Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081308

ARTICLE

RFA-SCA: Robust Feature Alignment for Side-Channel Analysis via Multi-Order Moment Alignment

Yuanzhen Wang[1], Hongxin Zhang[2,3,*], Shaofei Sun[1], Yaqi Zhang[2], Xing Fang[4] and Zhi Sun[2]

[1]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China
[2]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[3]Beijing Key Laboratory of Work Safety Intelligent Monitoring, Beijing University of Posts and Telecommunications, Beijing, China
[4]Beijing Institute of Computer Technology and Applications, Beijing, China

*Corresponding Author: Hongxin Zhang. Email: hongxinzhang@bupt.edu.cn

Received: 27 February 2026; Accepted: 14 May 2026; Published: 23 July 2026

ABSTRACT: The effectiveness of profiling deep learning side-channel attacks relies on the assumption that training and attack data follow the same distribution. However, when the profiling device differs from the target device, process-voltage-temperature (PVT) variations and clock jitter countermeasures cause distribution shifts in power traces, rendering models trained on the source device ineffective on the target. Existing domain adaptation methods typically rely on a single distributional constraint without jointly constraining kernel mean embeddings and covariance structure, thus limiting their effectiveness against strong defenses such as clock jitter. We propose Robust Feature Alignment for Side-Channel Analysis (RFA-SCA), an unsupervised domain adaptation framework that aligns source and target feature distributions without requiring the target-device key during training. RFA-SCA combines three loss components. MMD loss reduces distributional discrepancy via kernel mean embeddings in a reproducing kernel Hilbert space (RKHS), CORAL loss regularizes covariance-level feature discrepancies, and conditional entropy loss optimizes decision boundaries in the target domain. In randomized attack trials using a finite number of target-domain traces on four benchmark datasets (ASCAD, CHES CTF 2018, SAKURA-G, XMEGA), RFA-SCA achieves 100% key recovery across all six cross-device and countermeasure scenarios. Furthermore, RFA-SCA reduces the number of traces required for successful attacks by 14%–33% compared to the best baseline in the ASCAD Desync, ASCAD Gaussian Noise, and SAKURA-G scenarios. A systematic ablation study over seven loss-component variants across all six scenarios further supports the role and complementarity of the three components.

KEYWORDS: Side-channel analysis; deep learning; unsupervised domain adaptation; cross-device attack; multi-order moment alignment; conditional entropy; countermeasure

1 Introduction

Side-channel attacks extract keys by analyzing power consumption [1–4], electromagnetic radiation [5], or timing information [6,7] emitted during cryptographic device operation, compromising the security of deployed implementations. Although defense techniques such as masking [8], shuffling [9], and signal hiding [10] are widely deployed, the introduction of deep learning [11–13] enables attackers to learn complex leakage patterns and effectively attack even higher-order masked implementations. However, the effectiveness of profiling deep learning side-channel attacks relies on a critical assumption that training data (from the profiling device) and attack data (from the target device) follow the same distribution. When

the profiling and target devices differ, this assumption breaks down, leading to significant performance degradation [14].

Distribution shift in cross-device scenarios stems from both physical factors and defense mechanisms, affecting the feature distribution at multiple statistical orders [15,16]. Process-voltage-temperature (PVT) variations primarily shift the distribution centroid [17,18], while defense countermeasures such as clock jitter [19,20] and desynchronization [21] introduce temporal misalignment and trace-level perturbations that can induce covariance changes in the learned feature space [15]. This combination can limit single-order alignment methods under strong countermeasures.

Existing research on cross-device generalization follows two main approaches. The first aggregates training data from multiple devices to cover the target domain's variation space. X-DeepSCA [22] collects traces from multiple devices to train a universal model. Subsequent work extended this to electromagnetic channels [23] and heterogeneous hardware [24], exploring multi-device joint training [25,26] and data normalization [27]. However, such methods require access to multiple device copies [14], which is often impractical. The second approach uses unsupervised domain adaptation to reduce the distribution gap between source and target domains using only unlabeled target data, removing the need for the target-device key during the profiling phase. The CDPA method [28] first introduced maximum mean discrepancy (MMD) loss into side-channel attacks for kernel mean embedding alignment. Subsequently, PPPA [29] employs CORAL loss to constrain covariance matrix differences, and AL-PA [30] uses adversarial learning to train a domain discriminator. Transfer learning [31,32], meta-learning [33], and denoising preprocessing [34,35] have also been applied.

Despite these advances, two key limitations remain. First, existing methods apply only a single alignment constraint. Although MMD with a characteristic kernel can theoretically detect arbitrary distributional differences [36], under Gaussian kernels and finite samples the objective typically emphasizes lower-order weighted moment discrepancies more strongly than higher-order ones, so covariance and higher-order structure remain only implicitly constrained [37,38]. Similarly, CORAL aligns covariance matrices but does not directly correct mean shifts. When defense mechanisms induce temporal misalignment and trace-level perturbations, neither constraint alone is sufficient under limited sample budgets. Second, existing methods typically apply adaptation at a single layer, leaving the choice of alignment depth underexplored for hierarchical SCA representations.

To address these issues, we propose Robust Feature Alignment for Side-Channel Analysis (RFA-SCA), an unsupervised domain adaptation framework that jointly constrains distributional moments and decision boundaries to improve cross-domain feature alignment without requiring the target-device key during training. The novelty of this work lies in formulating, for cross-device and cross-countermeasure SCA, a task-specific integration that couples kernel-embedding alignment, covariance correction, and target-side decision-boundary refinement. To the best of our knowledge, existing cross-device SCA methods have not organized these three elements within a single unified framework, nor clarified their dependency structure under physically induced SCA domain shifts.

Our main contributions are as follows:

First, we propose RFA-SCA, an unsupervised domain adaptation framework that does not require the target-device key during training. The framework combines three objectives, each targeting a different aspect of domain shift: MMD loss aligns kernel mean embeddings, CORAL loss regularizes covariance-level feature discrepancies, and conditional entropy regularization refines decision boundaries on the target domain. This multi-order formulation provides joint distributional constraints for PVT and countermeasure-induced leakage shifts.

Second, we employ a dataset-adaptive multi-layer alignment strategy that applies domain-invariance constraints at multiple selected intermediate representations rather than solely at the final feature layer, reducing low-level temporal shifts together with device-specific and countermeasure-induced leakage biases through dataset-specific alignment points.

Third, we conduct experiments on four benchmarks (ASCAD, CHES CTF 2018, SAKURA-G, XMEGA) across six scenarios. Empirical evaluations show that RFA-SCA achieves competitive or superior performance across all six scenarios and is the only evaluated method to recover the key under the clock jitter countermeasure. It also lowers the attack complexity by reducing the required traces compared to baseline methods in the ASCAD Desync, ASCAD Gaussian Noise, and SAKURA-G scenarios.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the RFA-SCA framework; Section 4 presents experiments; Section 5 discusses results; Section 6 concludes.

2 Related Work

2.1 Deep Learning Side-Channel Analysis

Since Maghrebi et al. [11] first systematically applied MLPs and CNNs to side-channel attacks, deep learning has become a prominent tool in this domain. Early research primarily focused on algorithm comparison [39] and the introduction of data augmentation techniques [19]. Building upon these foundations, subsequent studies sought to formalize network construction, with Zaid et al. [12] establishing a systematic CNN design methodology and Wouters et al. [40] further refining architecture design principles. Deep learning has since been applied to break higher-order masking [41] and analyze large-scale polymorphic AES traces [42].

Despite this progress, deep learning-based side-channel analysis depends heavily on the independent and identically distributed (i.i.d.) assumption. When the profiling device and the target device differ, or when strong countermeasures like clock jitter are applied, this assumption is violated, leading to a severe degradation in attack performance. This limitation motivates cross-device generalization techniques.

2.2 Cross-Device SCA and Unsupervised Domain Adaptation

To address the portability challenge caused by distribution shifts [14,25], one major approach is multi-device data aggregation. For instance, X-DeepSCA [22] aggregates traces from multiple devices to train a universal model. However, a fundamental limitation of this paradigm is the practical difficulty of acquiring multiple physical device copies [14].

Alternatively, unsupervised domain adaptation (UDA) offers a practical alternative, as it aligns distributions using only unlabeled target data, removing the requirement for the target device's cryptographic key during training. Rooted in the domain adaptation theory established by Ben-David et al. [43], UDA has been successfully adapted for SCA. CDPA [28] pioneered this by introducing the Maximum Mean Discrepancy (MMD) loss, inspired by Long et al.'s Deep Adaptation Networks [44], to align the kernel mean embeddings of the source and target domains. To address different types of distortion, PPPA [29] employed the CORAL loss, originally proposed by Sun and Saenko [15], to explicitly constrain second-order covariance matrix differences. Meanwhile, AL-PA [30] utilized adversarial learning, akin to Ganin et al.'s DANN [45], to train a domain discriminator.

Beyond domain adaptation, other unsupervised paradigms have also been explored for SCA. For example, Kou et al. [46] proposed a multi-modal framework that combines power and electromagnetic signals via isometric compression and unsupervised clustering for key recovery without profiling. However, such methods operate in a single-device setting and do not address the cross-device distribution shift

problem. Within the UDA-based cross-device methods, existing approaches typically rely on a single alignment constraint (either mean or covariance), which is insufficient for complex distortions like clock jitter. Furthermore, they lack mechanisms to optimize the decision boundaries specifically for the target domain, a concept Grandvalet and Bengio [47] addressed via conditional entropy minimization.

Related combinations of discrepancy minimization and entropy regularization have been explored in the broader UDA literature. However, published cross-device SCA methods still mainly rely on a single adaptation mechanism. Our contribution is therefore not to claim a universally new UDA formulation, but to formulate and evaluate an SCA-oriented integration in which MMD, CORAL, and conditional entropy address complementary physically grounded failure modes.

Compared with generic computer-vision domain adaptation, cross-device SCA involves 1D temporal traces, smaller datasets, and structured shifts induced by PVT variations and temporal countermeasures. Recent contrastive- and transformer-based methods [48,49] have shown promise in broader UDA and in SCA model design, but we are not aware of a published cross-device SCA method that uses contrastive- or transformer-based domain adaptation as its core mechanism. We therefore use CDPA, PPPA, and AL-PA as representative published baselines for the current cross-device SCA setting, and leave such extensions to future work.

Table 1 summarizes existing cross-device SCA methods. RFA-SCA distinguishes itself by jointly optimizing MMD and CORAL while adding conditional entropy regularization, combining kernel-embedding and covariance-level alignment in an unsupervised adaptation setting.

Table 1: Comparison of cross-device side-channel attack methods.

Method	Core Technique	Alignment Target	Limitation
X-DeepSCA [22]	Multi-device aggregation	Data-level	Requires multiple devices
CDPA [28]	MMD loss	Mean embedding	No explicit covariance constraint
PPPA [29]	CORAL loss	Covariance	No independent mean discrepancy constraint
AL-PA [30]	Adversarial learning	Implicit	Requires discriminator
MTL-SCA [33]	Meta-transfer	Initialization	High compute cost

3 Methodology

3.1 Problem Definition

In a profiling attack, the attacker has labeled source data $\mathcal{D}_S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with $x_i^s \in \mathbb{R}^T$ (power traces) and $y_i^s \in \mathcal{Y}$ (labels: e.g., Hamming weight or identity). The attacker also has unlabeled target data $\mathcal{D}_T^{train} = \{x_j^t\}_{j=1}^{n_t}$ for adaptation (i.e., without requiring the target-device key) and $\mathcal{D}_T^{test}$ for key recovery. The source and target domains share the same label semantics $\mathcal{Y}$, but physical and countermeasure-induced effects alter the leakage distribution, so typically $P_S(x \mid y) \neq P_T(x \mid y)$. Let $P_S^{(f)} = G_f \# P_S$ and $P_T^{(f)} = G_f \# P_T$ denote the pushforward (induced) feature distributions via the feature extractor G_f. The goal is to learn $f_\theta : \mathbb{R}^T \to \mathcal{Y}$ by minimizing:

$$\min_\theta \underbrace{\mathbb{E}_{(x,y)\sim P_S}[\ell(f_\theta(x), y)]}_{\mathcal{L}_{CE}} + \gamma(e)\left(\alpha_{MMD}\,\mathcal{L}_{MMD} + \alpha_{CORAL}\,\mathcal{L}_{CORAL} + \alpha_{ENT}\,\mathcal{L}_{ENT}\right) \tag{1}$$

where $\mathcal{L}_{MMD}$ and $\mathcal{L}_{CORAL}$ measure the distributional discrepancy between the source and target feature distributions $P_S^{(f)}$ and $P_T^{(f)}$ (Sections 3.4.2 and 3.4.3), $\mathcal{L}_{ENT}$ minimizes the conditional entropy of target predictions (Section 3.4.4), and α_{MMD}, α_{CORAL}, α_{ENT} are the respective loss weights. The full multi-layer instantiation and warmup schedule are given in Eq. (9).

3.2 Framework Overview

The proposed RFA-SCA framework aims to learn domain-invariant and discriminative representations from side-channel traces. As illustrated in Fig. 1, the architecture consists of three main components.

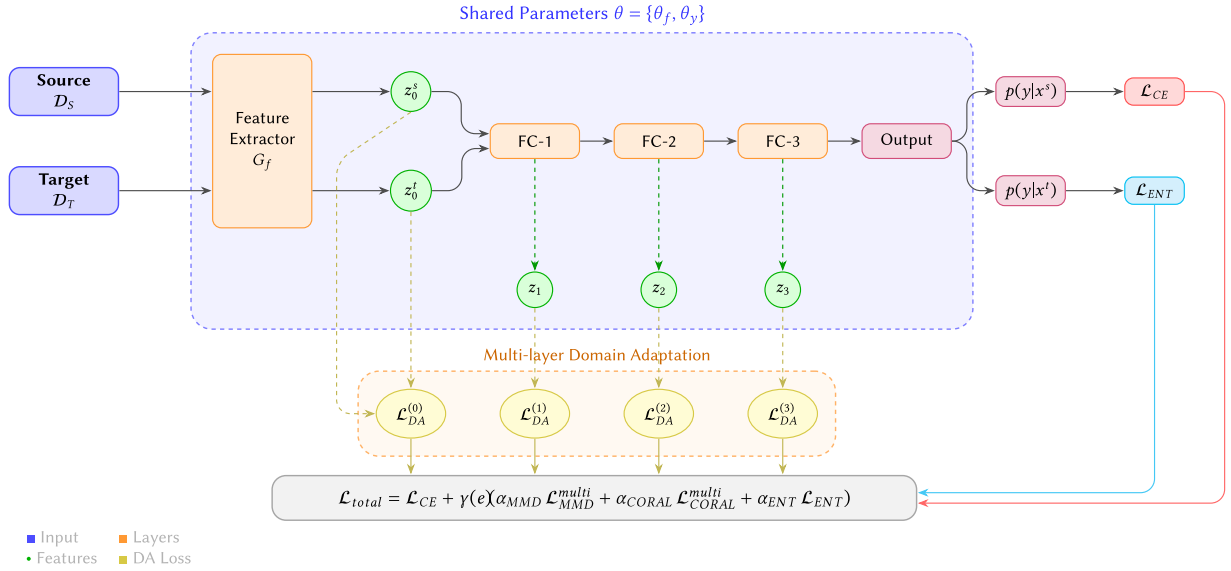


Figure 1: RFA-SCA framework. Source and target data pass through shared feature extractor and classifier. Domain adaptation constraints (MMD + CORAL) are applied at selected levels (up to z_0, z_1, z_2, z_3 depending on the dataset-specific architecture); conditional entropy regularizes target predictions.

The feature extractor $G_f(\cdot;\theta_f)$ is a 1D CNN that maps raw power traces $x \in \mathbb{R}^{1\times T}$ into a latent space $\mathcal{Z} \subset \mathbb{R}^d$, extracting local temporal patterns and hierarchical representations from the leakage signals. The classifier $G_y(\cdot;\theta_y)$, an MLP of fully connected layers, takes the extracted features $z = G_f(x)$ and outputs a probability distribution $\{p_k(x)\}_{k=1}^{|\mathcal{Y}|}$ over the sensitive intermediate values. A multi-layer domain adaptation module applies alignment constraints at selected network levels rather than solely at the final feature layer. The dataset-specific alignment choices are detailed in Section 3.4.5.

During the adaptation process, the source and target data are fed into the shared network. The classification loss ($\mathcal{L}_{CE}$) is computed using the labeled source data to maintain discriminative power. Simultaneously, the multi-order moment alignment losses (MMD and CORAL) are computed across the selected layers to bridge the domain gap, while the conditional entropy loss ($\mathcal{L}_{ENT}$) regularizes the target predictions. This joint optimization encourages features that are predictive of the cryptographic key while remaining invariant across devices and countermeasure conditions. This design is suited to cross-device SCA because leakage shifts affect both feature statistics and target-domain decision geometry under limited sample budgets. MMD and CORAL therefore address complementary distributional discrepancies, while conditional entropy handles residual boundary ambiguity after coarse alignment.

3.3 Geometric Intuition for Multi-Order Moment Alignment

Fig. 2 illustrates four geometric cases used as intuition for the alignment design: PVT-related translation, countermeasure-induced perturbation, residual mismatch after MMD-only alignment, and the reduced mismatch after adding CORAL.

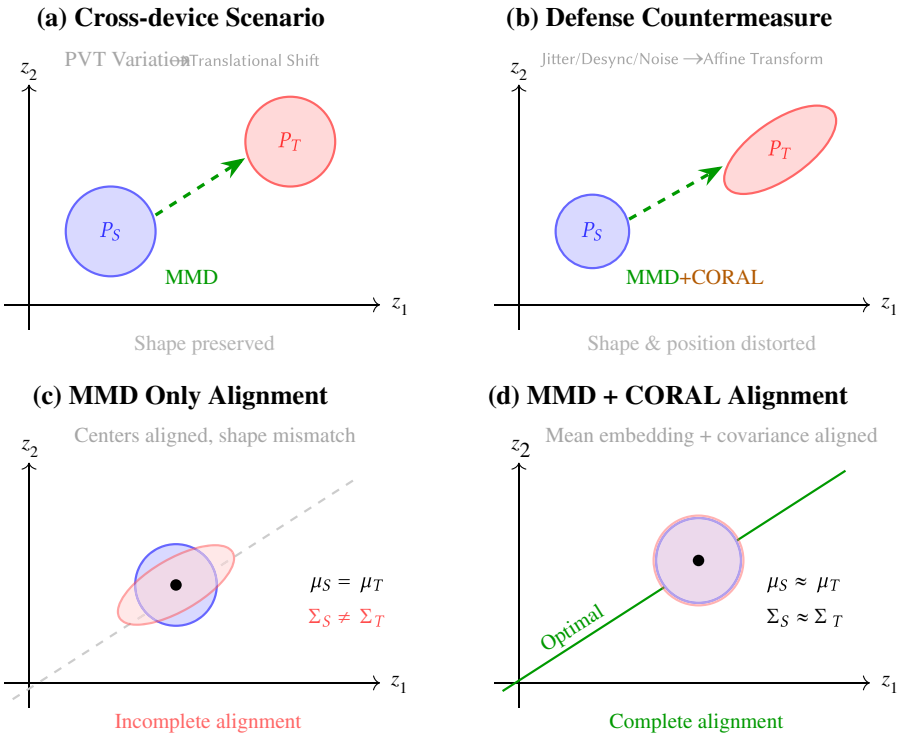


Figure 2: Geometric properties of domain shift types. (**a**) Cross-device: PVT causes translational shift. (**b**) Countermeasures: clock jitter and desynchronization introduce temporal misalignment and trace-level perturbations. (**c**) MMD-only: reduces low-order discrepancy but may leave shape mismatch. (**d**) MMD + CORAL: reduced feature-level discrepancy.

3.4 Loss Functions

Each loss component targets a distinct failure mode in cross-device and cross-implementation side-channel analysis: $\mathcal{L}_{CE}$ provides discriminative supervision from labeled source-domain samples, $\mathcal{L}_{MMD}$ reduces the discrepancy between source and target feature distributions in RKHS, $\mathcal{L}_{CORAL}$ aligns their second-order statistics, and $\mathcal{L}_{ENT}$ encourages confident predictions on unlabeled target samples.

3.4.1 Classification Loss

The classification loss trains the shared feature extractor to learn class-discriminative representations from the labeled source domain, forming the basis on which the domain adaptation objectives below rely.

Cross-entropy on source domain:

$$\mathcal{L}_{CE} = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{k=1}^{|\mathcal{Y}|}\mathbb{I}(y_i^s = k)\log p_k(x_i^s) \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $|\mathcal{Y}|$ is the number of classes, and $p_k(x)$ denotes the predicted probability for class k.

3.4.2 MMD Loss: Kernel Mean Embedding Alignment

MMD measures the discrepancy between source and target distributions via kernel mean embeddings in an RKHS [50]. Let $\mu_P = \mathbb{E}_{z\sim P}[\phi(z)]$ and $\mu_Q = \mathbb{E}_{z\sim Q}[\phi(z)]$; then $\mathrm{MMD}(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}$, and with a characteristic kernel such as Gaussian RBF, $\mathrm{MMD}(P, Q) = 0$ implies $P = Q$. In practice, we optimize the biased empirical estimate of MMD^2:

$$\mathcal{L}_{MMD} = \frac{1}{n_s^2}\sum_{i,j} k(z_i^s, z_j^s) + \frac{1}{n_t^2}\sum_{i,j} k(z_i^t, z_j^t) - \frac{2}{n_s n_t}\sum_{i,j} k(z_i^s, z_j^t) \tag{3}$$

We use a multi-scale Gaussian RBF kernel with five bandwidths centered at an adaptive base bandwidth b_0 estimated from the batch pairwise distances. Specifically, let $k_b(z, z') = \exp(-\|z - z'\|^2/b)$ denote a single-bandwidth Gaussian kernel. For clarity, we first discuss this single-bandwidth case, for which MMD^2 admits an equivalent weighted tensor-moment representation; the multi-scale case follows by summing the corresponding terms over bandwidths.

$$\mathrm{MMD}^2_{k_b}(P, Q) = \sum_{m=0}^{\infty} \|\mathbb{E}_{z\sim P}[\varphi_m(z)] - \mathbb{E}_{z\sim Q}[\varphi_m(z)]\|^2, \qquad \varphi_m(z) = \sqrt{\frac{2^m}{m!\, b^m}} \exp\left(-\frac{\|z\|^2}{b}\right) z^{\otimes m}. \tag{4}$$

Eq. (4) shows that Gaussian-kernel MMD does not reduce to ordinary input-space mean matching; rather, it matches an infinite sequence of weighted tensor moments in which the factorial normalization $\sqrt{2^m/(m!\, b^m)}$ suppresses contributions of sufficiently high order. Moreover, under bounded features $\|z\| \le R$, for any truncation order M the residual contribution of terms above order M admits a factorial tail bound:

$$\sum_{m=M+1}^{\infty} \|\mathbb{E}_P[\varphi_m(z)] - \mathbb{E}_Q[\varphi_m(z)]\|^2 \le 4 \sum_{m=M+1}^{\infty} \frac{(2R^2/b)^m}{m!}, \tag{5}$$

Although MMD with a characteristic kernel is theoretically sensitive to all moment orders, its finite-sample behavior concentrates the optimization signal on lower-order components. Specifically, from the norm bound $\|\varphi_m(z)\|^2 \le (2R^2/b)^m/m!$ and the standard sample-mean variance inequality, each per-component empirical mean $\frac{1}{n}\sum_{i=1}^n \varphi_m(z_i)$ has variance at most $\mathcal{O}\big((2R^2/b)^m/(n\, m!)\big)$ for bounded features. Related finite-sample rate analyses for MMD estimation can be found in [51]. We emphasize that φ_m is a Gaussian-weighted tensor moment rather than an ordinary moment such as the input-space mean $\mathbb{E}[z]$ or covariance $\mathbb{E}[zz^\top]$. The two coincide up to a global scale only in the near-isotropic regime $\|z\| \ll \sqrt{b}$. Under practical batch sizes, the MMD objective therefore predominantly aligns lower-order weighted components, while higher-order structure receives only weak statistical constraints. We note that this analysis is an analytical characterization under a bounded-feature assumption rather than a finite-sample theorem for deep-network features. Nevertheless, it provides a practical rationale for combining MMD with CORAL. MMD provides implicit distribution-level alignment through the Gaussian-weighted expansion above, whereas CORAL explicitly regularizes the input-space second-order statistics that remain sensitive to cross-device and countermeasure-induced distribution shift.

3.4.3 CORAL Loss: Second-Order Alignment

The CORAL loss complements MMD by explicitly aligning second-order statistics, reducing covariance-level distortions between source and target features. This constraint matters most when mean-embedding alignment alone cannot capture the structural mismatch between domains.

The CORAL loss [15] aligns covariance matrices:

$$\mathcal{L}_{CORAL} = \frac{1}{4d^2}\|C_S - C_T\|_F^2 \tag{6}$$

where $C_S = \frac{1}{n_s-1}\tilde{Z}_S^\top \tilde{Z}_S$ and $C_T = \frac{1}{n_t-1}\tilde{Z}_T^\top \tilde{Z}_T$ are the sample covariance matrices of mean-centered features $\tilde{Z}_S$ and $\tilde{Z}_T$, respectively.

3.4.4 Conditional Entropy Loss

Even after feature alignment, the decision boundary may still pass through high-density regions of the target domain, resulting in uncertain predictions. The conditional entropy loss mitigates this problem by encouraging confident predictions on unlabeled target samples, thereby refining the boundary once sufficient cross-domain overlap has been established. Based on [47]:

$$\mathcal{L}_{ENT} = -\frac{1}{n_t}\sum_{j=1}^{n_t}\sum_{k=1}^{|\mathcal{Y}|} p_k(x_j^t)\log p_k(x_j^t) \tag{7}$$

where n_t denotes the number of target-domain samples, $|\mathcal{Y}|$ is the number of classes, and $p_k(x_j^t)$ denotes the predicted probability that target sample x_j^t belongs to class k.

This regularization is effective only after MMD and CORAL have established sufficient cross-domain feature overlap. In RFA-SCA, the three losses therefore act sequentially and complementarily: MMD reduces the discrepancy between source and target feature distributions in RKHS, CORAL further aligns second-order statistics, and conditional entropy then sharpens the target-domain decision boundary in the aligned feature space. If the two domains remain poorly aligned, entropy minimization can instead be harmful, because the model may assign high-confidence predictions to target samples that have not yet been mapped to the correct class-conditional regions, thereby collapsing the decision boundary onto incorrect classes.

3.4.5 Multi-Layer Adaptation

Rather than applying domain adaptation constraints only at the final representation layer, we impose alignment losses at multiple intermediate layers so that both shallow and deep domain discrepancies can be reduced during training. This design allows the model to jointly reduce low-level temporal variation and higher-level class-discriminative leakage mismatch across domains.

The choice of which layers to align is guided by the feature transferability principle established by Yosinski et al. [52]. In deep networks, hidden representations evolve from relatively generic in shallow layers to increasingly task-specific in deeper layers, and their transferability decreases markedly in the higher layers under domain shift. For CNN-based side-channel analysis, this progression admits a natural physical interpretation. Early convolutional blocks primarily act as local leakage-pattern extractors, capturing temporal shapes and short-range dependencies in a comparatively device-invariant manner, whereas the terminal feature extractor output and the subsequent fully connected classifier layers recombine these local patterns into global discriminative representations that are more susceptible to device-specific and countermeasure-induced bias. This view is consistent with the cross-device SCA analysis of Cao et al. [28], whose discussion

and ablation of adaptation-layer placement indicate that cross-device degradation is driven mainly by device-specific features learned in the deeper decision-making layers and that multi-layer classifier-side adaptation provides a favorable trade-off. Following the Deep Adaptation Network (DAN) paradigm of Long et al. [44], we therefore prioritize alignment on multiple late-stage representations, specifically the classifier hidden layers and, when dimensionality permits, the terminal feature-extractor output z_0, rather than on early convolutional feature maps. These theoretical considerations identify which layers should be prioritized, while the exact subset of alignment points remains architecture- and dataset-dependent rather than theoretically unique.

Rather than applying domain adaptation constraints solely at the final feature layer, we compute MMD and CORAL losses at $L+1$ selected intermediate representations $\{z_l\}_{l=0}^{L}$, where z_0 typically denotes the feature extractor output and subsequent z_l correspond to classifier hidden layers:

$$\mathcal{L}_{MMD}^{multi} = \sum_{l=0}^{L} \mathcal{L}_{MMD}^{(l)}, \quad \mathcal{L}_{CORAL}^{multi} = \sum_{l=0}^{L} \mathcal{L}_{CORAL}^{(l)} \tag{8}$$

where each layer contributes with equal weight. The number and selection of alignment layers are determined by the dataset-specific architecture: for ASCAD variants (3-block CNN with three FC hidden layers), alignment is applied at three points (z_0, z_1, z_2); for XMEGA and SAKURA-G, at two points (z_0, z_1); for CHES CTF, at two classifier hidden layers (z_1, z_2) rather than at the high-dimensional feature extractor output (d = 70,112), as direct MMD/CORAL computation on such high-dimensional representations is computationally prohibitive and tends to produce noisy gradient estimates. This exclusion is also consistent with the transferability principle: in this architecture, the feature extractor output z_0 remains closer to the generic representation space and is therefore less likely to concentrate the device-specific bias that dominates the deeper classifier layers. This hierarchical design captures both low-level temporal shifts at shallow layers and higher-level device-specific leakage bias at deeper layers.

3.5 Total Loss

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \gamma(e)\left(\alpha_{MMD}\mathcal{L}_{MMD}^{multi} + \alpha_{CORAL}\mathcal{L}_{CORAL}^{multi} + \alpha_{ENT}\mathcal{L}_{ENT}\right) \tag{9}$$

where $\gamma(e) = \min(1, e/E_{warmup})$ is a dynamic warmup factor at epoch e, and α_{MMD}, α_{CORAL}, and α_{ENT} are the specific weights for each domain adaptation loss component.

Intuitively, $\gamma(e)$ prevents the adaptation losses from overwhelming source-domain discriminative learning in the early stages of training. The model first learns class-discriminative structure from labeled source samples, and then gradually strengthens cross-domain alignment and target-domain regularization as training proceeds. The linear warmup reinforces the loss ordering discussed in Section 3.4.4 by delaying strong adaptation until the source-domain discriminative structure has stabilized.

3.6 Training Strategy

The training process adopts a two-phase strategy, as summarized in Algorithm 1.

Phase 1: Source Pretraining. The model is first trained on the labeled source domain data $\mathcal{D}_S$ using only the classification loss $\mathcal{L}_{CE}$. This enables the model to learn a discriminative mapping from power traces to sensitive intermediate values. We employ the Adam optimizer with a fixed learning rate.

Phase 2: Domain Adaptation Fine-tuning. After loading the pretrained model parameters, the Batch Normalization layers are switched to evaluation mode, freezing their running mean and variance statistics. This prevents the target domain data from contaminating the normalization parameters learned from the

source domain. During fine-tuning, the classification loss and domain adaptation losses are jointly optimized using the Adam optimizer with a cosine annealing learning rate scheduler. Gradient clipping (max_norm = 5.0) is applied to prevent gradient explosion. Furthermore, a linear warmup strategy is adopted for the domain adaptation losses. The global warmup factor is increased from 0 to 1 over the initial E_{warmup} epochs, preventing large initial adaptation gradients from disrupting the discriminative structure learned by the pretrained classifier.

Algorithm 1: RFA-SCA Training Algorithm

Input: Source $\mathcal{D}_S$, target $\mathcal{D}_T$
Output: Trained parameters θ
// Phase 1: Source pretraining
for epoch $e = 1$ to E_{pre} **do**
  for each batch $(x^s, y^s) \sim \mathcal{D}_S$ **do**
    Forward: $\hat{y}^s = G_y(G_f(x^s))$; compute $\mathcal{L}_{CE}$
    Backward; update parameters
  end for
end for
// Phase 2: Domain adaptation
Load pretrained model; freeze BatchNorm
for epoch $e = 1$ to E_{fine} **do**
  Compute warmup factor $\gamma(e) = \min(1, e/E_{warmup})$
  for each batch **do**
    Get (x^s, y^s) and x^t; extract features $\{z_l^s\}$, $\{z_l^t\}$
    Compute $\mathcal{L}_{CE}$, $\mathcal{L}_{MMD}^{multi}$, $\mathcal{L}_{CORAL}^{multi}$, $\mathcal{L}_{ENT}$
    Compute $\mathcal{L}_{total}$ using Eq. (9)
    Backward; gradient clip (max_norm = 5.0); update
  end for
  Step CosineAnnealingLR scheduler
end for
return θ

4 Experiments

4.1 Experimental Setup

All experiments were conducted on a workstation equipped with dual NVIDIA GeForce RTX 4090 GPUs (24 GB VRAM each), using Python 3.12, PyTorch 2.7, and CUDA 12.8. The global random seed was fixed to 8 (covering Python, NumPy, and PyTorch CPU/CUDA random states) to ensure identical starting conditions and data ordering across all runs.

4.1.1 Datasets

Table 2 summarizes six scenarios: three cross-device (XMEGA, SAKURA-G, CHES CTF 2018) [28] and three countermeasure (ASCAD [21] Desync, Clock Jitter, Gaussian Noise).

Table 2: Experimental dataset configuration.

Dataset/Scenario	Trace Length	Train/Val/Test	Label Type
XMEGA (D1→D2)	500	20k/5k/4.5k	HW (9 classes)
SAKURA-G (D1→D2)	1000	85k/5k/9.4k	Identity (256)
CHES CTF 2018 (C→D)	2200	43k/1k/1k	HW (9 classes)
ASCAD Desync (0→100)	700	45k/5k/9k	Identity (256)
ASCAD Clock Jitter	700	45k/5k/9k	Identity (256)
ASCAD Gaussian Noise (σ = 8)	700	45k/5k/9k	Identity (256)

4.1.2 Baselines and Metrics

Baselines: CDPA [28], PPPA [29], AL-PA [30]. Metrics: Guessing Entropy (GE), Success Rate (SR), Measurements to Disclosure (MTD). To mitigate variance from random trace ordering, all GE, SR, and MTD values are computed via Monte Carlo averaging over 100 independent attack runs, each using a random permutation of the available test traces. At each run, cumulative log-probability scores are accumulated trace by trace, the rank of the correct key is recorded at every step, and the final GE curve is the element-wise mean of 100 rank trajectories; SR at each step is the fraction of runs in which the correct key ranks first.

These three metrics capture distinct aspects of attack performance but differ in discriminative power for domain adaptation evaluation. GE@Max (guessing entropy evaluated at the maximum available traces) quantifies the final rank of the correct key; however, it saturates at zero once sufficiently many traces are available, making it unable to distinguish methods that all achieve successful key recovery. SR@Max (success rate at the maximum traces) similarly saturates at 100%. In contrast, MTD defined as the minimum number of traces at which GE first reaches zero, captures the convergence speed of the attack. Better-aligned representations enable the model to accumulate correct key evidence from fewer traces. MTD also implicitly encodes attack success or failure, since MTD exceeding the trace limit indicates that the method was unable to recover the key. We therefore adopt MTD as the primary comparison metric throughout Sections 4 and 4.3, while reporting GE@Max and SR@Max in the corresponding results tables for completeness.

4.1.3 Implementation Details and Fairness Measures

To ensure a fair comparison, all baseline methods (CDPA, PPPA, AL-PA) were re-implemented from scratch within a unified codebase, rather than citing results from original publications. The following measures were taken to isolate the effect of domain adaptation strategies from confounding factors:

For each dataset, CDPA, PPPA, and RFA-SCA employ identical CNN feature extractors and classifier layers. The only difference lies in the domain adaptation loss applied during fine-tuning. Table 3 summarizes the per-dataset architecture. AL-PA shares the same base CNN feature extractor, but adds a bottleneck layer for ASCAD (256→32) and SAKURA-G (448→32) as required by its adversarial discriminator design [30]. For XMEGA and CHES CTF, all four methods use the same architectures.

Preprocessing, data splits, and initialization are also shared across methods. All methods use the same preprocessing pipeline (horizontal standardization for XMEGA and SAKURA-G; no preprocessing for others), identical data splits (Table 2), and the global random seed described in Section 4. CDPA, PPPA, and RFA-SCA share the same source-domain pretrained checkpoint, ensuring that performance differences originate solely from the domain adaptation strategy. AL-PA requires independent pretraining due to its bottleneck layer but uses the same source data and seed.

Table 3: Per-dataset CNN architecture. Conv blocks: number of convolutional blocks with channel progression; FC layers: fully connected layer dimensions.

Dataset	Conv Blocks (Channels)	FC Layers
XMEGA	3 blocks (16→32→64)	64→20
SAKURA-G	3 blocks (8→16→32)	448→2
CHES CTF	1 block (32, kernel 10)	70, 112→400→400
ASCAD	3 blocks (32→64→128)	256→20→20→20

Regarding hyperparameter tuning, for each baseline, hyperparameters (learning rate, loss weight, fine-tuning epochs) were individually tuned per dataset to achieve their best performance, following the guidelines in their respective original papers. This ensures that reported differences reflect the inherent capability of each domain adaptation strategy rather than suboptimal tuning.

Table 4 lists the RFA-SCA hyperparameter configuration for each dataset. The four tuning targets α_{MMD}, α_{CORAL}, α_{ENT}, and E_{warmup} were determined by sequential grid search with local refinement, using minimum validation-set guessing entropy at the maximum available traces as the selection criterion. The candidate grids and refinement rules are reported in Appendix A (Table A1).

Table 4: RFA-SCA hyperparameter configuration per dataset. α_{MMD}, α_{CORAL}, α_{ENT}: loss weights; E_{warmup}: warmup epochs; E_{fine}: fine-tuning epochs; η_{fine}: fine-tuning learning rate; Layers: alignment points.

Dataset	α_{MMD}	α_{CORAL}	α_{ENT}	E_{warmup}	E_{fine}	η_{fine}	Alignment Layers
XMEGA	1.0	10.0	0.1	20	60	5e−4	z_0, z_1
SAKURA-G	0.05	10.0	0.001	5	25	1e−3	z_0, z_1
CHES CTF 2018	0.1	2.0	0.05	0	10	2e−4	z_1, z_2
ASCAD Desync	0.8	0.5	0.01	10	80	1e−3	z_0, z_1, z_2
ASCAD Clock Jitter	0.33	0.033	0.005	3	80	3e−4	z_0, z_1, z_2
ASCAD Gaussian Noise	1.0	0.5	0.02	20	120	2e−4	z_0, z_1, z_2

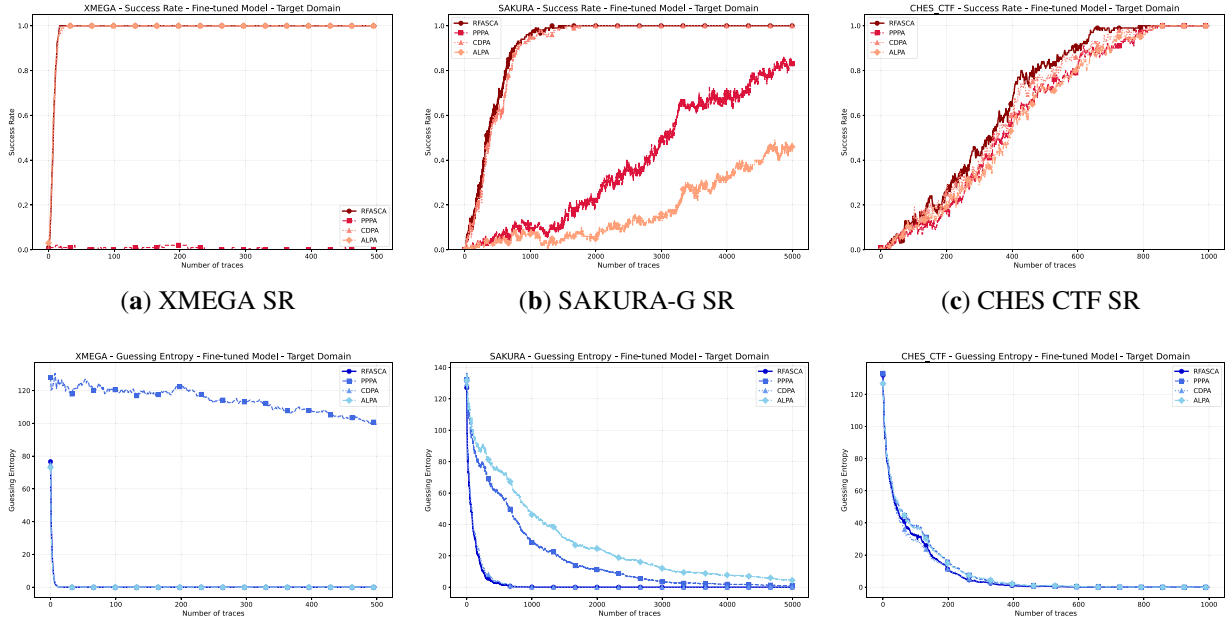
The final configurations exhibit interpretable regularities tied to dataset properties. The Gaussian RBF kernel operates on pairwise squared distances whose scale grows with alignment-layer width d. By contrast, the CORAL loss is already normalized by $1/(4d^2)$. To keep adaptation gradients balanced, α_{MMD} therefore generally decreases as d increases, e.g., from 1.0 at $d = 64$ (XMEGA) to 0.05 at $d = 448$ (SAKURA-G). The ASCAD Gaussian Noise scenario ($d = 256$, $\alpha_{MMD} = 1.0$) is a notable exception. Additive noise creates stochastic class overlap rather than systematic geometric distortion, requiring a stronger MMD signal to establish sufficient distributional overlap before entropy regularization can take effect. The ratio $\alpha_{CORAL}/\alpha_{MMD}$ reflects the dominant shift type. Cross-device PVT variations induce systematic covariance-level discrepancies that favor CORAL with ratios of 10–200×, whereas stochastic countermeasure perturbations shift the balance toward MMD with ratios of 0.1–0.6. Conditional entropy is also affected by class cardinality. Because $H_{\max} = \log|\mathcal{Y}|$ equals approximately 5.55 for 256-class identity labels and 2.20 for 9-class Hamming-weight labels, α_{ENT} for identity-label datasets (0.001–0.02) is roughly one order of magnitude smaller than for HW-label datasets (0.05–0.1). This smaller weight prevents per-sample entropy gradients from dominating the classification objective.

4.2 Method Comparison

Following Section 4.1.2, we use MTD as the primary metric for comparing convergence speed and attack success.

4.2.1 Cross-Device Results

To assess the robustness of the proposed method against physical PVT variations, we evaluate its performance across three cross-device scenarios (Fig. 3 and Table 5). RFA-SCA achieves the best or comparable MTD across all three datasets. On SAKURA-G dataset, it reduces the required traces by 14% compared to the best baseline (CDPA), whereas AL-PA and PPPA fail to reach 100% SR within the trace limit. On CHES CTF, RFA-SCA achieves a modest improvement over all baselines, and on XMEGA it matches CDPA's already-fast convergence.



(**a**) XMEGA SR (**b**) SAKURA-G SR (**c**) CHES CTF SR

(**d**) XMEGA GE (**e**) SAKURA-G GE (**f**) CHES CTF GE

Figure 3: Success rate (top) and guessing entropy (bottom) for cross-device scenarios.

Table 5: Cross-device key recovery results.

Dataset	Method	GE@Max	SR@Max	MTD
XMEGA	CDPA	0.00	100.0%	18
	AL-PA	0.00	100.0%	25
	PPPA	99.08	0.0%	>100
	RFA-SCA	0.00	100.0%	18
CHES CTF 2018	CDPA	0.00	100.0%	819
	AL-PA	0.00	100.0%	822
	PPPA	0.00	100.0%	828
	RFA-SCA	0.00	100.0%	786

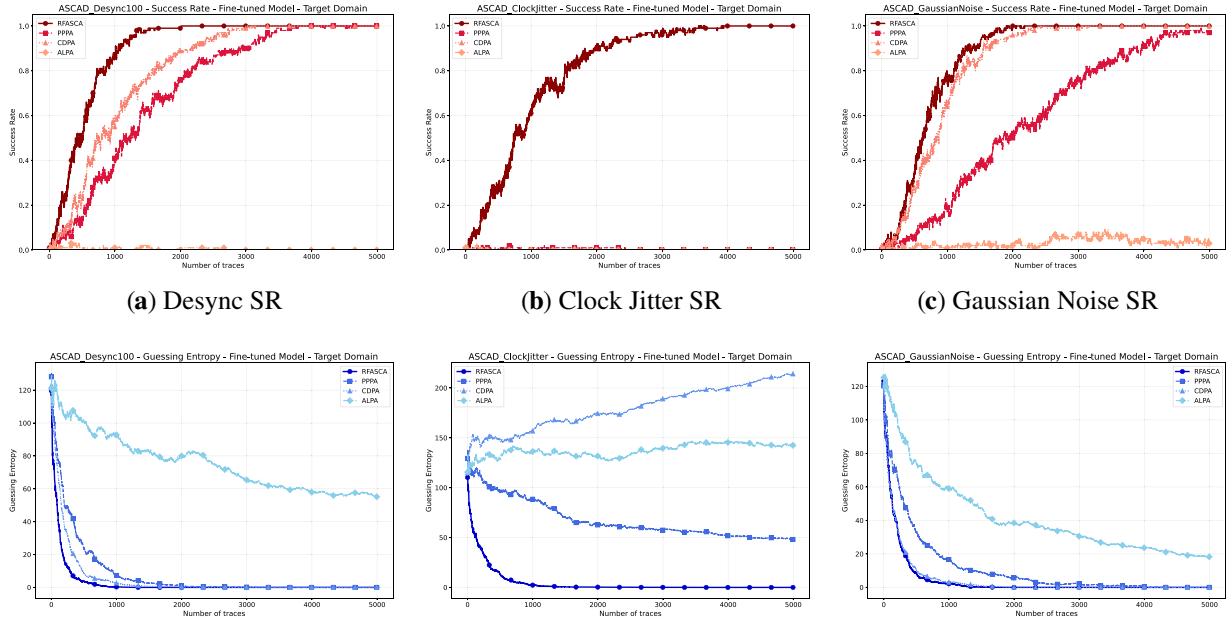
(Continued)

Table 5 (continued)

Dataset	Method	GE@Max	SR@Max	MTD
SAKURA-G	CDPA	0.00	100.0%	1530
	AL-PA	4.33	45.0%	>3000
	PPPA	0.85	82.0%	>3000
	RFA-SCA	0.00	100.0%	1316

4.2.2 Countermeasure Results

Beyond physical variations, we evaluate the methods against three countermeasure scenarios (Fig. 4 and Table 6). RFA-SCA reduces the required traces by 33% in the ASCAD Desync scenario and by 20% in the ASCAD Gaussian Noise scenario compared to the best baseline (CDPA). RFA-SCA is the only method among the four evaluated approaches (CDPA, PPPA, AL-PA, RFA-SCA) achieving successful key recovery under the severe clock jitter countermeasure. All three baselines fail to reach GE = 0 within the trace limit. The ablation study in Section 4.3 further examines the role of MMD-containing variants in this scenario and the seed sensitivity observed for Gaussian Noise.



(**a**) Desync SR (**b**) Clock Jitter SR (**c**) Gaussian Noise SR

(**d**) Desync GE (**e**) Clock Jitter GE (**f**) Gaussian Noise GE

Figure 4: Countermeasure scenario results. Among the four compared methods, RFA-SCA is the only one achieving key recovery under clock jitter.

Table 6: Countermeasure scenario key recovery results.

Scenario	Method	GE@Max	SR@Max	MTD
ASCAD Desync	CDPA	0.00	100.0%	3020
	AL-PA	55.15	0.0%	>5000

(Continued)

Table 6 (continued)

Scenario	Method	GE@Max	SR@Max	MTD
	PPPA	0.00	100.0%	3761
	RFA-SCA	0.00	100.0%	2017
ASCAD Clock Jitter	CDPA	214.38	0.0%	>5000
	AL-PA	143.28	0.0%	>5000
	PPPA	48.13	0.0%	>5000
	RFA-SCA	0.00	100.0%	3557
ASCAD Gaussian Noise	CDPA	0.00	100.0%	2381
	AL-PA	18.27	3.0%	>5000
	PPPA	0.03	97.0%	>5000
	RFA-SCA	0.00	100.0%	1901

4.3 Ablation Study

To quantify the individual and joint contributions of each loss component, we conduct a systematic ablation study across all six scenarios. Seven variants are evaluated: (1) Baseline (classifier only, no domain adaptation), (2) MMD only, (3) CORAL only, (4) MMD+CORAL, (5) MMD+Ent, (6) CORAL+Ent, and (7) Full (MMD+CORAL+Ent, i.e., the RFA-SCA formulation). All variants share the identical CNN backbone and training pipeline described in Section 4; the only difference is which domain adaptation loss terms are activated. To ensure statistical reliability, each variant is trained with five independent random seeds (0, 42, 123, 2024, 8). We report mean ± std MTD over five seeds in Table 7 and count successful scenarios using the same trace-limit criterion defined in Section 4.1.2.

Table 7: Ablation study: MTD (mean ± std over 5 seeds). Best MTD per scenario is **bolded**. Entries of the form > L indicate that MTD exceeds the corresponding trace limit L (attack failed). Underlined values highlight that Full (RFA-SCA) achieves successful key recovery (MTD < trace limit) in all six scenarios—the only variant to do so. Trace limits: 100 (XMEGA), 1000 (CHES CTF), 3000 (SAKURA-G), 5000 (others). The rightmost column tallies the number of successful scenarios per variant.

Variant	Cross-Device			Countermeasure			#Pass
	XMEGA	CHES CTF	SAKURA-G	Desync	Clock Jitter	Gaussian Noise	
Baseline	>100	842.8 ± 95.2	>3000	>5000	>5000	>5000	1/6
MMD	24.0 ± 2.3	759.6 ± 95.9	1351.6 ± 228.5	**1504. 6 ± 144.9**	3710.4 ± 752.1	>5000	5/6
CORAL	>100	871.6 ± 93.6	>3000	3112.6 ± 422.9	>5000	>5000	2/6
MMD+CORAL	**22.8 ± 0.7**	772.0 ± 86.7	1342.4 ± 173.3	1520.0 ± 131.1	**3135.8 ± 225.2**	>5000	5/6
MMD+Ent	24.4 ± 2.4	**738.4 ± 65.5**	**1295.6 ± 171.6**	1594.6 ± 122.1	3200.8 ± 205.3	>5000	5/6
CORAL+Ent	>100	>1000	>3000	>5000	>5000	>5000	0/6
Full (RFA-SCA)	23.2 ± 0.7	766.4 ± 99.5	1339.4 ± 169.8	1539.4 ± 87.4	3250.2 ± 342.3	**4430.0 ± 1140.0**	**6/6**

A note on the experimental protocol is warranted. The main comparison results in Tables 5 and 6 use a single seed (seed = 8) to ensure fair comparison with baseline methods under identical conditions, whereas this ablation study employs five independent seeds to assess stability and variance. Consequently, numerical

differences exist between the two tables and vary in magnitude across scenarios. For most scenarios the discrepancy is modest (e.g., Clock Jitter MTD: 3557 in Table 6 vs. 3250.2 ± 342.3 in Table 7, a ~9% difference). However, for Gaussian Noise the gap is substantially larger (MTD: 1901 vs. 4430 ± 1140), indicating that seed = 8 yields a particularly favorable outcome in this scenario while other seeds approach or reach the trace limit. This high seed sensitivity is analyzed in Section 5.

The MTD results reveal a clear hierarchy among the losses. MMD is the primary driver: MMD alone succeeds in 5 of 6 scenarios, whereas CORAL alone succeeds in only 2. Adding CORAL mainly benefits covariance-sensitive settings, lowering Clock Jitter MTD from 3710 to 3136 and reducing the reported cross-seed standard deviation from 752 to 225. Conditional entropy is most visible in the Gaussian Noise regime, where the Full variant achieves MTD < 5000 (4430 ± 1140) while every other variant reaches the trace limit. Conversely, CORAL without MMD causes negative transfer, and CORAL+Ent fails in all six scenarios. These observations support the role-structured design described in Section 3.4.4, without requiring each component to be individually optimal in every scenario.

The #Pass column summarizes scenario-level coverage. Full (RFA-SCA) reaches 6/6, while the next-best variants (MMD, MMD+CORAL, MMD+Ent) each reach 5/6. Its per-scenario MTD is consistently within 1%–4% of the best single-variant MTD, suggesting that the three loss components are mutually reinforcing rather than conflicting. This coverage offers a practical advantage when the nature and severity of the domain shift are unknown a priori, although the Gaussian Noise scenario remains seed-sensitive as analyzed in Section 5.

5 Discussion

5.1 Feature Distribution Visualization

The t-SNE visualizations presented in Figs. 5 and 6 provide qualitative support for multi-order moment alignment. Before adaptation, source domain (circle) and target domain (triangle) features are spatially separated; after RFA-SCA, they overlap and form compact mixed clusters per class while maintaining inter-class separation.

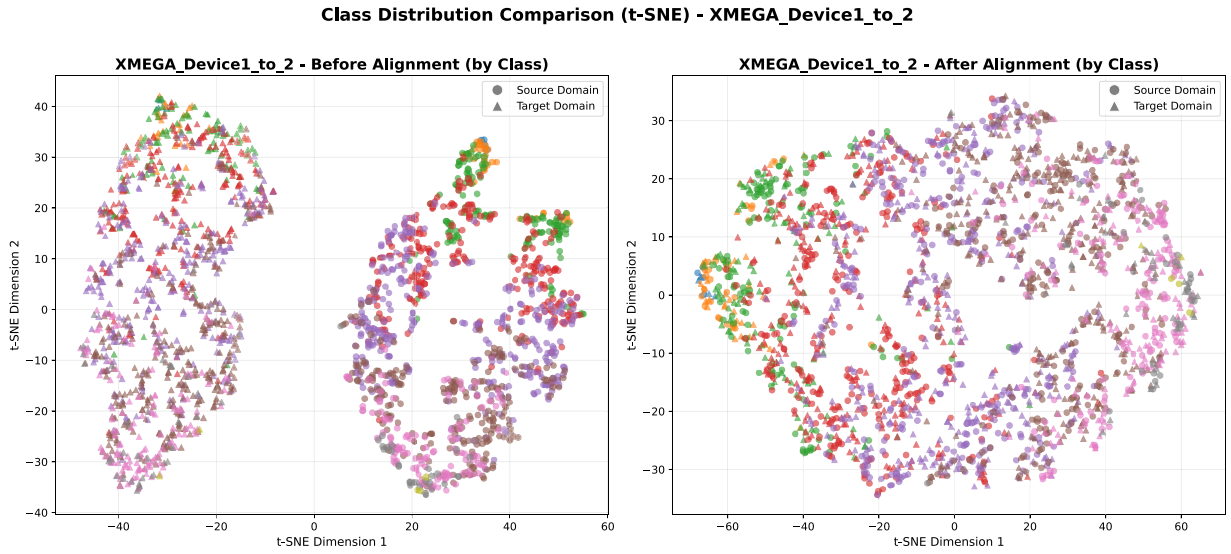
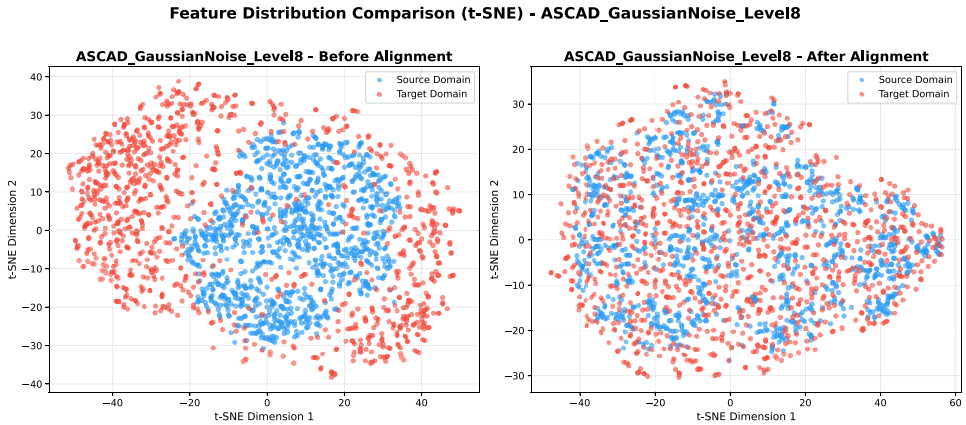
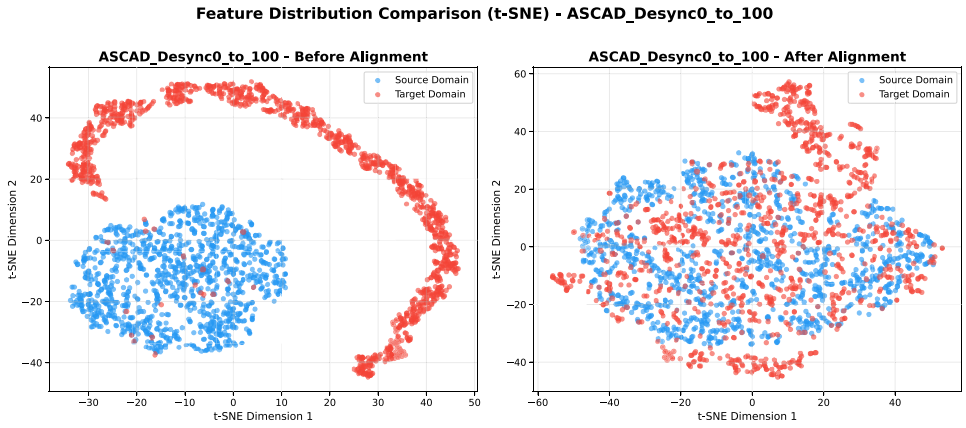


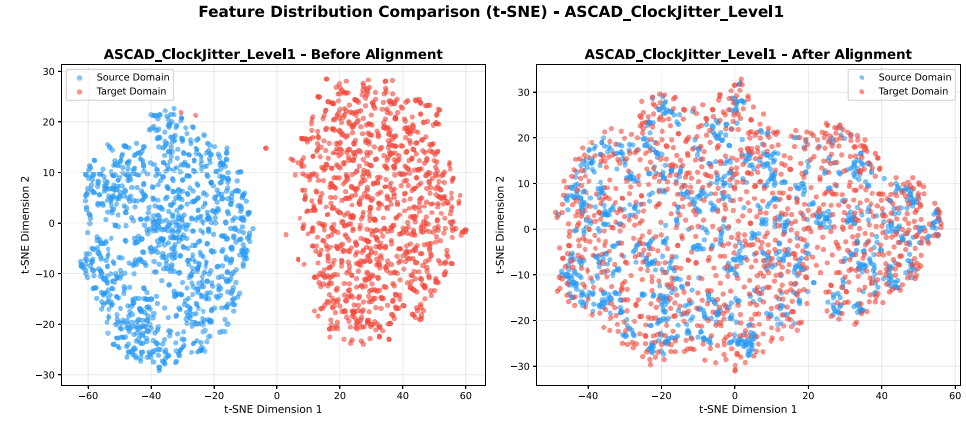
Figure 5: XMEGA cross-device t-SNE visualization. Left: before alignment; right: after RFA-SCA.



(a) Gaussian Noise



(b) Desync



(c) Clock Jitter

Figure 6: ASCAD t-SNE visualization for three countermeasure scenarios.

5.2 Domain Gap Analysis

The results reveal varying domain-gap characteristics across scenarios. In cross-device scenarios (XMEGA, CHES CTF), the domain gap is primarily driven by PVT variations, which manifest as translational shifts in the feature space. Notably, the CHES CTF dataset exhibits a particularly small domain gap.

Even the no-adaptation Baseline achieves MTD = 842.8 (Table 7), suggesting that the Challenge-to-Device distribution shift in this benchmark is inherently modest, likely because both devices share similar hardware implementations with limited manufacturing variation. We nonetheless include CHES CTF in our evaluation to check whether RFA-SCA degrades performance when the domain gap is small; it instead achieves a modest MTD improvement (766 vs. 843). Countermeasure scenarios show a different pattern. The failure of CDPA and PPPA in the clock jitter scenario underscores that aligning only the kernel-embedding discrepancy or only the covariance is insufficient under stronger perturbations. RFA-SCA's success in this scenario suggests that combining kernel-embedding discrepancy reduction with covariance-level regularization is important for such shifts.

5.3 Component Analysis and Alignment Strategy

Building on the four key patterns identified in Section 4.3, we provide deeper analysis of the underlying mechanisms.

Comparing explicit moment matching with adversarial alignment, the explicit MMD+CORAL formulation of RFA-SCA performs better than AL-PA's implicit adversarial approach in these experiments; AL-PA fails in 4 out of 6 scenarios. The ablation study further indicates that explicit moment constraints provide more stable optimization. MMD-containing variants generally exhibit lower cross-seed variance than CORAL-only or CORAL+Ent variants, indicating that covariance-only alignment without kernel-embedding-based mean correction is less stable in these runs than MMD-containing formulations.

Turning to the joint effect of MMD and CORAL, the Clock Jitter scenario is a useful case. The variance reduction when CORAL is added to MMD is consistent with the analysis in Section 3.4.2: the Gaussian RBF kernel's finite-sample sensitivity to higher-order moments decays rapidly. This result indicates that under strong temporal countermeasures, which induce covariance-level feature discrepancies, explicit second-order regularization is important in our experiments. Conversely, the failure of CORAL+Ent suggests that covariance alignment without prior kernel-embedding-based correction operates in an incorrectly centered feature space, and conditional entropy amplifies this error by sharpening predictions toward wrong classes. This agrees with the ablation-level interpretation that MMD-based alignment supplies the stable base for additional corrections.

Conditional entropy regularization plays a distinctive role and is most useful when domain-level alignment is achieved but class boundaries remain ambiguous. The Gaussian Noise scenario exemplifies this, as additive noise ($\sigma = 8$) creates stochastic overlap between class-conditional distributions, and entropy minimization pushes decision boundaries away from these high-density overlap regions. However, this scenario also exhibits the highest seed sensitivity among all six scenarios. The Full variant achieves MTD = 4430 ± 1140 (coefficient of variation ≈ 26% vs. ≤13% elsewhere), and the single-seed result in Table 6 (MTD = 1901) corresponds to a particularly favorable initialization. This discrepancy shows that although RFA-SCA can recover the key under Gaussian noise, the outcome depends more on random initialization than in other scenarios, an inherent consequence of noise-induced class overlap, where the optimization landscape contains multiple local minima of varying quality.

5.4 Computational Overhead

The multi-layer domain adaptation design introduces additional computational cost compared to single-constraint methods, as forward passes must extract intermediate features at multiple layers and compute MMD and CORAL losses at each alignment point. In our experiments, the per-dataset wall-clock training time for a single RFA-SCA variant (Phase 2 fine-tuning) is approximately 1130 s for XMEGA (60 epochs), 140 s for CHES CTF (10 epochs), and 2000–4800 s for ASCAD variants (80–120 epochs) on a

single RTX 4090 GPU. These training costs are incurred only once during the offline profiling phase and do not affect the attack phase, where key recovery requires only a single forward pass per trace. We consider this overhead acceptable given the reported performance gains, particularly in challenging scenarios such as clock jitter where no baseline method succeeds.

From a theoretical perspective, the per-iteration computational complexity of the domain adaptation losses is $O\left(\sum_{l\in\mathcal{L}}(n^2 d_l + n\, d_l^2)\right)$, where n is the mini-batch size, d_l is the feature dimension at alignment layer l, and $\mathcal{L}$ denotes the set of alignment layers. Specifically, the MMD loss with a Gaussian RBF kernel requires $O(n^2 d_l)$ for pairwise kernel evaluations at each layer, and the CORAL loss requires $O(n\, d_l^2)$ for covariance matrix estimation and Frobenius-norm computation. In practice, n (typically 256–512) and $|\mathcal{L}|$ (2–4 layers) are small relative to the backbone forward pass, so the adaptation overhead constitutes a modest fraction of total training time.

6 Conclusions

This paper presented RFA-SCA, an unsupervised domain adaptation framework for side-channel analysis that combines kernel-embedding discrepancy reduction, covariance-level regularization, and target-domain decision-boundary refinement. The ablation study shows that the full combination provides the broadest scenario coverage, while individual components contribute unevenly across shift types. Across the six evaluated scenarios, RFA-SCA reaches GE@Max = 0 and SR@Max = 100%; in the scenarios where a successful baseline is available, it reduces the required traces by 14%–33%, and under clock jitter it is the only evaluated method that recovers the key. Our contribution is therefore a task-specific integration for cross-device SCA rather than a generic claim of a new hybrid UDA formulation.

The current evaluation also leaves several limitations. First, the experiments primarily focus on the Advanced Encryption Standard (AES). Future research will extend this framework to other cryptographic algorithms, including public-key cryptography (e.g., RSA, ECC) and post-quantum cryptographic implementations. Second, the loss weights α_{MMD}, α_{CORAL}, and α_{ENT} are empirically determined on a per-dataset basis. Although Section 4 identifies interpretable regularities linking the selected weights to feature dimensionality, shift type, and class cardinality, the exact values remain sensitive to the experimental setting. As in other multi-objective domain adaptation frameworks, such dataset-specific tuning, which is also required by baselines such as AL-PA and CDPA, limits direct deployment to new targets and motivates more systematic hyperparameter sensitivity analysis together with adaptive or self-tuning weighting strategies. The multi-layer adaptation strategy also uses dataset-specific alignment points; future work should compare single-layer and multi-layer configurations and develop principled layer-selection criteria. Finally, applying domain adaptation constraints across multiple network layers inherently increases the computational overhead during the profiling phase. Exploring lightweight adaptation strategies or optimal transport-based matching to reduce this computational burden remains an important direction for future work.

Acknowledgement: The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers and editors, which have improved the presentation.

Funding Statement: This research was funded by the National Natural Science Foundation of China (62071057, 62302036), BUPT innovation and entrepreneurship support program (2023-YC-A165,2025-YC-A018).

Author Contributions: Conceptualization, Hongxin Zhang and Yuanzhen Wang; methodology, Yuanzhen Wang; software, Yuanzhen Wang; validation, Yaqi Zhang, Xing Fang and Zhi Sun; formal analysis, Yuanzhen Wang; investigation, Shaofei Sun; resources, Hongxin Zhang and Shaofei Sun; data curation, Yaqi Zhang, Xing Fang and Zhi Sun; writing—original draft preparation, Yuanzhen Wang; writing—review and editing, Yuanzhen Wang, Hongxin Zhang and Shaofei Sun; visualization, Yuanzhen Wang; supervision, Hongxin Zhang; project administration, Hongxin Zhang and Shaofei

Sun; funding acquisition, Hongxin Zhang and Shaofei Sun. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study (ASCAD, CHES CTF 2018, SAKURA-G, XMEGA) are publicly available. The code will be made available upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Hyperparameter Search Ranges

Table A1: Hyperparameter search grids and local refinement rules for loss weights.

Parameter	Coarse Grid (Log-Spaced)	Local Refinement around Best Coarse Value c
α_{MMD}	$\{0.01, 0.05, 0.1, 0.5, 1.0, 5.0\}$	$\{\frac{2}{3}c,\ c,\ 1.6c\}$, clipped to $[0.01, 5.0]$
α_{CORAL}	$\{0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0\}$	$\{\frac{2}{3}c,\ c,\ 1.6c\}$, clipped to $[0.01, 10.0]$
α_{ENT}	$\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$	$\{0.5c,\ c,\ 2c\}$, clipped to $[0.001, 0.5]$
E_{warmup}	$\{0, 3, 5, 10, 20\}$	—

References

1. Kocher P, Jaffe J, Jun B. Differential power analysis. In: Annual International Cryptology Conference. Berlin/Heidelberg, Germany: Springer; 1999. p. 388–97.
2. Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model. In: International Workshop on Cryptographic Hardware and Embedded Systems. Berlin/Heidelberg, Germany: Springer; 2004. p. 16–29.
3. Moos T, Moradi A, Richter B. Static power side-channel analysis—an investigation of measurement factors. IEEE Trans Very Large Scale Integr (VLSI) Syst. 2019;28(2):376–89. doi:10.1109/tvlsi.2019.2948141.
4. Zhao M, Suh GE. FPGA-based remote power side-channel attacks. In: Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP); 2018 May 20–24; San Francisco, CA, USA. New York, NY, USA: IEEE. p. 229–44.
5. Haas G, Aysu A. Apple vs. EMA: electromagnetic side channel attacks on apple CoreCrypto. In: Proceedings of the 59th ACM/IEEE Design Automation Conference; 2022 Jul 10–14; San Francisco, CA, USA. p. 247–52.
6. Zhang J, Chen C, Cui J, Li K. Timing side-channel attacks and countermeasures in CPU microarchitectures. ACM Comput Surv. 2024;56(7):1–40. doi:10.1145/3645109.
7. Ali U, Sahni SAR, Khan O. Characterization of timing-based software side-channel attacks and mitigations on network-on-chip hardware. ACM J Emerg Technol Comput Syst. 2023;19(3):1–23. doi:10.1145/3585519.
8. Prouff E, Rivain M. Masking against side-channel attacks: a formal security proof. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques; 2013 May 26–30; Athens, Greece. Berlin/Heidelberg, Germany: Springer; 2013. p. 142–59.
9. Veyrat-Charvillon N, Medwed M, Kerckhof S, Standaert FX. Shuffling against side-channel attacks: a comprehensive study with cautionary note. In: Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security; 2012 Dec 4–8; Beijing, China. Berlin/Heidelberg, Germany: Springer; 2012. p. 740–57.
10. Levi I, Bellizia D, Bol D, Standaert FX. Ask less, get more: side-channel signal hiding, revisited. IEEE Trans Circuits Syst I Regul Pap. 2020;67(12):4904–17.
11. Maghrebi H, Portigliatti T, Prouff E. Breaking cryptographic implementations using deep learning techniques. In: Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering; 2016 Dec 16–18; Hyderabad, India. Berlin/Heidelberg, Germany: Springer; 2016. p. 3–26.

12. Zaid G, Bossuet L, Habrard A, Venelli A. Methodology for efficient CNN architectures in profiling attacks. IACR Trans Cryptogr Hardw Embed Syst. 2020;2020(1):1–36. doi:10.46586/tches.v2020.i1.1-36.
13. Lerman L, Bontempi G, Markowitch O. Power analysis attack: an approach based on machine learning. Int J Appl Cryptogr. 2014;3(2):97–115. doi:10.1504/ijact.2014.062722.
14. Bhasin S, Chattopadhyay A, Heuser A, Jap D, Picek S, Ranjan R. Mind the portability: a warriors guide through realistic profiled side-channel analysis. In: Proceedings of the NDSS 2020-Network and Distributed System Security Symposium; 2020 Feb 23–26; San Diego, CA, USA. p. 1–14.
15. Sun B, Saenko K. Deep coral: correlation alignment for deep domain adaptation. In: Proceedings of the European Conference on Computer Vision; 2016 Oct 11–14; Amsterdam, The Netherlands. Berlin/Heidelberg, Germany: Springer; 2016. p. 443–50.
16. Courty N, Flamary R, Tuia D, Rakotomamonjy A. Optimal transport for domain adaptation. IEEE Trans Pattern Anal Mach Intell. 2016;39(9):1853–65. doi:10.1109/tpami.2016.2615921.
17. Nassif S. Delay variability: sources, impacts and trends. In: Proceedings of the 2000 IEEE International Solid-State Circuits Conference; 2000 Feb 9; San Francisco, CA, USA. New York, NY, USA: IEEE; 2000. p. 368–9. Digest of Technical Papers (Cat. No. 00CH37056).
18. Bowman KA, Duvall SG, Meindl JD. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. IEEE J Solid-State Circuits. 2002;37(2):183–90. doi:10.1109/4.982424.
19. Cagli E, Dumas C, Prouff E. Convolutional neural networks with data augmentation against jitter-based countermeasures: profiling attacks without pre-processing. In: Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems; 2017 Sep 25–28; Taipei, Taiwan. Cham, Switzaerland: Springer; 2017. p. 45–68.
20. Awad SS. Analysis of accumulated timing-jitter in the time domain. IEEE Trans Instrum Meas. 2002;47(1):69–73. doi:10.1109/19.728792.
21. Benadjila R, Prouff E, Strullu R, Cagli E, Dumas C. Deep learning for side-channel analysis and introduction to ASCAD database. J Cryptogr Eng. 2020;10(2):163–88.
22. Das D, Golder A, Danial J, Ghosh S, Raychowdhury A, Sen S. X-DeepSCA: cross-device deep learning side channel attack. In: Proceedings of the 56th Annual Design Automation Conference 2019; 2019 Jun 2–6; Las Vegas, NV, USA. p. 1–6.
23. Danial J, Das D, Golder A, Ghosh S, Raychowdhury A, Sen S. EM-X-DL: efficient cross-device deep learning side-channel attack with noisy EM signatures. ACM J Emerg Technol Comput Syst (JETC). 2021;18(1):1–17. doi:10.1145/3465380.
24. Zhang F, Shao B, Xu G, Yang B, Yang Z, Qin Z, et al. From homogeneous to heterogeneous: leveraging deep learning based power analysis across devices. In: Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC); 2020 Jul 20–24; San Francisco, CA, USA. New York, NY, USA: IEEE; 2020. p. 1–6.
25. Wu L, Won YS, Jap D, Perin G, Bhasin S, Picek S. Ablation analysis for multi-device deep learning-based physical side-channel analysis. IEEE Trans Dependable Secur Comput. 2023;21(3):1331–41. doi:10.1109/tdsc.2023.3278857.
26. Golder A, Das D, Danial J, Ghosh S, Sen S, Raychowdhury A. Practical approaches toward deep-learning-based cross-device power side-channel attack. IEEE Trans Very Large Scale Integr (VLSI) Syst. 2019;27(12):2720–33. doi:10.1109/tvlsi.2019.2926324.
27. Montminy DP, Baldwin RO, Temple MA, Laspe ED. Improving cross-device attacks using zero-mean unit-variance normalization. J Cryptogr Eng. 2013;3(2):99–110. doi:10.1007/s13389-012-0038-y.
28. Cao P, Zhang C, Lu X, Gu D. Cross-device profiled side-channel attack with unsupervised domain adaptation. IACR Trans Cryptogr Hardw Embed Syst. 2021;2021(4):27–56. doi:10.46586/tches.v2021.i4.27-56.
29. Li X, Yang N, Liu W, Chen A, Zhang Y, Wang S, et al. Domain-adaptive power profiling analysis strategy for the metaverse. Int J Netw Manag. 2025;35(1):e288. doi:10.22541/au.171668883.35894850/v1.
30. Cao P, Zhang H, Gu D, Lu Y, Yuan Y. AL-PA: cross-device profiled side-channel attack using adversarial learning. In: Proceedings of the 59th ACM/IEEE Design Automation Conference; 2022 Jul 10–14; San Francisco, CA, USA. p. 691–6.

31. Cui X, Zhang H, Fang X, Wang Y, Wang D, Fan F, et al. A secret key classification framework of symmetric encryption algorithm based on deep transfer learning. Appl Sci. 2023;13(21):12025. doi:10.3390/app132112025.
32. Zhang Q, Zhang H, Cui X, Fang X, Wang X. Side channel analysis of speck based on transfer learning. Sensors. 2022;22(13):4671.
33. Yu H, Shan H, Panoff M, Jin Y. Cross-device profiled side-channel attacks using meta-transfer learning. In: Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC); 2021 Dec 5–9; San Francisco, CA, USA. New York, NY, USA: IEEE; 2021. p. 703–8.
34. Yu H, Wang M, Song X, Shan H, Qiu H, Wang J, et al. Noise2clean: cross-device side-channel traces denoising with unsupervised deep learning. Electronics. 2023;12(4):1054.
35. Woo JE, Jeon Y, Kim JH, Han DG. Novel deep learning-based side-channel attack on different-device: novel deep learning-based side-channel attack on different-device. Soft Comput. 2025;29(8):3847–54. doi:10.1007/s00500-025-10610-2.
36. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A kernel two-sample test. J Mach Learn Res. 2012;13(1):723–73.
37. Chen C, Fu Z, Chen Z, Jin S, Cheng Z, Jin X, et al. HoMM: higher-order moment matching for unsupervised domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2020 Feb 7–12; New York, NY, USA. Vol. 34, p. 3422–9.
38. Zellinger W, Grubinger T, Lughofer E, Natschläger T, Saminger-Platz S. Central moment discrepancy (CMD) for domain-invariant representation learning. In: Proceedings of the International Conference on Learning Representations; 2017 Apr 24–26; Toulon, France.
39. Picek S, Heuser A, Jovic A, Ludwig SA, Guilley S, Jakobovic D, et al. Side-channel analysis and machine learning: a practical perspective. In: Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN); 2017 May 14–19; Anchorage, AK, USA. New York, NY, USA: IEEE; 2017. p. 4095–102.
40. Wouters L, Arribas V, Gierlichs B, Preneel B. Revisiting a methodology for efficient CNN architectures in profiling attacks. IACR Trans Cryptogr Hardw Embed Syst. 2020;2020(3):147–68. doi:10.46586/tches.v2020.i3.147-168.
41. Bronchain O, Standaert FX. Breaking masked implementations with many shares on 32-bit software platforms: or when the security order does not matter. IACR Trans Cryptogr Hardw Embed Syst. 2021;2021(3):202–34. doi:10.46586/tches.v2021.i3.202-234.
42. Masure L, Belleville N, Cagli E, Cornelie MA, Couroussé D, Dumas C, et al. Deep learning side-channel analysis on large-scale traces: a case study on a polymorphic AES. In: Proceedings of the European Symposium on Research in Computer Security. Cham, Switzerland: Springer; 2020. p. 440–60.
43. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. Mach Learn. 2010;79(1):151–75. doi:10.1007/s10994-009-5152-4.
44. Long M, Cao Y, Wang J, Jordan M. Learning transferable features with deep adaptation networks. In: Proceedings of the International Conference on Machine Learning; 2015 Jul 6–11; Lille, France. Cambridge, MA, USA: PMLR; 2015. p. 97–105.
45. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, et al. Domain-adversarial training of neural networks. J Mach Learn Res. 2016;17(59):1–35. doi:10.1007/978-3-319-58347-1_10.
46. Kou X, Yang W, Li L, Zhang G. Multi-modal side-channel analysis based on isometric compression and combined clustering. Symmetry. 2025;17(9):1483. doi:10.3390/sym17091483.
47. Grandvalet Y, Bengio Y. Semi-supervised learning by entropy minimization. In: Proceedings of the 18th International Conference on Neural Information Processing Systems; 2004 Dec 1; Vancouver, BC, Canada. p. 529–36.
48. Liu A, Wang A, Sun S, Wei C, Ding Y, Wang Y, et al. CL-SCA: a contrastive learning approach for profiled side-channel analysis. IEEE Trans Inf Forensics Secur. 2025;20:5109–22. doi:10.1109/tifs.2025.3570123.
49. Hajra S, Saha S, Alam M, Mukhopadhyay D. Transnet: shift invariant transformer network for side channel analysis. In: Proceedings of the International Conference on Cryptology in Africa; 2022 Jul 18–20; Fes, Morocco. Cham, Switzerland: Springer; 2022. p. 371–96.
50. Muandet K, Fukumizu K, Sriperumbudur B, Schölkopf B. Kernel mean embedding of distributions: a review and beyond. Found Trends Mach Learn. 2017;10(1–2):1–141.

51. Tolstikhin IO, Sriperumbudur BK, Schölkopf B. Minimax estimation of maximum mean discrepancy with radial kernels. In: Proceedings of the 30th International Conference on Neural Information Processing Systems; 2016 Dec 5–10; Red Hook, NY, USA. p. 1938–46.
52. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems 27 (NeurIPS 2014). Cambridge, MA, USA: MIT Press; 2014. p. 3320–8.