption, electromagnetic emissions, or timing information to extract sensitive data from cryptographic devices [28]. Profiled SCA, which trains models on profiling devices to attack target devices, has proven especially effective [19]. Early profiled attacks required manual preprocessing but suffered from hardware variability and noise sensitivity, motivating a shift toward DLSCA that extracts features automatically from raw traces.

Deep learning fundamentally transformed profiled SCA by eliminating manual feature engineering and improving robustness against countermeasures. Maghrebi et al. [1], [2] demonstrated CNNs' ability to overcome temporal jitter, establishing DLSCA as a promising approach. Recent works have advanced DLSCA in various directions. For example, Li et al. [29] proposed gradient-based techniques for pre-silicon leakage detection, significantly speeding verification. Adaptive chosen-plaintext attacks guided by neural leakage analysis [4] reduce trace requirements on ARM devices. Other efforts include neural architecture deconstruction via f-ANOVA [5] and sophisticated feature engineering for IoT countermeasure

detection [6]. Despite these advances [4], [5], generalization remains a major challenge. Model performance degrades across datasets, noise levels, and operational conditions [1], [11], [13], [15]. To mitigate this, hyperparameter optimization and ensembles have been explored [10], alongside CNN design guidelines tailored for SCA [13].

Addressing these generalization challenges has led to significant innovation in loss function design for DLSCA applications. While categorical cross-entropy loss ($\mathcal{L}_{CCE}$) [14], [19] remains the standard choice due to its theoretical connection to perceived information and its role as a mutual information lower bound [30], its limitations in handling specific SCA challenges have sparked the development of specialized alternatives. Zhang et al. [7] introduced cross-entropy ratio loss ($\mathcal{L}_{CER}$) to improve prediction consistency through comparative analysis of profiling traces with original and shuffled labels. Building upon this foundation, Zaid et al. [8] proposed ranking loss ($\mathcal{L}_{RKL}$), which employs sigmoid functions for pairwise comparison between correct and incorrect key hypotheses, thereby enhancing separation quality. Subsequently, focal loss ratio ($\mathcal{L}_{FLR}$) [9] was developed to refine ($\mathcal{L}_{CER}$) by specifically addressing class imbalance issues prevalent in SCA datasets.

Recent advances in representation learning have focused on enhancing feature discriminability through various optimization strategies. Wu et al. [17] pioneered triplet-based losses optimizing relative distances between anchor, positive, and negative samples. Yap et al. [16] proposed soft nearest neighbor and center losses to improve embedding separation, reducing required traces. Complementary label correlation metrics [12] enhanced these efforts by developing label correlation-based approaches that improve generalization through learning from label distributions rather than traditional one-hot encodings. Most recently, Li et al. [18] proposed a comprehensive framework employing normalized lifted structured loss ($\mathcal{L}_{NLS}$) combined with pairwise similarity learning strategies, demonstrating both improved attack effectiveness and enhanced training efficiency.

However, despite these advances in optimization techniques, the fundamental geometric constraints of Euclidean embedding spaces may inherently limit their capacity to capture the complex hierarchical relationships present in side-channel data. The flat geometry and limited volume growth of Euclidean space hinder the natural representation of tree-like structures that reflect the nested organization of cryptographic operations. In contrast, hyperbolic geometry, with its exponential volume growth and negative curvature, has proven highly effective for representing hierarchical relationships in other domains, showing success in NLP [22] and graph embeddings [25]. Works by Yang et al. [31] and Kim et al. [26] confirm hyperbolic embeddings' effectiveness for hierarchical data, yet their application to DLSCA remains unexplored.

This paper presents the first systematic study of hyperbolic embeddings for cryptographic SCA, aiming to leverage their geometric properties to better capture the inherent hierarchical structure of side-channel traces and improve both discriminative power and generalization in DLSCA models.

TABLE I: Key notations

Notation	Description
Data and Feature Notations	
$\mathcal{X}$	Domain of side-channel traces (input features).
$\mathcal{Y}$	Domain of output classes (e.g., intermediate values).
T_i^p, T_j^a	ith and jth traces in profiling and attack datasets.
y_i	Label corresponding to the ith trace.
D_p, D_a	Profiling and attack datasets.
Model, Loss Functions, and Geometry	
$f_\theta(\cdot)$	Profiling model trained on D_p.
θ	Parameters of the profiling model.
$\{x_i, x_j, x_k\}$	Feature vectors in embedding space.
ρ_{ij}	Hierarchical proxy for x_i and x_j in hyperbolic space.
$\mathbb{D}_c^n$	n-dimensional Poincaré ball (hyperbolic space).
$d_E(\cdot,\cdot)$	Euclidean distance.
$d_H(\cdot,\cdot)$	Hyperbolic distance.
$d_{\text{hybrid}}(\cdot,\cdot)$	Hybrid (Euclidean-based) distance.
$d'(l_x, l_y)$	Normalized distance between label embeddings.
δ_E, δ_H	Margins for triplet and HIER losses.
α	Scale factor for d_{hybrid}.
Cryptographic Notations	
$\mathcal{K}$	Keyspace of all possible secret keys.
k^*	Ground-truth secret key used in encryption.

III. Preliminaries

This section reviews the background knowledge of SCA, focusing on profiled SCA and DLSCA. These foundational techniques are critical for understanding the core objectives and mechanisms of SCA, as well as the proposed HypSCA. TABLE I summarizes the key notations used throughout this study, particularly in the context of profiled SCA.

A. Profiled SCA

Profiled SCA is one of the most powerful techniques in the field of SCA, leveraging a clone device to infer secret information from a target device. This technique consists of two distinct phases: the *profiling phase* and the *attack phase*, with the ultimate goal of recovering the secret cryptographic key k^*.

In the *profiling phase*, the evaluator collects a labeled dataset D_{p} from the clone device, defined as:

$$D_p = \{(T_i^p, y_i) \mid i \in [1, N_p]\}, \tag{1}$$

where $T_i^p \in \mathcal{X}$ represents the ith side-channel trace, $y_i \in \mathcal{Y}$ is the corresponding label (e.g., an intermediate cryptographic value), and $N_p = |D_p|$. A profiling model $f : \mathcal{X} \to \mathcal{Y}$ is trained on this dataset.

In the *attack phase*, an unlabeled dataset D_a is collected from the target device, defined as:

$$D_a = \{T_j^a \mid j \in [1, N_a]\}, \tag{2}$$

where $T_j^a \in \mathcal{X}$ represents the jth side-channel trace, and $N_a = |D_a|$. The profiling model f is applied to each trace T_j^a, producing posterior probabilities over possible labels:

$$f(T_j^a) = P(\hat{y}_j | T_j^a), \hat{y}_j \in \mathcal{Y}. \tag{3}$$

These probabilities are further processed to infer the secret key k^* using statistical techniques such as maximum likelihood estimation or key enumeration.

B. Deep Learning-based SCA (DLSCA)

DLSCA has attracted increasing attention for its strong capability in capturing complex, high-dimensional, and non-linear structures in leakage data [19]. The core objective of deep learning in SCA is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps input features (e.g., side-channel traces) to discrete output classes.

During the *training phase*, a DL model, such as CNN, is trained on the labeled dataset D_p by minimizing a loss function $\mathcal{L}(f(T_i^p; \theta), y_i)$, where θ represents the network parameters. Based on this, we can predict the class probability $P(y \mid T_j^a)$ for each attack trace T_j^a:

$$P(y \mid T_j^a) = f_{\text{CNN}}(T_j^a; \theta_{\text{CNN}}), \tag{4}$$

These probabilities derived from Eq. (3), and (4) can all be used to recover the secret key k^* through maximum likelihood estimation or key enumeration. DLSCA methods demonstrate strong resilience against noise, high dimensionality, and desynchronized traces. Furthermore, DLSCA does not rely on strong assumptions about the leakage distribution, making it well-suited for practical scenarios where noise and misalignment are significant factors. However, these methods typically require a large number of traces for training as well as high computational resources.

IV. Our Hyperbolic Embedding Method for SCA

In this section, we introduce the proposed HypSCA method for profiled SCA, the core of which is the HIER loss that preserves the global hierarchical relationship and the triplet loss that improves local discriminability. These two loss are combined into the Hyperloss function, supported by theoretical foundations that demonstrate its effectiveness. Before this, we define the threat model and present the problem statement.

A. Threat Model

We assume that the attacker $\mathcal{A}$ has access to both the target device and a cloned device, enabling them to collect side-channel traces D (e.g., electromagnetic radiation or power consumption) from cryptographic operations. $\mathcal{A}$ controls the profiling phase, where they collect a set of traces D_{train} associated with input values p_i and keys k_i on the cloned device. During the attack phase, $\mathcal{A}$ only observes traces D_{attack}, which are collected from the target device using unknown keys k^* and input values p_i, without direct access to the cryptographic keys or internal states. The goal of $\mathcal{A}$ is to optimize key recovery while minimizing feature engineering and hyperparameter tuning.

B. Problem Statement and Overview

Profiled SCA exploits physical leakages, such as power consumption and electromagnetic emissions, to recover secret information from cryptographic devices. A key challenge in SCA is to learn feature representations that can distinguish closely related attack classes while maintaining robustness to variations in attack conditions, such as environmental noise, desynchronization effects, and device variability. Existing approaches, such as those based on triplet loss [32] and NLS loss [18], focus on optimizing local discriminative power by minimizing the distance between feature vectors of the same class while maximizing the distance between different classes. While effective for local discrimination, these Euclidean-based methods often fail to capture the hierarchical relationships among attack scenarios, where certain attack categories share common characteristics at different granularity levels. This limitation results in reduced attack performance and limited generalization across diverse datasets, particularly when dealing with fine-grained key distinctions and varying leakage models.

To address this limitation, we propose HypSCA, which integrates hyperbolic embedding with conventional discriminative learning. Our method proposes a dual-space optimization strategy with two complementary objectives: (1) a HIER loss that leverages hyperbolic geometry to preserve global hierarchical relationships among attack categories, and (2) a triplet loss that enhances local discriminative power in Euclidean space. This combination ensures that the learned feature space simultaneously reflects both global structure and local separability, significantly improving the model's ability to generalize across diverse attack scenarios and remain robust against common SCA challenges.

C. HypSCA: A Dual-Space Representation Learning Method

As illustrated in Fig. 2, we propose HypSCA, a unified dual-space representation learning framework for profiled SCA. Side-channel traces naturally exhibit hierarchical structures: for instance, in 8-bit implementations, traces often form coarse-grained clusters based on Hamming weights, with finer distinctions arising from specific bit patterns. To capture both global structure and local detail, HypSCA jointly embeds learned features into two distinct geometric spaces. Given a side-channel trace $T_i \in \mathcal{X}$ with label $y_i \in \mathcal{Y}$, a shared deep encoder first extracts a feature vector $x_i \in \mathbb{R}^n$. This representation is simultaneously projected into Euclidean space, where a triplet loss enforces local discriminability, and into hyperbolic space, where exponential mapping facilitates the organization of features within a negatively curved manifold that captures hierarchical relationships.

This dual-space design enables HypSCA to align semantically similar traces while maximizing inter-class separation, thereby improving both attack performance and generalization. Specifically, coarse-grained structures tend to be preserved near the hyperbolic origin (e.g., via star-shaped hierarchical proxies), while fine-grained variations are distributed radially, as illustrated by the curved "pull" paths in Fig. 2. The synergy between hyperbolic and Euclidean objectives forms the basis for the unified optimization strategy introduced in the next section. To provide a more detailed understanding of the HypSCA framework, we now decompose the dual-space learning process into its two core components. We begin by discussing how hierarchical structure is preserved through hyperbolic embedding.

1) Preserving Hierarchical Structure in Hyperbolic Space: A key innovation of HypSCA is the use of hyperbolic space to explicitly capture the inherent hierarchical structure of side-channel traces, which is difficult to represent effectively with

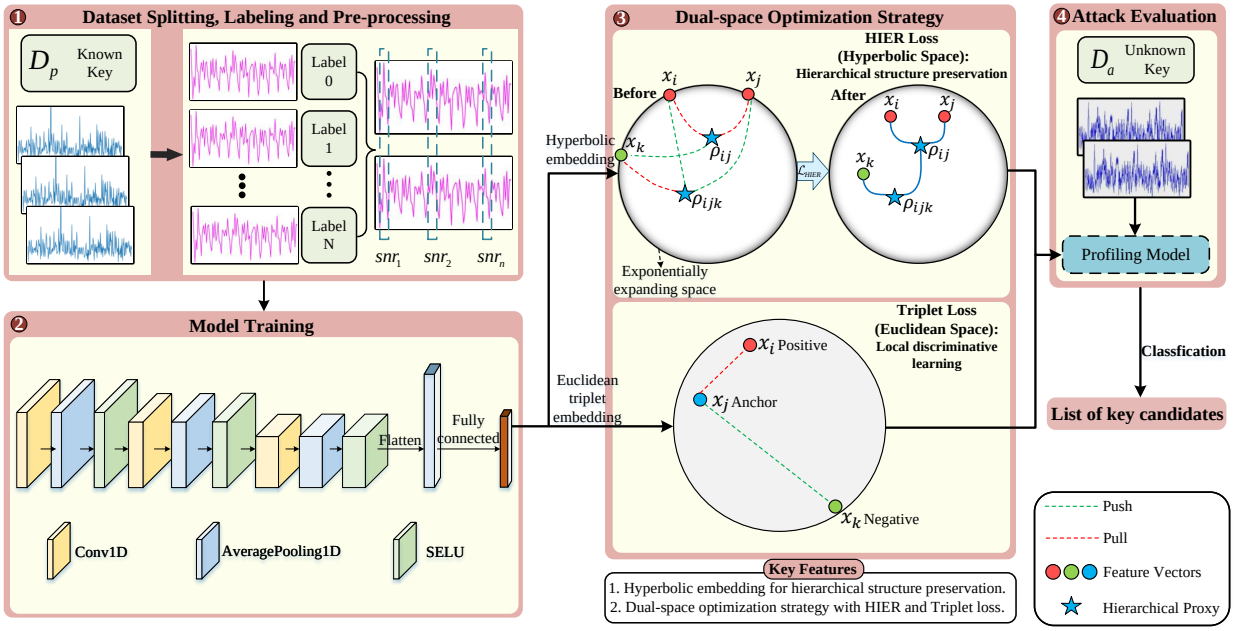


Fig. 2: Overview of the proposed HypSCA framework. The pipeline consists of four stages: (1) dataset preprocessing; (2) feature extraction via a shared encoder; (3) dual-space optimization with hyperbolic embedding for hierarchical structure preservation and Euclidean triplet loss for local discriminability; and (4) key recovery attack evaluation.

standard Euclidean geometry as illustrated in Fig.1. Hyperbolic space is a Riemannian manifold with constant negative curvature [33] that naturally encodes hierarchical relationships through exponential volume growth in low-dimensional embeddings. Specifically, the n-dimensional Poincaré ball model $(\mathbb{D}_c^n, g^{\mathbb{D}})$ is defined as:

$$\mathbb{D}_c^n = \{\mathbf{x} \in \mathbb{R}^n : c||\mathbf{x}||^2 < 1\}, \tag{5}$$

where $c > 0$ is the curvature parameter. The Riemannian metric $g^{\mathbb{D}}$ is scaled by a conformal factor $\lambda_c = \frac{2}{1-c||\mathbf{x}||^2}$, making the Poincaré ball a non-Euclidean space. This conformal structure causes distances to grow exponentially as points approach the boundary of the ball, which is essential for modeling hierarchical relationships.

In the context of side-channel traces, the exponential volume growth corresponds to the nested parent-child and ancestor-descendant relationships intrinsic to cryptographic operations, enabling a compact yet expressive embedding space, as illustrated in Fig. 1(b). For example, traces belonging to the same class cluster tightly around their corresponding class-specific proxy nodes (parent nodes). These proxy nodes act as hierarchical representatives, summarizing the shared characteristics of their child traces. This arrangement reflects the intrinsic nested hierarchy of cryptographic operations, where fine-grained trace details aggregate into coherent class structures while maintaining clear separation between different classes. To embed Euclidean feature vectors $x_i \in \mathbb{R}^n$ into hyperbolic space, HypSCA employs the exponential map centered at the origin, projecting x_i into the Poincaré ball as follows:

$$\exp_0(x_i) = \tanh\left(\sqrt{c}||x_i||\right)\frac{x_i}{\sqrt{c}||x_i||}. \tag{6}$$

Once mapped, relationships in hyperbolic space are measured using the hyperbolic distance, defined for any two points $x_i, x_j \in \mathbb{D}_c^n$ as:

$$d_H(x_i, x_j) = \frac{2}{\sqrt{c}}\text{arctanh}\big(\sqrt{c}||-x_i \oplus_c x_j||\big), \tag{7}$$

where $\oplus_c$ denotes Möbius addition [24]. Unlike Euclidean distance, hyperbolic distance expands exponentially near the boundary of the Poincaré ball, enabling it to naturally capture hierarchical relationships in $\mathcal{X}$, as illustrated in Fig. 1(b). As the curvature parameter c approaches zero, hyperbolic distance converges smoothly to the Euclidean metric.

To organize trace features hierarchically, we adapt HIER loss [26], which incorporates hierarchical clustering to establish similarity relationships. Dasgupta [34] defined a cost function based on the lowest common ancestor (LCA) of data points:

$$\mathcal{C} := \sum_{x_i, x_j \in \mathcal{X}} w_{i,j}|\text{leaves}(x_i \vee x_j)|, \tag{8}$$

where $w_{i,j}$ represents the similarity between two feature vectors x_i and x_j, $x_i \vee x_j$ denotes their LCA, and $\text{leaves}(x_k)$ is the set of leaves in the subtree rooted at x_k.

Building on the hierarchical relationships modeled in hyperbolic space, HIER loss operates on triplets of feature vectors $\{x_i, x_j, x_k\}$, where x_i and x_j are semantically related, and x_k is unrelated. The method enforces that x_i and x_j share the same LCA in the hierarchy, while x_k is assigned a different ancestor. This hierarchy is constructed without predefined annotations, making it highly adaptable to profiled SCA. To optimize the hierarchical structure, we define triplet sampling using K-reciprocal nearest neighbors in hyperbolic space $\mathbb{D}_c^n$:

$$\mathcal{T} = \{(x_i, x_j, x_k) \mid (x_j \in R_K(x_i)) \wedge (x_k \notin R_K(x_i))\}, \tag{9}$$

where $R_K(x_i)$ is the K-reciprocal nearest neighbors of x_i.

Hierarchical proxies $\rho \in P$ act as learnable ancestors in hyperbolic space. The probability that a proxy ρ is the LCA of x_i and x_j is given by:

$$\pi_{ij}(\rho) = \exp\Big(-\max\big\{d_H(x_i, \rho), d_H(x_j, \rho)\big\}\Big), \tag{10}$$

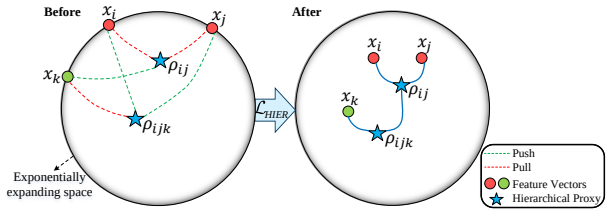


Fig. 3: Illustration of the hierarchical component in HypSCA. Before optimization (left), feature vectors (colored circles) and hierarchical proxies (blue stars ★) are randomly distributed in the hyperbolic space. The HIER loss attracts positive samples (green dashed lines) toward their corresponding proxy while repelling negative samples (red dashed lines). After optimization (right), embeddings form a structured hierarchy where samples cluster around their class-specific proxies.

where $d_H(\cdot,\cdot)$ is the hyperbolic distance. The most likely LCA ρ_{ij} is sampled using the Gumbel-max trick [35]:

$$\rho_{ij} = \arg\max_{\rho} \left(\pi_{ij}(\rho) + g_{ij}\right), \quad g_{ij} \sim \text{Gumbel}(0,1). \tag{11}$$

Using these elements, we define the hierarchical component of our loss function to encourage semantically related feature vectors x_i and x_j to be closer to their shared LCA ρ_{ij}, while ensuring that unrelated vector x_k is closer to its own proxy ρ_{ijk}:

$$\begin{aligned}\mathcal{L}_{HIER} = &\left[d_H(x_i,\rho_{ij}) - d_H(x_i,\rho_{ijk}) + \delta_H\right]_+ \\ &+ \left[d_H(x_j,\rho_{ij}) - d_H(x_j,\rho_{ijk}) + \delta_H\right]_+ \\ &+ \left[d_H(x_k,\rho_{ijk}) - d_H(x_k,\rho_{ij}) + \delta_H\right]_+,\end{aligned} \tag{12}$$

where δ_H is the margin in hyperbolic space and $[\cdot]_+$ denotes the ReLU function. This margin δ_H controls the separation between related and unrelated points in hyperbolic space, ensuring that points with hierarchical similarities are embedded closer to their shared ancestor. This loss ensures that related feature vectors x_i and x_j are closer to their shared LCA ρ_{ij} than to a higher-level ancestor ρ_{ijk}, while the unrelated feature vector x_k is closer to ρ_{ijk} than to ρ_{ij}. A representation of this hierarchical component is shown in Fig. 3.

2) Enhancing Local Discriminability in Euclidean Space: While hyperbolic representations excel at capturing hierarchical structure, they can sometimes sacrifice fine-grained discriminability needed for precise key recovery in SCA. To address this, we complement our hyperbolic approach with discriminative learning in Euclidean space through an enhanced triplet loss. The standard triplet loss, refined by Schroff et al. [32], employs a triplet-based input (anchor, positive, negative) and is defined as:

$$\mathcal{L}_{triplet} = \max\left(0, d_E(f(x_i), f(x_j)) - d_E(f(x_i), f(x_k)) + \delta_E\right), \tag{13}$$

where $d_E(\cdot,\cdot)$ denotes the Euclidean distance, and δ_E is the margin.

To better handle the intricate correlations in SCA where similar keys often produce similar traces, we adapt the hybrid distance metric proposed by Wu et al. [17] that incorporates both embedding distances and label similarities:

$$d_{\text{hybrid}}(x,y) = \frac{d_E^2(f(x), f(y))}{\alpha^{d'(l_x,l_y)}}, \quad \alpha \in (0,1], \tag{14}$$

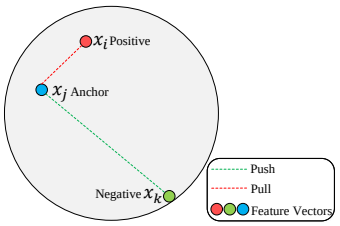


Fig. 4: Illustration of the discriminative component in HypSCA. The anchor sample x_j (blue) is pulled closer to the positive sample x_i (red) while being pushed away from the negative sample x_k (green), enhancing local feature separability.

where $d_E^2(\cdot)$ is the squared Euclidean distance between embeddings, $d'(l_x, l_y)$ is the normalized distance between labels, and α controls sensitivity to label differences. This formulation reduces distances between embeddings with similar labels and increases distances for dissimilar labels. By incorporating this hybrid distance, our discriminative loss component becomes:

$$\begin{aligned}\mathcal{L}_{triplet} = \max\big(0, &d_{\text{hybrid}}(f(x_i), f(x_j)) \\ &- d_{\text{hybrid}}(f(x_i), f(x_k)) + \delta_E\big).\end{aligned} \tag{15}$$

This enhanced triplet loss ensures fine-grained discriminability of features in Euclidean space, complementing the broader hierarchical organization captured in hyperbolic space. Fig. 4 illustrates this discriminative component of our approach.

Having established the two complementary components of our method, we now present how these components are integrated through a unified optimization strategy.

D. HyperLoss: Unified Dual-Space Optimization Strategy

While the hierarchical and discriminative components each address specific aspects of SCA representation learning, their true power emerges when combined synergistically. To achieve this, we propose HyperLoss, a unified optimization objective that balances global hierarchical structure preservation with local discriminative capability:

$$\mathcal{L}_{Hyper} = \lambda_{hier} \cdot \mathcal{L}_{HIER} + \lambda_{triplet} \cdot \mathcal{L}_{triplet}, \tag{16}$$

where λ_{HIER} and $\lambda_{triplet}$ are weighting hyperparameters that control the relative contributions of each component.

The balance between these components is crucial for optimal performance across different SCA scenarios. The hierarchical component $\mathcal{L}_{HIER}$ enables the model to capture broad patterns and relationships among different attack classes, particularly beneficial for generalization across diverse datasets and attack conditions. Meanwhile, the discriminative component $\mathcal{L}_{triplet}$ ensures that fine-grained distinctions critical for precise key recovery are preserved in the feature space. By dynamically balancing these hyperparameters, HyperLoss can adapt to diverse SCA requirements. When facing highly complex hierarchical relationships among attack traces, increasing λ_{hier} emphasizes the global structure preservation. Conversely, for scenarios requiring fine-grained discrimination between very similar keys (for instance, when key bytes differ by only a few bits), increasing $\lambda_{triplet}$ prioritizes local discriminability.

Through extensive experimentation, we have determined effective ranges for these hyperparameters across common SCA scenarios, providing practical guidance for implementation. The unified HyperLoss approach creates a more comprehensive feature space that simultaneously captures both the hierarchical relationships between trace categories and the precise discriminative features needed for effective key recovery.

This dual-space optimization strategy represents the core innovation of our HypSCA method, enabling substantial improvements in attack performance and generalization capabilities compared to single-space optimization methods. In the following section, we provide a theoretical analysis of HypSCA.

E. Theoretical Analysis of HypSCA

In this section, we provide a deeper theoretical analysis of the proposed HypSCA framework. Our dual-space optimization strategy integrates hyperbolic embedding with Euclidean-based learning to achieve a balance between global hierarchical structure preservation and local discriminability, which are two critical aspects for effective SCA.

1) Information-Theoretic Perspective of Hyperbolic Embedding: HIER loss preserves hierarchical relationships among attack traces by embedding them into hyperbolic space, offering specific advantages for SCA through two complementary mechanisms:

Information organization and uncertainty reduction: The HIER loss operates by simultaneously reducing entropy within hierarchical clusters while maximizing divergence between different clusters. This dual effect creates embeddings that are both compact within each attack category and well-separated between different categories. The hyperbolic geometry naturally amplifies separation due to its exponential expansion property, allowing for better preservation of the global hierarchical structure while minimizing uncertainty within each cluster.

Why hyperbolic space excels for SCA: Hyperbolic geometry is particularly well-suited for SCA due to its inherent properties: *exponential capacity*, as hyperbolic spaces can embed hierarchical structures with lower distortion than Euclidean spaces of the same dimension [24], enabling more efficient representation of complex trace relationships; *natural hierarchy modeling*, since in SCA, attack traces form natural hierarchies where similar patterns (e.g., traces with similar Hamming weights or intermediate values) share common features at different granularity levels; and *desynchronization handling*, where the hierarchical embedding helps maintain structural relationships even when traces are temporally desynchronized, providing implicit robustness against desynchronization effects, which is a common challenge in practical SCA scenarios.

2) Complementary Role of Euclidean Discrimination: While hyperbolic embedding excels at preserving global structure, Euclidean-based triplet loss complements it by refining local discriminability. This combination addresses the full spectrum of SCA challenges where both global patterns and fine-grained differences contain crucial information.

Geometric principles and SCA benefits: Triplet loss enforces that for any triplet (x_i, x_j, x_k) where x_i (anchor) and x_j (positive) belong to the same class, and x_k (negative) belongs to a different class:

$$d_{\mathrm{E}}(f(x_i), f(x_j)) + \delta_E \leq d_{\mathrm{E}}(f(x_i), f(x_k)), \qquad (17)$$

where d_{E} is the Euclidean distance and $\delta_E > 0$ is a margin parameter.

This approach provides several SCA-specific advantages: it enhances sensitivity to subtle leakage differences that may indicate different key values, creates clear decision boundaries between key-dependent trace classes, and improves robustness against noise variations by focusing on relative distances rather than absolute values.

3) Dual-space Optimization Strategy: The core innovation of HypSCA lies in its dual-space optimization strategy that leverages complementary geometric properties for SCA:

- *Global hierarchical modeling* (Hyperbolic Space): Captures the natural organization of attack traces, preserving relationships at different granularity levels.
- *Local discriminative refinement* (Euclidean Space): Ensures fine-grained separation between similar traces with different key-related information.
- *Joint optimization*: The combined loss function: $\mathcal{L}_{Hyper}$ balances these complementary objectives to simultaneously preserve hierarchical structure and maintain local discriminability.

4) Theoretical Advantages for SCA: The theoretical advantages of HypSCA can be summarized as follows:

- *Hierarchical generalization*: HIER loss preserves the global structure of attack categories, enabling the model to generalize to more complex SCA scenarios.
- *Local robustness*: Triplet loss ensures fine-grained separation of samples, reducing intra-class variance and inter-class confusion for better key discrimination.
- *Desynchronization resilience*: The dual-space optimization strategy effectively mitigates temporal misalignment and jitter in attack traces.
- *Balanced representations*: HypSCA achieves a balance between structure preservation and sample-level discrimination, resulting in efficient and discriminative feature embeddings.

These advantages explain why HypSCA consistently outperforms conventional approaches in our experiments.

F. Neural Network Architecture

This paper employs a convolutional neural network (CNN) for side-channel trace classification. The model is implemented in Python 3.9 using TensorFlow 2.10.0 and trained on an NVIDIA RTX 4060 Ti GPU to efficiently handle large datasets.

The network consists of two convolutional blocks, followed by the flatten and fully connected layers. Each convolutional block includes a convolutional layer with SeLU activation and an average pooling layer for dimensionality reduction. The flattening layer converts the feature maps into a one-dimensional vector, which is passed to a fully connected layer

TABLE II: Network Architecture and Training Configuration

Component	Configuration
Input	
Input Shape	(4000, 1) or (1250, 1) depending on dataset
Convolutional Block 1	
Layer Type	1D Convolution
Filters	64
Kernel Size	15
Activation	SeLU
Pooling	AvgPool1D (pool size = 15)
Convolutional Block 2	
Layer Type	1D Convolution
Filters	128
Kernel Size	3
Activation	SeLU
Pooling	AvgPool1D (pool size = 2)
Fully Connected Layer	
Layer Type	Dense (Fully Connected)
Neurons	32
Activation	None
Training Configuration	
Optimizer	Adam
Loss Function	$\mathcal{L}_{Hyper}$ (joint HIER + triplet loss)
Learning Rate	5×10^{-4}
Batch Size	512

with 32 neurons for classification. The Adam optimizer is used to train the model, and the $\mathcal{L}_{Hyper}$ loss function is applied to improve the separability of leakage classes in the feature space. The detailed architecture is shown in TABLE II.

V. Experiment

In this section, we first present the experiment setup, including a description of datasets, leakage models, and evaluation metrics. Then, we show experimental results concerning synchronized traces, desynchronized traces, and the evaluation of hyperparameters.

A. Experiment Setup

1) Datasets: We conduct our experiments using three widely adopted datasets in the SCA community: ASCAD_F, ASCAD_R [36], and AES_HD [37]. These datasets are selected due to their diversity in terms of masking countermeasures, key randomness, and the nature of the traces (software vs. hardware implementations). Below is a detailed description of each dataset:

- *ASCAD_F:* This dataset features AES-128 electromagnetic traces collected from an 8-bit AVR microcontroller with a fixed key. It contains 50,000 traces for profiling and 10,000 traces for the attack phase. Each trace originally includes 700 features, but we extend the time window to include 4,000 features, focusing on capturing the Signal-to-Noise Ratio (SNR) peaks associated with both the secret shares $s_{r,2}$ and r_2, to better target the processing of the third masked key byte.
- *ASCAD_R:* This version of the dataset also consists of EM traces but uses random keys. It includes 200,000 traces for profiling and 100,000 traces for the attack phase. To capture more detailed information, we expand the pre-selected window based on the SNR of the S-box output, resulting in 4,000 features per trace. For consistency across experiments, we downsample the dataset to 50,000 profiling traces and 10,000 attack traces, similar to ASCAD_F.
- *AES_HD:* This dataset represents an unprotected hardware implementation of AES-128. The traces are collected using a near-field electromagnetic probe positioned above a decoupling capacitor on the power line of an FPGA board. The Hamming distance (HD) leakage model is employed here, focusing on the expression $Sbox^{-1}(c_7 \oplus k_7) \oplus c_{11}$ during the final round of AES. The dataset contains 45,000 traces for profiling and 5,000 traces for the attack phase, with each trace consisting of 1,250 features.

These datasets provide a comprehensive basis for evaluating the model across different attack scenarios, including software and hardware implementations, as well as traces with fixed and random keys.

2) Leakage Models: In our research, we focus on two leakage models to evaluate the model's performance across different scenarios. First, the Hamming Weight (HW) model assumes that the leakage correlates with the Hamming weight of the sensitive variable. For a single intermediate byte in AES, the HW model divides the byte into nine possible classes. Second, the Identity (ID) model treats the actual intermediate values as the leakage, resulting in 256 distinct classes for each byte. This model provides a more granular representation of the leakage but is inherently more challenging due to the larger number of classes. The HW model is more common in practical SCA tasks, whereas the ID model offers a more stringent evaluation of the model's classification capability.

3) Evaluation Metrics: In this study, we use the T_{GE0} metric [19] to evaluate the effectiveness of our attack model. The T_{GE0} metric represents the number of attack traces required to reduce the Guessing Entropy (GE) to zero. Specifically, it measures the minimum number of traces needed such that the correct key is the most probable in the key-guessing vector, meaning the correct key has been identified as the top candidate.

The T_{GE0} metric is particularly useful in real-world SCA scenarios as it directly reflects the efficiency of an attack. A lower T_{GE0} value indicates that fewer traces are required to recover the secret key, thus signifying a more efficient attack. In all of our experiments, the model's performance is evaluated by determining the number of attack traces needed to achieve T_{GE0}. This metric allows us to compare the effectiveness of our approach across different datasets and configurations.

B. Experimental Results

Two leakage models, HW and ID, are considered for the ASCAD_F and ASCAD_R datasets, whereas only the HW leakage model is considered for the AES_HD dataset. Initially, we showcase the outcomes derived from publicly accessible primary datasets. Afterward, we evaluate the robustness against perturbations across varying desynchronization levels. In line with previous research, to ensure representative

TABLE III: Median T_{GE0} of DLSCA methods on synchronized traces across datasets

Dataset	Leakage	[13]	[15]	$\mathcal{L}_{CCE}$ [14]	$\mathcal{L}_{CER}$ [7]	$\mathcal{L}_{RKL}$ [8]	$\mathcal{L}_{FLR}$ [9]	$\mathcal{L}_{triplet}$ [17]	$\mathcal{L}_{FLR+softNN}$ [16]	$\mathcal{L}_{FLR+center}$ [16]	$\mathcal{L}_{NLS}$ [18]	[12]	$\mathcal{L}_{Hyper}$ This work
ASCAD_F	HW	174	×	>2000	540	>2000	500	159	832	790	122	544	**93**(23.8%)
	ID	191	80	963	600	1760	520	64	716	3867	50	87	**44**(12.0%)
ASCAD_R	HW	×	×	2840	950	>3000	800	197	2592	3867	138	650	**110**(20.3%)
	ID	×	×	>3000	>3000	>3000	>3000	188	>10000	9681	105	487	**91**(13.3%)
AES_HD	HW	4415	×	×	×	×	×	1768	×	×	1252	×	**974**(22.2%)

Note: Light green indicates the best result, light yellow indicates the second-best result, and "×" indicates invalid data. Numbers in parentheses in the last column indicate the relative reduction (%) in the number of attack traces required by Hyperloss, compared to the best-performing baseline in each row.

results and fair comparisons, all experimental scenarios are independently trained, executed, and attacked 20 times. We select the median of T_{GE0} from 20 attacks to evaluate attack performance.

1) Results on Synchronized Traces: This section compares the attack performance of Hyperloss with recent state-of-the-art DLSCA methods on synchronized traces, as summarized in TABLE III. The performance is reported using the median T_{GE0} across three datasets (ASCAD_F, ASCAD_R, AES_HD) and two leakage models (HW and ID). The best performances are highlighted in green, the second-best in yellow, and "×" indicate invalid cases where GE did not reach zero. All models were evaluated under consistent neural network configurations, with dataset-specific feature dimensions (700 for ASCAD_F, 1400 for ASCAD_R, and 1250 for AES_HD).

The experimental results show that Hyperloss achieves the best or near-best performance consistently across all scenarios. These improvements highlight HypSCA's ability to balance global structural modeling and local feature discriminability, both critical for profiled SCA. By incorporating HIER loss and triplet loss, HypSCA uncovers latent semantic hierarchies while enhancing the discriminative power of local features. This dual-space optimization strategy enables HypSCA to extract robust and meaningful embeddings, leading to significant reductions in T_{GE0} compared to state-of-the-art methods. The following sections provide a detailed performance analysis across different datasets and leakage models.

- *Performance on the ASCAD_F dataset:* Under the HW leakage model, HypSCA achieves a T_{GE0} of 93, surpassing all baseline methods: the triplet model (159), $\mathcal{L}_{FLR}$ (500), [13] (174), and [12] (544). Compared to the second-best method ($\mathcal{L}_{NLS}$ with 122), this represents a 23.8% reduction in the number of required traces. Under the ID model, HypSCA achieves the best T_{GE0} of 44, outperforming [12] (87), the triplet model (64), and $\mathcal{L}_{CCE}$ (963). This corresponds to a 12.0% improvement over the second-best method ($\mathcal{L}_{NLS}$ with 50).
- *Performance on the ASCAD_R dataset:* For the HW leakage model, HypSCA achieves a T_{GE0} of 110, outperforming the triplet model (197), $\mathcal{L}_{CER}$ (950), and [12] (650). Compared to the second-best method ($\mathcal{L}_{NLS}$ with 138), this results in a 20.3% reduction in trace requirement. Under the ID model, HypSCA again ranks first with 91, surpassing all baselines including the triplet model (188), [12] (487), and $\mathcal{L}_{NLS}$ (105), achieving a 13.3% reduction compared to the second-best.
- *Performance on the AES_HD dataset:* For the HW leakage model, HypSCA achieves a T_{GE0} of 974, outperforming the triplet baseline (1768) and $\mathcal{L}_{NLS}$ (1252). Compared to the second-best approach ($\mathcal{L}_{NLS}$), this corresponds to a 22.2% reduction in required traces. Other methods either fail to provide valid results or perform significantly worse (e.g., $\mathcal{L}_{CCE}$ with 4415).

To further assess the effectiveness of HypSCA under aligned trace conditions, we compare its performance with the top two baseline methods across multiple datasets and leakage models. As reported in the results, HypSCA achieves an average improvement of **30.1%** in reducing the number of attack traces required to reach $GE = 0$, with improvements ranging from **12.0%** to **51.6%**. The superiority of HypSCA is consistently observed under both HW and ID leakage models, demonstrating its ability to outperform state-of-the-art approaches even in well-aligned scenarios.

Notably, HypSCA exhibits exceptional performance on challenging datasets such as *ASCAD_R* and *AES_HD*, where many existing methods fail to generalize effectively. This robustness can be attributed to the design of the proposed HIER loss, which models latent semantic hierarchies within hyperbolic space. By capturing both inter-class separability and intra-class compactness, the HIER loss enables the network to construct structured and informative representations, thereby facilitating better generalization across diverse datasets and leakage conditions. Furthermore, the integration of triplet loss complements HIER loss by enhancing local feature discrimination, achieving a balance between global structure modeling and local separability.

In contrast, traditional techniques such as $\mathcal{L}_{CCE}$ and $\mathcal{L}_{FLR}$ often fail to produce valid results in complex or desynchronized scenarios, underscoring their limited generalization capability. HypSCA, by contrast, consistently delivers stable and accurate performance across both simple and complex settings. These characteristics collectively establish HypSCA as a robust and effective method for profiled SCA, setting a new benchmark for deep learning-based SCA methodologies.

2) Results on Desynchronized Traces: Attacks on desynchronized traces are crucial for SCA. Typically, trace values in desynchronized traces are less correlated with intermediate data than those in synchronized traces. Similarly to prior research, we simulated and tested two levels of desynchronization (50 and 100) to showcase the effects of tracking

TABLE IV: Median T_{GE0} on DLSCA methods on desynchronized traces across datasets

Dataset	Leakage	[13]	[11]	$\mathcal{L}_{CCE}$ [14]	$\mathcal{L}_{FLR}$ [9]	$\mathcal{L}_{triplet}$ [17]	$\mathcal{L}_{FLR+softNN}$ [16]	$\mathcal{L}_{FLR+center}$ [16]	$\mathcal{L}_{NLS}$ [18]	$\mathcal{L}_{Hyper}$ This work
ASCAD_F_desync50	HW	×	×	4107	5062	251	1685	2021	190	**171**(10.0%)
	ID	×	2850	>10000	×	191	6798	>10000	141	**107**(24.1%)
ASCAD_F_desync100	HW	×	×	8652	5277	382	4862	4406	338	**283**(16.3%)
	ID	×	×	×	×	582	×	×	480	**374**(22.1%)
ASCAD_R_desync50	HW	×	7805	>20000	>20000	2251	×	×	1701	**1606**(5.6%)
	ID	×	3715	×	×	3385	×	×	**2919**	3697
ASCAD_R_desync100	HW	×	×	>20000	16344	6386	>20000	16209	5925	**4432**(25.2%)
	ID	×	×	×	>20000	9932	×	×	9899	**8270**(16.5%)

Note: Light green indicates the best result, light yellow indicates the second-best result, and "×" indicates invalid data. Numbers in parentheses in the last column indicate the relative reduction (%) in the number of attack traces required by HypSCA, compared to the best-performing baseline in each row.

desynchronization. The training parameters in this section align with those in Section V-B1. Introducing noise will further augment the difficulty of attacks and render certain attack methods ineffective.

TABLE IV show the median attack performance obtained from 20 independent training runs, with the corresponding models referred to as median models. The comparison of experimental results obtained without adding desynchronization noise highlights the significant impact of temporal perturbations on the attack performance of traditional deep learning-based attack methods. Furthermore, in scenarios with trace desynchronization, traditional dimensionality reduction techniques (LDA, PCA, ELLE) encounter greater difficulty in extracting informative features. [17] utilized the triplet method to accomplish a successful SCA attack by effectively compromising the target in diverse attack scenarios, thereby validating the efficacy of the DML based attack method in such contexts. Moreover, HypSCA outperforms the findings of [17] when applied to public datasets featuring diverse desynchronization levels and leakage models. The T_{GE0} indicators exhibit a significant decrease, underscoring the superiority of the HypSCA proposed in this study for datasets with adding desynchronization noise.

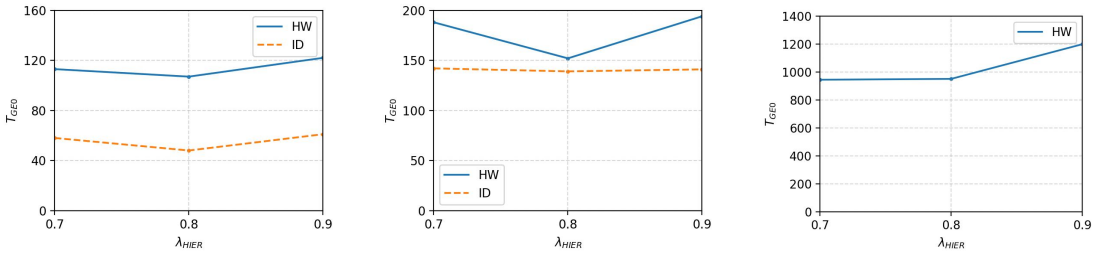
- *Performance on the ASCAD_F_desync50 dataset:* Under the HW leakage model, HypSCA achieves a T_{GE0} of 171, outperforming the second-best method $\mathcal{L}_{NLS}$ (190) with a 10.0% reduction in trace requirement. Under the ID model, HypSCA achieves a T_{GE0} of 107, significantly outperforming $\mathcal{L}_{NLS}$ (141) by 24.1%. Other methods, including $\mathcal{L}_{CCE}$ (4107) and $\mathcal{L}_{FLR}$ (5062), require far more traces to reach convergence.
- *Performance on the ASCAD_F_desync100 dataset:* For the HW model, HypSCA achieves a T_{GE0} of 283, reducing the number of required traces by 16.3% compared to $\mathcal{L}_{NLS}$ (338). Under the ID model, HypSCA achieves the best result with 374, outperforming the second-best method $\mathcal{L}_{NLS}$ (480) by 22.1%. In contrast, methods such as $\mathcal{L}_{CCE}$ (8652) and $\mathcal{L}_{FLR}$ (5277) perform poorly, failing to deliver practical attack results.
- *Performance on the ASCAD_R_desync50 dataset:* Under the HW model, HypSCA achieves a T_{GE0} of 1606, outperforming $\mathcal{L}_{NLS}$ (1701) with a 5.6% reduction. However, for the ID model, HypSCA records a T_{GE0} of 3697, which is 26.6% higher than $\mathcal{L}_{NLS}$ (2919), indicating a slight performance degradation in this specific setting. Nonetheless, HypSCA remains competitive, as other methods either fail to converge or require significantly more traces.
- *Performance on the ASCAD_R_desync100 dataset:* For the HW model, HypSCA achieves a T_{GE0} of 4432, outperforming the second-best method $\mathcal{L}_{NLS}$ (5925) by 25.2%. Under the ID model, HypSCA achieves 8270, reducing the trace requirement by 16.5% compared to $\mathcal{L}_{NLS}$ (9899). Other methods, such as $\mathcal{L}_{CCE}$ and $\mathcal{L}_{FLR}$, fail to reach $GE = 0$ within the maximum trace budget.

To evaluate the overall advantage of the proposed method, we compare it with the top two performing baseline methods in each task, excluding the outlier case of *ASCAD_R_desync50* under the ID leakage model. Across the remaining seven tasks involving desynchronized traces, HypSCA achieves an average improvement of **23.4%** in reducing the number of attack traces required to reach $GE = 0$. The observed improvement ranges from **5.6%** to **44.0%**, highlighting the method's consistent advantage under most desynchronized conditions. These results validate the effectiveness and robustness of HypSCA in addressing misalignment challenges that frequently occur in real-world side-channel scenarios.

Moreover, the performance of HypSCA remains superior across a wide range of datasets and leakage models. Compared to the Triplet model, HypSCA consistently requires fewer traces to reach $GE = 0$, thanks to the joint optimization of global and local feature structures. In addition, it significantly outperforms other baselines such as $\mathcal{L}_{CCE}$ and $\mathcal{L}_{FLR}$, which often require orders of magnitude more traces to achieve the same level of key recovery success. The ability of HypSCA to maintain high attack performance across both HW and ID leakage models, even under severe desynchronization, underscores its practical robustness and generalization capability. These characteristics establish HypSCA as a reliable and powerful method for DLSCA in real-world environments.

C. Hyperparameter Evaluation

This section investigates the influence of several key hyperparameters on the performance of HypSCA. A summary of the



(a) ASCAD_F (b) ASCAD_R (c) AES_HD

Fig. 5: Quantitative impact of λ_{HIER} on attack performance (T_{GE0}) across different datasets and leakage models.

overall hyperparameter search space is provided in TABLE V, covering architectural, geometric, and training-related factors. For each hyperparameter, we evaluate its effect on the attack efficiency using the T_{GE0} metric across multiple datasets and leakage models. The analysis focuses on six parameters: the hierarchical and triplet loss weights (λ_{HIER} and $\lambda_{triplet}$), the hybrid distance weight α, the angular and margins (δ_H and δ_E), and the number of proxies. Detailed results and empirical trends are discussed in the following subsections.

TABLE V: Hyperparameter search space

Hyperparameter	**Range**
λ_{HIER}	0.7, 0.8, 0.9
λ_{triplet}	0.1, 0.2, 0.3
Hybrid distance weight α	0.1 to 1.0 (step 0.1)
Margin δ_H	0.05 to 0.30 (step 0.05)
Margin δ_E	0.1 to 1.0 (step 0.1)
Number of proxies	4 to 24

1) λ_{HIER} and $\lambda_{triplet}$: This section investigates the impact of λ_{HIER} on the overall attack performance. Given the constraint $\lambda_{HIER} + \lambda_{triplet} = 1.0$, the evaluation focuses solely on λ_{HIER}, as varying this parameter implicitly adjusts $\lambda_{triplet}$. Fig. 5 illustrates the variation of the T_{GE0} metric across different values of λ_{HIER} for three datasets and two label models.

As shown in Fig. 5a, the attack performance on ASCAD_F is relatively stable across different λ_{HIER} settings. The HW model exhibits minor fluctuations, while the ID model remains largely unaffected. In contrast, Fig. 5b demonstrates that for ASCAD_R, both HW and ID models achieve optimal performance when $\lambda_{HIER} = 0.8$, with a noticeable degradation observed at $\lambda_{HIER} = 0.7$ and 0.9. This suggests the presence of a trade-off between the hierarchical and triplet constraints. For the AES_HD dataset in Fig. 5c, a more significant sensitivity to λ_{HIER} is observed, with performance deteriorating as the value deviates from 0.8 under the HW model.

In general, setting $\lambda_{HIER} = 0.8$ provides a favorable balance between the two loss modules across all evaluated scenarios. While the influence of λ_{HIER} is less pronounced compared to other key hyperparameters such as network architecture or loss function design, appropriate tuning remains essential, particularly in desynchronized environments.

2) Number of Proxies: This section investigates the impact of the number of proxies on attack performance. Fig. 6 shows the T_{GE0} values across varying proxy counts (from 4 to 24) on three datasets and two label models. On ASCAD_F Fig. 6a, both HW and ID models exhibit limited sensitivity to this parameter, with only minor fluctuations. In ASCAD_R Fig. 6b, performance varies more significantly: the HW model achieves optimal results at 12 proxies, while the ID model performs best at 16. For AES_HD Fig. 6c, the HW model shows moderate sensitivity, with the lowest T_{GE0} observed at 8 proxies.

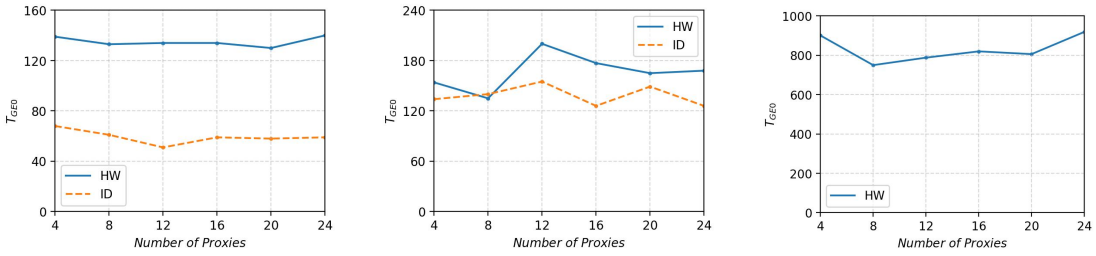
In general, using a moderate number of proxies (between 8 and 16) results in more consistent performance across datasets. Although this parameter does not dominate attack effectiveness, extreme settings may negatively affect convergence or separability in the embedding space.

3) Hybrid Distance Weight α: This section examines the effect of the hybrid distance weight α on attack performance. As shown in Fig. 7, α controls the balance between Euclidean and angular components in the hybrid distance metric. In ASCAD_F Fig. 7a, both HW and ID models maintain relatively stable T_{GE0} values when $\alpha \leq 0.7$, but performance degrades sharply as α approaches 0.9. On ASCAD_R Fig. 7b, a similar trend is observed: both models achieve lower T_{GE0} values when α is between 0.3 and 0.5, with a marked increase beyond 0.7. For the AES_HD dataset Fig. 7c, the HW model performs best at $\alpha = 0.5$, and degrades significantly at higher values.

Overall, the results suggest that setting α within the range of 0.3 to 0.5 yields more favorable and consistent performance across datasets. Excessively high values of α may overweight the angular component, leading to reduced discriminability in the learned embeddings.

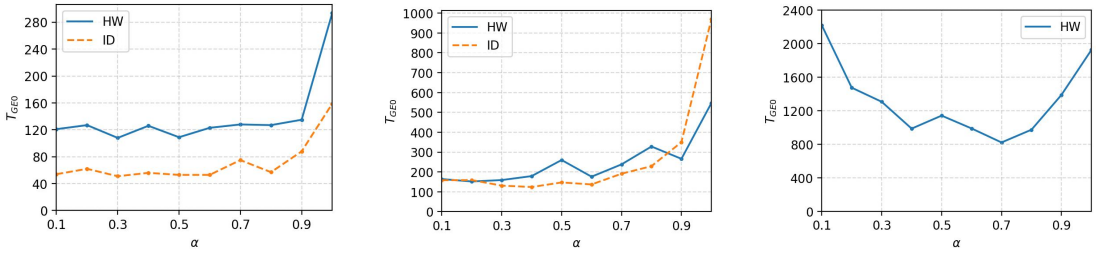
4) Margin δ_H and δ_E: This section investigates the influence of the margin parameters δ_H and δ_E on attack performance. The parameter δ_H represents the margin in hyperbolic space. As shown in Fig. 8a–8c. δ_H exhibits limited impact on the ASCAD_F and AES_HD datasets, where both HW and ID models maintain relatively stable T_{GE0} values across the tested range (0.05 to 0.30). In contrast, slightly more variation is observed on ASCAD_R Fig. 8b, where both models perform best around $\delta_H = 0.20$. However, the overall sensitivity to δ_H remains low, and no consistent degradation is observed at extreme values within the studied range.

The parameter δ_E defines the Euclidean margin in the proxy-based embedding. As presented in Fig. 8d–8f, the effect of δ_E is more pronounced. On ASCAD_F Fig. 8d, the HW model shows increased T_{GE0} when $\delta_E \geq 0.7$, while the ID model remains relatively robust. In ASCAD_R Fig. 8e,



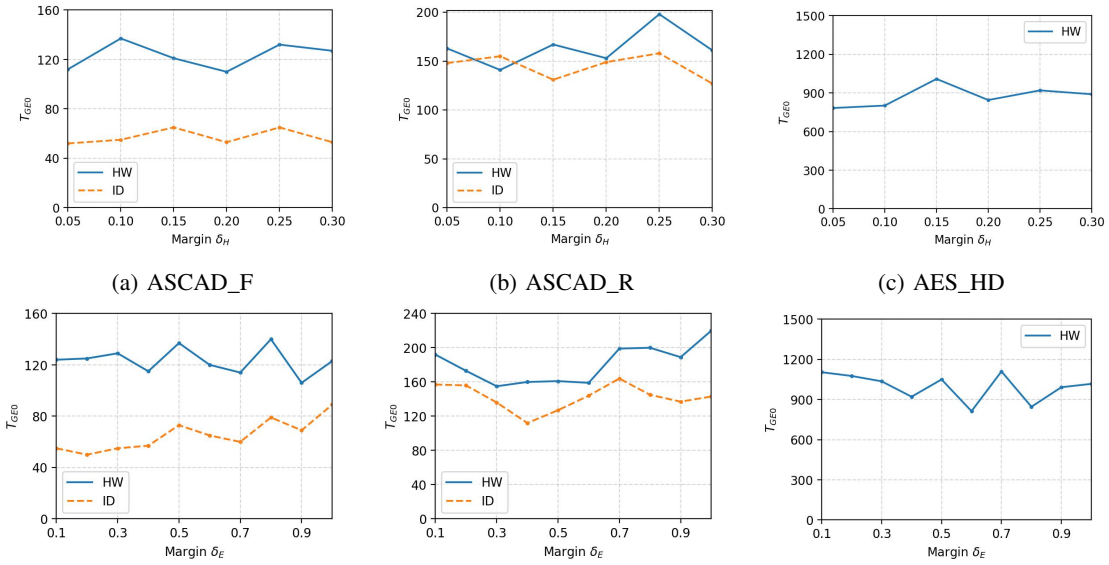
(a) ASCAD_F (b) ASCAD_R (c) AES_HD

Fig. 6: Quantitative impact of Number_of_Proxies on attack performance (T_{GE0}) across different datasets and leakage models.



(a) ASCAD_F (b) ASCAD_R (c) AES_HD

Fig. 7: Quantitative impact of hybrid distance weight α on attack performance (T_{GE0}) across different datasets and leakage models.



(a) ASCAD_F (b) ASCAD_R (c) AES_HD

(d) ASCAD_F (e) ASCAD_R (f) AES_HD

Fig. 8: Quantitative impact of Margin δ_H and δ_E on attack performance (T_{GE0}) across different datasets and leakage models.

both models obtain their lowest T_{GE0} at $\delta_E = 0.5$, with performance degrading at both lower and higher margins. The AES_HD dataset Fig. 8f exhibits moderate fluctuations, with improved performance around $\delta_E = 0.5$ and increased instability when $\delta_E > 0.7$.

In summary, both margin parameters affect the geometry of the learned embedding space, but δ_E exhibits stronger influence on attack performance across datasets. While δ_H appears relatively robust to variation, improper choices of δ_E may impair intra-class compactness or inter-class separation, particularly under desynchronized conditions.

VI. Conclusion and Future Work

Profiled SCA faces two critical limitations: poor generalization across different datasets and performance bottlenecks stemming from insufficient modeling of hierarchical relationships in cryptographic implementations. These limitations significantly reduce attack performance, as most SCA methods operating in Euclidean space struggle to capture the complex hierarchical structures present in side-channel traces.

To overcome these limitations, we proposed HypSCA, a novel method leveraging hyperbolic geometry for enhanced SCA. Our dual-space optimization approach integrates hyperbolic embedding with Euclidean-based triplet loss to effectively capture both global hierarchical structures and local discriminative features in side-channel traces. Experiments on three benchmark datasets show HypSCA outperforms state-of-the-art approaches by up to 51.6% under both Hamming Weight and Identity leakage models. HypSCA demonstrates exceptional resilience against trace desynchronization and achieves superior performance with lower embedding dimensions compared to conventional Euclidean approaches, making it suitable for resource-constrained scenarios.

Future work will extend HypSCA to diverse cryptographic implementations including post-quantum schemes, explore hybrid methodologies combining hyperbolic learning with other profiling techniques, and develop more efficient feature extraction methods. We will also investigate applying dual-space optimization to non-profiled attacks and hardware countermeasure evaluation, further advancing DML for SCA.

REFERENCES

[1] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Security, Privacy, and Applied Cryptography Engineering: 6th International Conference*, 2016, pp. 3–26.

[2] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures: Profiling attacks without pre-processing," in *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2017, pp. 45–68.

[3] X. Lu, C. Zhang, P. Cao, D. Gu, and H. Lu, "Pay attention to raw traces: A deep learning architecture for end-to-end profiling attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 235–274, 2021.

[4] Y. Li, Y. Huang, Y. Guo, C. Ge, F. Kong, and Y. Ren, "Adaptive chosen-plaintext deep learning-based side-channel analysis," *IEEE Internet of Things Journal*, 2024.

[5] Y. Chen, B. Wang, C. Su, A. Li, G. Li, and Y. Tang, "Striving for understanding: Deconstructing neural networks in side-channel analysis," *Pattern Recognition*, p. 111374, 2025.

[6] S. Alabdulwahab, M. Cheong, A. Seo, Y.-T. Kim, and Y. Son, "Enhancing deep learning-based side-channel analysis using feature engineering in a fully simulated iot system," *Expert Systems with Applications*, vol. 266, p. 126079, 2025.

[7] J. Zhang, M. Zheng, J. Nan, H. Hu, and N. Yu, "A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, no. 3, pp. 73–96, 2020.

[8] G. Zaid, L. Bossuet, F. Dassance, A. Habrard, and A. Venelli, "Ranking loss: Maximizing the success rate in deep learning side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 25–55, 2021.

[9] M. Kerkhof, L. Wu, G. Perin, and S. Picek, "Focus is key to success: A focal loss function for deep learning-based side-channel analysis," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2022, pp. 29–48.

[10] G. Perin, Ł. Chmielewski, and S. Picek, "Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 337–364, 2020.

[11] J. Rijsdijk, L. Wu, G. Perin, and S. Picek, "Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 677–707, 2021.

[12] L. Wu, L. Weissbart, M. Krček, H. Li, G. Perin, L. Batina, and S. Picek, "Label correlation in deep learning-based side-channel analysis," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3849–3861, 2023.

[13] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient cnn architectures in profiling attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 1–36, 2020.

[14] M. Kerkhof, L. Wu, G. Perin, and S. Picek, "No (good) loss no gain: systematic evaluation of loss functions in deep learning-based side-channel analysis," *Journal of Cryptographic Engineering*, vol. 13, no. 3, pp. 311–324, 2023.

[15] L. Deng, L. Li, Y. Ou, J. Xiang, and S. Xia, "Tripm: a multi-label deep learning sca model for multi-byte attacks," *International Journal of Machine Learning and Cybernetics*, pp. 1–16, 2025.

[16] T. Yap, S. Picek, and S. Bhasin, "Beyond the last layer: Deep feature loss functions in side-channel analysis," in *Proceedings of the 2023 Workshop on Attacks and Solutions in Hardware Security*, 2023, pp. 73–82.

[17] L. Wu, G. Perin, and S. Picek, "The best of two worlds: Deep learning-assisted template attack," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 413–437, 2022.

[18] K. Li, Y. Liang, H. Meng, and Z. Zhou, "Deep metric learning-based side-channel analysis with improved robustness and efficiency," *Applied Intelligence*, vol. 55, no. 7, p. 712, 2025.

[19] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "Sok: Deep learning-based physical side-channel analysis," *ACM Computing Surveys.*, vol. 55, pp. 1–35, 2023.

[20] J. Wang, K. Hu, Z. Zhong, L. Sun, and Q. Huo, "Detect-order-construct: A tree construction based approach for hierarchical document structure analysis," *Pattern Recognition*, vol. 156, p. 110836, 2024.

[21] J. Yao, J. Pan, and Y. Zeng, "Gauging-δ: A non-parametric hierarchical clustering algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[22] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 6338–6347.

[23] Y. Liu, Z. He, and K. Han, "Hyperbolic category discovery," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 9891–9900.

[24] P. Fang, "Adaptive indefinite kernels in hyperbolic spaces," *Neural Networks*, vol. 181, p. 106803, 2025.

[25] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 10 023–10 044, 2022.

[26] S. Kim, B. Jeong, and S. Kwak, "Hier: Metric learning beyond class labels via hierarchical regularization," in *Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 903–19 912.

[27] A. Bdeir, K. Schwethelm, and N. Landwehr, "Fully hyperbolic convolutional neural networks for computer vision," *arXiv preprint arXiv:2303.15919*, 2023.

[28] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*. New York, NY: Springer, 2008.

[29] Y. Li, J. Zhu, Z. Liu, M. Tang, and S. Ren, "Deep learning gradient visualization-based pre-silicon side-channel leakage location," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 2340–2355, 2024.

[30] K. Papagiannopoulos, O. Glamočanin, M. Azouaoui, D. Ros, F. Regazzoni, and M. Stojilović, "The side-channel metrics cheat sheet," *ACM Computing Surveys*, pp. 1–38, 2023.

[31] Z. Yang, M. Bastan, X. Zhu, D. Gray, and D. Samaras, "Hierarchical proxy-based loss for deep metric learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1859–1868.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[33] J. W. Cannon, W. J. Floyd, R. Kenyon, W. R. Parry *et al.*, "Hyperbolic geometry," *Flavors of geometry*, vol. 31, no. 59-115, p. 2, 1997.

[34] S. Dasgupta, "A cost function for similarity-based hierarchical clustering," in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 2016, pp. 118–127.

[35] I. A. Huijben, W. Kool, M. B. Paulus, and R. J. Van Sloun, "A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1353–1371, 2022.

[36] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ascad database," *J. Cryptogr. Eng.*, pp. 163–188, 2020.

[37] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 148–179, 2019.