IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2025, No. 3, pp. 164–211. DOI:10.46586/tches.v2025.i3.164-211

Optimal Dimensionality Reduction using Conditional Variational AutoEncoder

Sana Boussam[2,3], Mathieu Carbone[3], Benoît Gérard[1], Guénaël Renault[1,2] and Gabriel Zaid[3]

[1] ANSSI, Paris, France
{firstname.lastname}@ssi.gouv.fr
[2] LIX, INRIA, CNRS, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France
{firstname.lastname}@inria.fr
[3] Thales ITSEF, Toulouse, France
{firstname.lastname}@thalesgroup.com

Abstract. The benefits of using Deep Learning techniques to enhance side-channel attacks performances have been demonstrated over recent years. Most of the work carried out since then focuses on discriminative models. However, one of their major limitations is the lack of theoretical results. Indeed, this lack of theoretical results, especially concerning the choice of neural network architecture to consider or the loss to prioritize to build an optimal model, can be problematic for both attackers and evaluators. Recently, Zaid *et al.* addressed this problem by proposing a generative model that bridges conventional profiled attacks and deep learning techniques, thus providing a model that is both explicable and interpretable. Nevertheless the proposed model has several limitations. Indeed, the architecture is too complex, higher-order attacks cannot be mounted and desynchronization is not handled by this model. In this paper, we address the first limitation namely the architecture complexity, as without a simpler model, the other limitations cannot be treated properly. To do so, we propose a new generative model that relies on solid theoretical results. This model is based on conditional variational autoencoder and converges towards the optimal statistical model *i.e.* it performs an optimal attack. By building on and extending the state-of-the-art theoretical works on dimensionality reduction, we integrate into this neural network an optimal dimensionality reduction *i.e.* a dimensionality reduction that is achieved without any loss of information. This results in a gain of $\mathcal{O}(D)$, with D the dimension of traces, compared to Zaid *et al.* neural network in terms of architecture complexity, while at the same time enhancing the explainability and interpretability. In addition, we propose a new attack strategy based on our neural network, which reduces the attack complexity of generative models from $\mathcal{O}(N)$ to $\mathcal{O}(1)$, with N the number of generated traces. We validate all our theoretical results experimentally using extensive simulations and various publicly available datasets covering symmetric, asymmetric pre and post-quantum cryptography implementations.

Keywords: Side-Channel Attacks · Deep Learning · Optimal Dimensionality Reduction · Optimal Attacks · Generative Models · Variational AutoEncoder

1 Introduction

Context. Side-Channel Attacks (SCA) are a type of physical attacks in which an attacker exploits implementation weaknesses of a cryptographic algorithm in embedded systems in order to extract secret information, by targeting all types of physical leaks (*e.g.* execution time [Koc96], power consumption [KJJ99], electromagnetic radiation [AARR03], etc.)

Licensed under Creative Commons License CC-BY 4.0.
Received: 2025-01-15 Accepted: 2025-03-15 Published: 2025-06-05

generated during the execution of this algorithm. Among the various types of SCA possible, the attacker can consider *profiled attacks* as it is one of the most effective attacks. *Profiled attacks* take place in two stages and require the attacker to have access to an open device identical to the one he intends to attack, which he controls completely. The aim of this type of attack is to solve a classification problem by estimating the unknown probability distribution conditioned by the secret information the attacker wants to recover. Template-based attacks [CRR03] and stochastic-based attacks [SLP05] are examples of conventional *profiled attacks*. These attacks are relevant because they derive directly from theoretical results, making them interpretable and explicable. Template and stochastic-based attacks are consequently interesting and beneficial for evaluators, as their theoretical properties offer them a quick and effective method for retrieving secret information [HRG14]. However, the major limitation of these conventional attacks lies in the fact that a preliminary pre-processing step on traces is required to ensure a correct estimation of the unknown conditional probability distribution. Recently, the use of deep learning techniques in the context of SCA has emerged to address these limitations (Deep Learning based side-channel attacks or DLSCA). Indeed, the use of neural networks to carry out *profiled attacks* is advantageous, as they automatically handle the pre-processing step while solving a classification problem by directly estimating the unknown conditional probability distribution [MPP16, CDP17]. So far, the only approach considered in DLSCA state-of-the-art is the use of discriminative models as it leads to better attacks performance [PHJ+18, ZBHV20, WPP22]. Nevertheless, while this approach addresses limitations of conventional attacks and leads to more successful attacks, it also raises new challenges. Indeed, given the limited theoretical results on discriminative models [MDP19, IUH22, IUH24], it is difficult to explain the decision-making process of these models, resulting thus in using them as black boxes (*i.e.* this results in a loss of interpretability and explicability). To avoid this black-box approach, the discriminative models used are highly complex, so as to define more generic functions. This makes them more difficult to train for an attacker because their architecture is difficult to both configure and justify. As a result, the total number of hyperparameters an evaluator has to take into account increases dramatically, since architecture hyperparameters (*e.g.* number of layers, number of neurons for each layer...) are added to those related to training. This therefore represents a serious problem for an evaluator seeking to extract information in a restricted time frame. Indeed, the latter cannot be assured that his models will end up converging towards the true unknown conditional probability distribution.

Generative models in SCA. To circumvent these issues, Zaid *et al.* proposed an architecture based on generative models which combines theoretical results of conventional *profiled attacks* and deep learning techniques [ZBC+23]. Hence, this work introduces the first deep learning based model in DLSCA literature that is both explainable and interpretable. However, the use of this model is recent and suffers from many limitations *i.e.* a high architecture complexity and no possibility of dealing with desynchronization or performing higher-order attacks without pre-processing. The purpose of this paper is to address the first limitation since without a less complex model, it is not feasible to tackle the remaining limitations. One way of reducing architecture complexity while preserving attack performance is to perform a dimensionality reduction within the neural network, without any loss of information to retrieve the secret. Optimal dimensionality reduction *i.e.* dimensionality reduction without loss of information is a topic that has been addressed in SCA [BGH+15] but has never yet been integrated into generative models. Indeed, usually, when using a generative model, the dimensionality reduction step is carried out beforehand, outside the model. To do this, an attacker can apply dimensionality reduction techniques such as Linear Discriminant Analysis (LDA) [SA08]. This technique is particularly interesting, as previous studies have shown that it asymptomatically performs

an optimal dimensionality reduction [BGH+15]. However, this technique suffers from *Small Sample Size* (SSS) or *High-Dimension Low Sample Size* (HDLSS) problem, which makes it unusable in practice [JSLH22]. Our work aims to address both the existing limitations of generative models and those of LDA through theory-based investigations.

Contributions. The main contribution of our paper is the proposition of a generative model, called **cVAE-OSM**, that converges towards the optimal statistical model [HRG14] while having a better architecture complexity than the state-of-the-art generative models [ZBC+23]. Moreover, based on this model, we also propose a new **end-to-end profiled attack strategy** that reduces attack complexity, while preserving the same performances as those achieved with generative models [ZBC+23].

cVAE-OSM is a new generative model based on conditional variational autoencoder with a simplified and explainable architecture that performs an optimal dimensionality reduction. The integration of this optimal dimensionality reduction into the neural network enables a gain in complexity by a factor of $\mathcal{O}(D)$ (D being the number of samples per traces) compared to state-of-the-art DLSCA generative models (see Proposition 1), with no information loss. We experimentally assess and validate through simulations all cVAE-OSM theoretical properties.

Based on this model, we develop an end-to-end profiled attack strategy allowing the complexity of an attack using DLSCA generative models to drop down from a $\mathcal{O}(N)$ complexity [ZBC+23] to a $\mathcal{O}(1)$ complexity, where N is the number of generated traces (see Proposition 4). We experimentally assess and validate our claims by comparing attack performances of our model to those of state-of-the-art SCA models (*i.e.* discriminative/generative models and conventional profiled attacks) using publicly available datasets covering symmetric and asymmetric/post-quantum cryptography implementations with and without countermeasures.

To properly design this new generative model, we also provide further theoretical contributions. First, we extend in Theorem 2 Bruneau *et al.*'s work on optimal dimensionality reduction [BGH+15] to fit to a more generic context as their results are not longer applicable when considering a multivariate leakage model. Based on this extended optimal dimensionality reduction, we therefore design a conditional variational autoencoder architecture that is fully explicable and interpretable compared to state-of-the-art work on DLSCA generative models [ZBC+23] *i.e.* deterministic architecture in which all the weights included correspond to predefined quantities namely the deterministic part and variance of traces. Moreover, to ensure that our model converges correctly towards the optimal statistical model [HRG14], we design a customized loss, adapted to our problem through Proposition 2 and Proposition 3. Based on this loss, we provide proofs of convergence for all weights of our architecture *i.e.* all the weights of the proposed architecture converge towards expected values namely the deterministic part and variance of traces (see Theorem 5 and Theorem 6).

Finally, all these theoretical contributions are also supplemented by complementary results that are also experimentally validated. In fact, besides providing a model that allows to conduct an optimal attack, our model can also be used as a dimensionality reduction tool, without any practical constraints. Indeed, the use of neural networks enables our proposed model to overcome the SSS/HDLSS problem, from which LDA suffers [JSLH22]. In addition, our model offers a more generic dimensionality reduction than LDA: no assumption needs to be made about the leakage model, as this is handled directly by our neural network. The whole implementation of cVAE-OSM model, as well as notebooks containing all the experiments carried out in this article, are available in the following GitHub repository: https://github.com/Sbsm157/cVAE-OSM.

Organization. Our paper is organized as follows: in Section 2, we define our notation and recall important concepts that will be reused throughout the paper. Then, we present in Section 3 our explicable neural network architecture that converges towards the optimal statistical model *i.e.* it allows to both perform an optimal attack and optimal dimensionality reduction. In Section 4, we propose a detailed description of the learning strategy allowing our neural network to be fully interpretable and then present an end-to-end profiled attack using our model. We experimentally validate all the theoretical claims made in the previous sections in Section 5 using simulations and a wide range of publicly available datasets. Finally, we conclude in Section 6.

2 Background

In this section, we define notations, recall definitions and summarize notions that will be used in the rest of the paper.

2.1 Notations and definitions

Probability theory. Calligraphic letters as $\mathcal{X}$ denote sets. The cardinality of a given set $\mathcal{X}$ -if finite- is denoted $|\mathcal{X}|$. The corresponding bold capital letter $\mathbf{X}$ denotes a random variable over $\mathcal{X}$. The realization of $\mathbf{X}$ is denoted by its corresponding bold lowercase $\mathbf{x}$.

Let $\mathbf{X}$ be a random variable and $\mathbf{x}$ an event of $\mathbf{X}$, we denote $\mathbb{P}(\mathbf{X} = \mathbf{x})$ also written $\mathbb{P}(\mathbf{x})$ (resp. $\mathbb{P}(\mathbf{X} = \mathbf{x}|\mathbf{Y} = \mathbf{y})$ also written $\mathbb{P}(\mathbf{x}|\mathbf{y})$) the probability of observing $\mathbf{x}$ (resp. the conditional probability of observing $\mathbf{x}$ knowing a realization $\mathbf{y}$ of the random variable $\mathbf{Y}$). A Θ-parametric conditional probability $\mathbb{P}(\mathbf{X}|\mathbf{Y}, \Theta)$ is an approximation of the true unknown conditional probability $\mathbb{P}(\mathbf{X}|\mathbf{Y})$ given a parameter set Θ.

We denote the first moment (*i.e.* expected value) and second moment (*i.e.* variance) of $\mathbf{X}$ as $\mathbb{E}[\mathbf{X}]$ and $\mathbb{V}[\mathbf{X}]$ respectively. The expected value (resp. variance) of the random variable $\mathbf{Y}$ over the distribution of $\mathbf{X}$ is denoted $\mathbb{E}_{\mathbf{X}}[\mathbf{Y}]$ (resp. $\mathbb{V}_{\mathbf{X}}[\mathbf{Y}]$).

The notation $\mathbf{X} \sim \mathcal{N}_D(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (resp. $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma^2})$) refers to a random vector $\mathbf{X}$ of dimension D (resp. a random scalar) that follows a multivariate Gaussian distribution (resp. Gaussian distribution) of parameters $\boldsymbol{\mu} \in \mathbb{R}^D$ (resp. $\boldsymbol{\mu} \in \mathbb{R}$) and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ (resp. $\boldsymbol{\sigma^2} \in \mathbb{R}$). Depending on whether $\mathbf{X}$ is of dimension $D > 1$ or $D = 1$, the mean and covariance matrix or variance of $\mathbf{X}$ is denoted $\boldsymbol{\mu_X}$ and $\boldsymbol{\Sigma_X}$ or $\sigma^2_{\mathbf{X}}$ respectively.

For proving some of our convergence results, we use the Kullback-Leibler (KL) divergence $\mathcal{D}_{KL}$, a statistical divergence measure derived from information theory that quantifies the difference between two probability distributions. Let $\mathbb{P}$ and $\mathbb{Q}$ be two discrete probability distributions on $\mathcal{X}$, $\mathcal{D}_{KL}$ is defined as

$$\mathcal{D}_{KL}(\mathbb{P} \,||\, \mathbb{Q}) = \sum_{\mathbf{x} \in \mathcal{X}} \mathbb{P}(\mathbf{x}) \log \Big(\frac{\mathbb{P}(\mathbf{x})}{\mathbb{Q}(\mathbf{x})} \Big).$$

This quantity is always non-negative and equals zero if and only if $\mathbb{P} = \mathbb{Q}$.

Linear algebra. Throughout this paper, we adopt a matrix notation. Consequently, the ith entry of a vector A is denoted A_i. For a given matrix M, the $(i, j)^{th}$ entry (*i.e.* the entry of the ith row and j^{th} column) is denoted M_{ij}. The same notation is used for a random vector $\mathbf{A}$ or a random matrix $\mathbf{M}$. We use the following notation $\mathbf{k^N}$ and $\mathbf{k^{M \times N}}$ to refer to a N-dimensional vector and $M \times N$-dimensional matrix where all the entries are the same constant k. The transpose and inverse operator will be denoted $(\cdot)^T$ and $(\cdot)^{-1}$ respectively. The identity matrix of size $D \times D$ is denoted $\mathbf{I_D}$. Finally, for the sake of simplicity, the diagonal matrix $M \in \mathbb{R}^{D \times D}$ is denoted $M = \text{diag}[m_1, \ldots, m_D]$ in this paper.

Side Channel Attacks terminology. In this work, $\mathcal{K}$ refers to the set of potential keys where $k^* \in \mathcal{K}$ is the correct secret key. A leakage trace $\mathbf{T} \in \mathbb{R}^D$ is a random vector which can be decomposed into two parts, a deterministic one and a random one:

$$\mathbf{T} = \Psi\big(Y^{k^*}\big) + \mathbf{N},$$

where $Y^{k^*} = f(P, k^*)$ is a sensitive value which results from the application of a cryptographic operation f given a plaintext P and the secret key k^*. The first summand (*i.e.* the Ψ function) represents the deterministic part of the trace and aims at modeling the dependence between Y^{k^*} and the side-channel observables $\mathbf{T}$. This function is also known as the leakage model. There are several ways to characterize it. In this paper, we opt for the most generic way of characterizing this leakage model *i.e.* we consider Ψ as a pseudo-boolean function [Car10]. Ψ can thus be written using a multi-linear polynomial where each monomial represents the contribution of bits' interactions of Y^k for a given key $k \in \mathcal{K}$. Let consider S as the maximal bit length of values taken by Y^k and $(Y^k)_b$ as the bth bit of Y^k. Y^k can be decomposed as the following multi-linear polynomial Γ of maximal degree S:

$$\Gamma(Y^k) = 1 + \sum_{i=1}^{S} \sum_{1 \leq j_1 < \ldots < j_i \leq S} r\big((Y^k)_{j_1}, (Y^k)_{j_2}, ..., (Y^k)_{j_i}\big), \tag{1}$$

where $j_i \in [1, S]$ and r is some boolean operations that characterize bits' interactions, *e.g.* a XOR or an AND. The degree 0 term corresponds to the bias (*i.e.* when no interactions between bits are considered). As the maximal degree of $\Gamma(Y^k)$ is S, this polynomial is therefore composed of $\sum_{i=0}^{S} \binom{S}{i} = 2^S$ monomials, each representing a bits' interaction. For the sake of simplicity, $\Gamma(Y^k)$ will be considered as vector in which each entry corresponds to a monomial instead of a polynomial *i.e.* $\Gamma(Y^k) \in \mathbb{F}_2^{2^S}$. Hence, based on this decomposition, we can formally define the leakage model Ψ as:

$$\begin{array}{rccc} \Psi & : & \mathbb{F}_2^S & \rightarrow & \mathbb{R}^D \\ & & Y^k & \mapsto & \alpha \cdot \Gamma(Y^k) \end{array} \tag{2}$$

where $\alpha \in \mathbb{R}^{D \times 2^S}$ is a matrix of coefficients representing contribution of each monomial/bits' interaction for all D time samples. We note $\mathcal{B}$ the basis on which Γ is projected. One of the bases we essentially consider in the rest of our paper is Guilley *et al.* orthonormal basis [GHMR17]. This basis choice is based on the fact that, thanks to its orthonormality, it allows a clear and comprehensible reading of the contribution of each monomial/bits' interaction (see Section 5.1.1).

Following the Gaussian independent noise assumption, we assume that the random part $\mathbf{N}$ of $\mathbf{T}$ follows a multivariate normal distribution $\mathcal{N}_D(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Moreover, we assume without loss of generality that the noise is centered, *i.e.* $\boldsymbol{\mu} = \mathbf{0^D}$. We define a trace $\mathbf{T}$ with a white Gaussian noise as a trace where the random part $\mathbf{N} \sim \mathcal{N}_D(\mathbf{0^D}, \boldsymbol{\Sigma})$ is such that $\boldsymbol{\Sigma} = \mathrm{diag}[\boldsymbol{\sigma}_1^2, \ldots, \boldsymbol{\sigma}_D^2]$.

This means that we consider independent and identically distributed noise values at each time sample $i \in [1, D]$ of $\mathbf{T}$ and thus $\mathbf{T} \sim \mathcal{N}_D(\Psi(Y^{k^*}), \boldsymbol{\Sigma})$. This approximation is sound, since previous work in SCA literature have shown that the information contained in the covariance matrix $\boldsymbol{\Sigma}$ is essentially located in its diagonal [BGH+15].

Machine Learning terminology. In this paper, we distinguish *explainability* and *interpretability* concepts. Indeed, while these two notions are closely related, both apply to different scales. The notion of *explainability* is associated with a more global vision of the neural network and is formally defined as follows:

Definition 1. Explainability. A model qualified as *explainable* or *explicable* when it is easy to identify and understand the purpose of all layers involved in its decision-making process. In the context of DLSCA, dimensionality reduction, leakage model extraction or impact of countermeasures mitigations are examples of operations involved in the model's decision-making process.

Whereas the notion of *interpretability* focuses on the internal aspects of the model under consideration. It can be formally defined as:

Definition 2. Interpretability. A model qualified as *interpretable* when all its inner variables *i.e.* its weights and bias correspond to quantities clearly defined, and for which it is possible to determine, thanks to the optimization process, towards which values they converge.

2.2 Deep Learning based Side Channel Attacks

The goal of machine learning is to model an unknown (non-)linear function g that underlies the predictive relationship between data and labels. Depending on the application context and the data type, the function g that solves this relationship can be obtained by two different learning strategies: *supervised* and *unsupervised* learning algorithms. Throughout this paper, only the former strategy will be considered. Supervised learning algorithms construct solutions to problems asking to provide predictions on the basis of labeled examples (*i.e. training dataset*) *e.g.* decision trees, neural networks [JWHT13].

One of the goals of SCA can be reduced to solving an instantiation of a classification problem as described below.

Problem 1. ***SCA classification problem.*** *Let $k \in \mathcal{K}$ be a secret key, $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_Q\}$ be a set of Q side-channel traces. Let us also define $\mathcal{Y} = \{y_1, \ldots, y_P\}$ as the set of values that can be taken by k. In the context of a SCA, the goal of an attacker is to correctly associate each leakage trace $\mathbf{t}_i$ for $i \in [1, Q]$ with the correct value of k namely y_j for $j \in [1, P]$.*

To address this problem, an attacker can consider one type of attack namely *profiled attacks* [CRR03,SLP05]. Theoretically, it is the most powerful SCA but it requires an open device similar to the targeted one in order to build a learning database that models the device behavior (*i.e.* the leakage model) according to the values of sensitive variables that depend on the secret key. To conduct such attacks, an adversary needs to detect and select a subset of informative points that are known as *points of interest* (denoted PoIs). A PoI corresponds to a time sample in a trace where the observed signal depends on secret key value manipulation. Indeed, to ensure a powerful attack, it is imperative that an attacker defines all the PoIs in order to reduce the uninformative time samples that can alter the attack performance. This preliminary attack step can quickly become very difficult and new approaches based on machine learning have recently emerged to overcome these constraints [PPM+23]. Indeed, as explained previously, the goal of machine learning is to model an unknown (non-)linear function. Thus, using *supervised* machine learning techniques in the context of SCA would maximize chances of recovering the most likely key because these techniques would be able to model automatically, from collected side-channel information, a function capable of estimating the number of needed PoIs, selecting them and then applying a weighting on these points in order to find the secret key [MPP16,CDP17]. Over the recent years, attention has focused more on *profiled attacks*, more specifically on how to link them to *supervised* deep learning algorithms [MDP19, ZZN+20, ZBD+20, IUH22]. Additional works investigate the suitability of some machine learning models *e.g.* autoencoders for enhancing pre-processing methods such that denoising, feature extraction and dimensionality reduction [WP20, KKH21, LYW+22, PBA21, Bot22].

In the following sections, we propose to theoretically and experimentally extend the Zaid *et al*'s work, which applied the conditional variational autoencoders into the SCA context [ZBC+23], by incorporating the notion of *optimal dimensionality reduction* while both maintaining the explainability and enhancing the interpretability of the underlying model.

2.3 Optimal distinguisher in SCA

In order to properly define the notion of optimal dimensionality reduction, we first need to introduce the concept of *optimal distinguisher in SCA*, from which the former is derived. By drawing a parallel between communication theory and SCA, Heuser *et al.* [HRG14] show that the optimal distinguisher in SCA depends only on the noise distribution and corresponds to the *maximum likelihood* when the leakage model is known *i.e.* when the leakage prediction made by the attacker coincides exactly with deterministic part Ψ of the leakage. This optimal distinguisher is defined as follows:

Definition 3. Optimal distinguisher [HRG14]. Given a key hypothesis $k \in \mathcal{K}$, a set of plaintexts $\mathcal{P}$ and a set of Q side-channel traces $\mathcal{T}$ such that $\mathbf{t}_q = \Psi(y_q^{k^*}) + \mathbf{n}_q$, where Ψ corresponds to the leakage model associated with targeted sensitive value $y_q^{k^*} = f(p_q, k^*)$, the optimal distinguisher $\mathcal{D}_{opt}^k$ is defined as

$$\mathcal{D}_{opt}^k(\mathcal{T}, \mathcal{P}) = \sum_{q=1}^{Q} \log\big(\mathbb{P}(\mathbf{t}_q \,|\, p_q, k)\big) = \sum_{q=1}^{Q} \log\Big(\mathbb{P}_{\mathbf{n}_q}\big(\mathbf{t}_q - \Psi(y_q^k)\big)\Big),$$

where $\mathbb{P}_{\mathbf{n}_q}$ denotes noise probability distribution.

Based on this optimal distinguisher, we can now formally define the optimal attack concept in SCA:

Definition 4. Optimal attack [HRG14]. The optimal attack based on the distinguishing rule $\mathcal{DR}_{opt}$ consists in finding the key candidate k that maximizes this optimal distinguisher $\mathcal{D}_{opt}^k$.

$$\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P}) = \arg\max_k \mathcal{D}_{opt}^k(\mathcal{T}, \mathcal{P}).$$

This optimal attack, under the assumption of Gaussian noise, corresponds to *template attacks* [CRR03] which are the SCA denomination of the Quadratic Discriminant Analysis (QDA) [Smi46, HTF01] commonly used in machine learning. Actually, if we add an assumption of homoscedacity *i.e.* same covariance matrix $\mathbf{\Sigma}$ for all classes, it corresponds to the *pooled* version of *template attacks* [CK13] also referred to as Linear Discriminant Analysis (LDA) [Fis36, HTF01]. LDA can also be used as a dimensionality reduction algorithm, which is its main use in SCA [SA08].

2.4 Optimal dimensionality reduction in SCA

Given $\mathbf{T}$ a D-dimensional leakage trace such that $\mathbf{T} \sim \mathcal{N}_D\big(\Psi(Y^{k^*}), \mathbf{\Sigma}\big)$ where each sample $\mathbf{T}_i = \big(\Psi(Y^{k^*})\big)_i + \mathbf{N}_i = \alpha_i \cdot l(Y^{k^*}) + \mathbf{N}_i$, Bruneau *et al.* show, following on from Heuser *et al.*'s work, that the optimal attack integrates an optimal dimensionality reduction [BGH+15] considering a restricted leakage model Ψ. This leakage model, defined in [BGH+15], corresponds to the weighting of a certain leakage function l (*e.g.* Hamming Weight) such that $\Psi : Y^k \mapsto \alpha \cdot l(Y^k)$ where $\alpha \in \mathbb{R}^D$ is a coefficients vector. In other words, Ψ is defined such that there exists a linear relationship between Ψ and the leakage function l *i.e.* for all traces, we associate with each sample i the same α_i without taking into account bits and bits' interaction values taken by Y^k.

Theorem 1. ***Optimal dimensionality reduction [BGH+15].*** *The optimal attack on the multivariate traces* $\mathbf{T} \in \mathbb{R}^D$ *with a noise covariance matrix* $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$ *is equivalent to the optimal attack on the monovariate traces* $\tilde{\mathbf{T}}$ *with a scalar noise of variance* $\tilde{\boldsymbol{\sigma}} \in \mathbb{R}$. *These monovariate traces* $\tilde{\mathbf{T}} \in \mathbb{R}$ *are obtained from a linear combination of the time samples of* $\mathbf{T}$*:*

$$\tilde{\mathbf{T}} = \frac{\alpha^T \mathbf{\Sigma}^{-1} \mathbf{T}}{\alpha^T \mathbf{\Sigma}^{-1} \alpha} \qquad \textit{with} \qquad \tilde{\boldsymbol{\sigma}} = \left(\alpha^T \mathbf{\Sigma}^{-1} \alpha\right)^{1/2}.$$

where α *corresponds to a* D*-dimensional vector representing coefficients weighting the leakage function for all time samples.*

Proof. see [BGH+15, Proof of Theorem 1]. □

Hence, Bruneau *et al.*'s work shows that the optimal attack includes a *linear dimensionality reduction without loss of information*. Furthermore, they also demonstrated that asymptotically, LDA computes exactly this optimal dimensionality reduction.

Nevertheless, these results are not generic since they hold only when there exists a linear relationship between the restricted leakage model Ψ (*i.e.* weighting of a leakage function) defined in [BGH+15] and the leakage function l (*e.g.* Hamming Weight) *i.e.* $\alpha \in \mathbb{R}^D$. Indeed, when considering a more complex/generic leakage model such as a weighted sum of bits interactions of the sensitive variable for a given time sample (weighted sum of Γ for a given time sample) *i.e.* $\alpha \in \mathbb{R}^{D \times 2^S}$, where S is the number of monomials, like in [SLP05] or Equation 2, these results are no longer applicable.

Another limitation concerns the method for achieving this optimal dimensionality reduction, namely LDA. Indeed, the main drawback of LDA is that it is subject to the *Small Sample Size* (SSS) problem [Fuk90] sometimes also referred as *High-Dimension Low Sample Size* (HDLSS) problem [BL04, MTA07], rendering it inapplicable. This problem arises whenever the number of traces is smaller than the number of samples, which can frequently occur in SCA [CDP15]. Consequently, due to this limitation, LDA is often set aside in practice.

In the following section, we present a new dimensionality reduction algorithm based on autoencoders (AE) that addresses the aforementioned limitations and allows to still perform an optimal attack when considering more complex leakage models.

3 Autoencoder based optimal dimensionality reduction

This section focuses on explainability aspect of our proposed neural networks by presenting an autoencoder architecture that integrates an explicit optimal dimensionality reduction. First, we extend in Section 3.1 the results obtained in Theorem 1 [BGH+15]. Then, we introduce the type of autoencoder used and the related architecture in Section 3.2 and Section 3.3.

3.1 Extension of the state-of-the-art optimal dimensionality reduction

As highlighted in Section 2.4, the results presented in [BGH+15] only hold when there exists a linear relationship between the leakage function l and the restricted leakage model Ψ. In this section, we aim at extending these results and show the optimal dimensionality reduction implicitly done during the optimal attack when considering a more generic leakage model, *i.e.* a weighted sum of bits' interactions of the sensitive value (see Equation 2). In the rest of the paper, we only consider this generic leakage model. As defined in Section 2.1, we assume a D-dimensional leakage trace $\mathbf{T}$ such that $\mathbf{T} = \Psi(Y^{k^*}) + \mathbf{N}$ where the deterministic part Ψ can be defined as $\Psi : Y^{k^*} \mapsto \alpha \cdot \Gamma(Y^{k^*})$ with $\alpha \in \mathbb{R}^{D \times 2^S}$ and the

random part $\mathbf{N}$ represents a white Gaussian noise. The latter assumption is acceptable as in practice the information contained in the covariance matrix $\mathbf{\Sigma}$ is essentially located across its diagonal [BGH+15, Figure 3]. By extending the results presented in [BGH+15], we show that, when considering a more generic leakage model, performing an optimal attack on traces $\mathbf{T}$, as defined in Definition 4, is equivalent to performing an optimal attack on monovariate traces $\tilde{\mathbf{T}}$ which result from a linear combination of time samples of $\mathbf{T}$. In other words, this means that when maximizing the optimal distinguisher (Definition 3), we implicitly perform a dimensionality reduction on $\mathbf{T}$ *i.e.* the same information contained in $\mathbf{T}$ and $\tilde{\mathbf{T}}$ is used to perform the attack.

Theorem 2. ***Optimal dimensionality reduction for a generic leakage model.*** *Under the assumption of a white Gaussian noise, the optimal attack on given multivariate traces $\mathbf{T} \in \mathbb{R}^D$ for all key hypothesis $k \in \mathcal{K}$ is equivalent to the optimal attack considering the set of monovariate traces $\tilde{\mathbf{T}} \in \mathbb{R}$ which are obtained from a linear combination of the time samples of $\mathbf{T}$:*

$$\tilde{\mathbf{T}} = \sum_{i=1}^{D} \frac{\left(\mathbf{T}_i - \sum_{j=1}^{2^S} \alpha_{ij} \big(\Gamma(Y^k)\big)_j\right)^2}{\boldsymbol{\sigma}_i^{\mathbf{2}}}. \tag{3}$$

Proof. This proof is an extension of [BGH+15] results. The optimal dimensionality reduction formula derives directly from the optimal attack equation $\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P})$, defined in Definition 4 considering a more generic leakage model as defined in Equation 2. Hence, without loss of generality, let consider $\mathcal{T}$ as a set of Q D-dimensional side-channel traces and $\mathcal{P}$ as the set of associated plaintexts. Following Definition 4, one has:

$$\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P}) = \arg\max_k \sum_{q=1}^{Q} \log\Big[\mathbb{P}_{\mathbf{n}_q}\big(\mathbf{t}_q - \alpha\,\Gamma(y_q^k)\big)\Big].$$

Since $\mathbf{n}_q \sim \mathcal{N}_D(\mathbf{0^D}, \mathbf{\Sigma})$ this equality can be written as

$$\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P}) = \arg\max_k \sum_{q=1}^{Q} \log\left[\frac{1}{\sqrt{(2\pi)^D \det(\mathbf{\Sigma})}} \exp\left(-\frac{\big(\mathbf{t}_q - \alpha\,\Gamma(y_q^k)\big)^T \mathbf{\Sigma}^{-1} \big(\mathbf{t}_q - \alpha\,\Gamma(y_q^k)\big)}{2}\right)\right].$$

By discarding constants *i.e.* k-independent variables and denoting $\stackrel{c}{=}$ the equality up to an additive constant, we obtain:

$$\begin{aligned}
\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P}) &\stackrel{c}{=} \arg\min_k \sum_{q=1}^{Q} \big(\mathbf{t}_q - \alpha\,\Gamma(y_q^k)\big)^T \mathbf{\Sigma}^{-1} \big(\mathbf{t}_q - \alpha\,\Gamma(y_q^k)\big) \\
&\stackrel{c}{=} \arg\min_k \sum_{q=1}^{Q} {\mathbf{t}_q}^T \mathbf{\Sigma}^{-1} \mathbf{t}_q - \Gamma(y_q^k)^T \alpha^T \mathbf{\Sigma}^{-1} \mathbf{t}_q - {\mathbf{t}_q}^T \mathbf{\Sigma}^{-1} \alpha\,\Gamma(y_q^k) \\
&\qquad + \Gamma(y_q^k)^T \alpha^T \mathbf{\Sigma}^{-1} \alpha\,\Gamma(y_q^k).
\end{aligned}$$

Since $\mathbf{t}_q^t \mathbf{\Sigma}^{-1} \mathbf{t}_q$ is independent of k, this equality can be simplified:

$$\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P}) \stackrel{c}{=} \arg\min_k \sum_{q=1}^{Q} -2\big(\Gamma(y_q^k)^T \alpha^T \mathbf{\Sigma}^{-1} \mathbf{t}_q\big) + \Gamma(y_q^k)^T \alpha^T \mathbf{\Sigma}^{-1} \alpha\,\Gamma(y_q^k).$$

As $\mathbf{\Sigma}$ is a diagonal matrix (white Gaussian noise assumption), this equality can be written as

$$\mathcal{DR}_{opt}(\mathcal{T}, \mathcal{P}) \stackrel{c}{=} \arg\min_k \sum_{q=1}^{Q} \sum_{i=1}^{D} \frac{-2(\mathbf{t}_q)_i \sum_{j=1}^{2^S} \alpha_{ij} \left(\Gamma(y_q^k)\right)_j + \left(\sum_{j=1}^{2^S} \alpha_{ij} \left(\Gamma(y_q^k)\right)_j\right)^2}{\boldsymbol{\sigma}_i^{\mathbf{2}}}.$$

By factorizing and discarding constants, we finally obtain:

$$\mathcal{DR}_{opt}(\mathcal{T},\mathcal{P}) \overset{c}{=} \arg\min_{k} \sum_{q=1}^{Q}\sum_{i=1}^{D} \frac{\left((\mathbf{t}_q)_i - \sum_{j=1}^{2^S} \alpha_{ij} \left(\Gamma(y_q^k)\right)_j\right)^2}{\boldsymbol{\sigma_i^2}}.$$

The right term corresponds to the sum of Q monovariate traces $\tilde{\mathbf{t}}_\mathbf{q}$. These monovariate traces are linear combinations of the Q multivariate traces $\mathbf{t}_q$, as expected. This therefore completes the proof. □

To correctly design an autoencoder based on the optimal attack that performs an optimal dimensionality reduction, we need to characterize the distribution of the monovariate traces $\tilde{\mathbf{T}}$ obtained in Theorem 2 considering the correct key hypothesis k^* as we consider in this paper a profiled attack strategy.

Corollary 1. *For the correct key hypothesis i.e. $k^* \in \mathcal{K}$, the monovariate traces $\tilde{\mathbf{T}}$ are approximately normally distributed with mean $\mu = D$ and variance $\sigma^2 = 2D$.*

Proof. From Theorem 2 the monovariate traces verify $\tilde{\mathbf{T}} = \sum_{i=1}^{D} \mathbf{X}_i^2$ with $\mathbf{X}_i = \dfrac{\mathbf{T}_i - \sum_{j=1}^{2^S} \alpha_{ij} \left(\Gamma(Y^{k^*})\right)_j}{\boldsymbol{\sigma}_i}$. Since $\mathbf{T}_i \sim \mathcal{N}(\sum_{j=1}^{2^S} \alpha_{ij} \left(\Gamma(Y^{k^*})\right)_j, \boldsymbol{\sigma_i^2})$ (see Section 2.1) the $\mathbf{X}_i$'s verify $\mathbf{X}_i \sim \mathcal{N}(0,1)$ and thus $\tilde{\mathbf{T}}$ follows a chi-squared distribution with D degrees of freedom. Hence, by the central limit theorem, as D tends to ∞, $\tilde{\mathbf{T}} \sim \chi^2(D)$ is approximately normally distributed with mean $\boldsymbol{\mu} = D$ and variance $\boldsymbol{\sigma^2} = 2D$. □

Remark 1. In practice, the convergence of the chi-squared distribution to a normal distribution is considered to be achieved when $D > 100$ [BHH78, Beh18]. In the SCA context, since a trace $\mathbf{T}$ has usually $D > 100$ samples, one can thus consider that its associated monovariate trace $\tilde{\mathbf{T}}$ follows a normal distribution with mean $\boldsymbol{\mu} = D$ and variance $\boldsymbol{\sigma^2} = 2D$. In Section 5, we assess the relevance of this heuristic by considering simulated traces of various dimensions.

According to Theorem 2, maximizing the optimal distinguisher given a trace $\mathbf{T}$ is equivalent to performing an optimal attack on reduced traces $\tilde{\mathbf{T}}$. When considering a more generic leakage model under the assumption of a white Gaussian noise, the optimal dimensionality reduction induced during the maximization of the optimal distinguisher takes the form of a normalized squared Euclidean distance (or squared Mahalanobis distance) between $\mathbf{T}$ and its deterministic part Ψ when considering the correct key hypothesis $k^* \in \mathcal{K}$. It approximately follows a normal distribution $\mathcal{N}(D, 2D)$. One way to achieve this optimal dimensionality reduction is to construct an appropriate autoencoder architecture in which the aforementioned distance is explicitly computed.

In the following section, we briefly introduce autoencoders and justify why the use of an autoencoder is relevant for computing this optimal dimensionality reduction.

3.2 Conditional Variational AutoEncoder (cVAE)

Preliminary notions on autoencoders. Autoencoders [RHW86] are a particular type of neural network commonly used in unsupervised learning for dimensionality reduction, denoising or data generation. An autoencoder consists of an input layer, an output layer and additionally one or several deep layers. It aims at learning two non-trivial parametric functions $E : \mathcal{T} \to \mathcal{Z}$ and $D : \mathcal{Z} \to \mathcal{T}$ using two different neural networks: an *encoder* and a *decoder* which respectively approximates E and D such that given an input $\mathbf{T}$, we have $D\big(E(\mathbf{T})\big) = \mathbf{T}'$, in such a way that the output $\mathbf{T}'$ is similar, following an Euclidean distance, to the input [GBC16, BKG20]. The *encoder*'s role is to learn a lower-dimensional representation (*i.e.* dimensionality reduction) of the input data, the so-called *latent space*

or *latent representation*. While, the *decoder*'s role is to reconstruct the output $\mathbf{T}'$ from the *latent space*. Thus the aim of an autoencoder is to train its *encoder* and *decoder* jointly in order to minimize the Euclidean distance between $\mathbf{T}$ and $\mathbf{T}'$. To achieve this, it is essential that during training the *encoder* is able to capture the most important features of the input, so as to provide the *decoder* with the most relevant information during its reconstruction phase.

In 2014, Kingma *et al.* [KW14] introduce a new type of autoencoder based on a generative approach called variational autoencoder (VAE). A variational autoencoder aims at explicitly computing an approximation of the unknown conditional probability distribution $\mathbb{P}(\mathbf{T} \mid \mathbf{Z})$, thereby making VAEs more explicable [Goo17]. Like classical autoencoders, VAEs consist of two parts *i.e.* an *encoder* and a *decoder*. However the major difference between classical autoencoders and VAEs lies in the computation of the *latent space* which models the unknown posterior probability distribution $\mathbb{P}(\mathbf{Z} \mid \mathbf{T})$. This modeling process is conducted by the encoder such that it is generally assumed that $\mathbb{P}(\mathbf{Z} \mid \mathbf{T})$ follows a Gaussian distribution with estimated parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$. From this *latent space*, a latent variable $\mathbf{z} \in \mathcal{Z}$ is randomly sampled such that $\mathbf{z} \sim \mathbb{P}(\mathbf{Z} \mid \mathbf{T})$ *i.e.* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then, given $\mathbf{z}$, the *decoder* generates an output $\mathbf{T}'$ such that $\mathbf{T}$ and $\mathbf{T}'$ come from the same probability distribution and thus are similar. Conditional VAE (cVAE) [SLY15] was developed as an extension of VAE to model *latent variables* and output data conditioned on auxiliary variables. By introducing these auxiliary variables into the *encoder* and *decoder*, it becomes possible to insert additional information into the *latent space* and provide more structure to the output data [SLY15, Doe21].

cVAE in SCA. In the SCA context, cVAEs were introduced by Zaid *et al.* as the first DLSCA using generative models [ZBC+23]. Indeed, they establish a link between cVAEs and classical profiled SCA by deriving an explainable cVAE architecture from theoretical results provided by stochastic attacks [SLP05]. This particular cVAE, which aims at retrieving the real unknown leakage model, allows thus to carry out interpretable and efficient DLSCA by reducing hyperparameters selection (interpretable attack) and mitigating, thanks to the loss, the impact of uninformative samples on the attack performance. Nevertheless, no investigation was conducted in [ZBC+23] about the dimensionality reduction aspect offered by cVAEs to perform the proposed attack.

This is the main objective of our work to extend those of Zaid *et al.* [ZBC+23] and Bruneau *et al.* [BGH+15] by addressing the limitations identified in each of these works so as to provide a cVAE that converges towards the optimal statistical model regardless of the leakage model considered. We opted for a cVAE to perform the optimal attack, as this type of autoencoder meets all the requirements we identified in Section 2.4 and can be considered as a good alternative to practical limitations, *e.g.* SSS or HDLSS problem [JSLH22].

Indeed, numerous studies have highlighted the surprising ability of neural networks to generalize well when dealing with high-dimensional data and thus to overcome the curse of dimensionality [OGRW11, OWR+13]. Several hypotheses have been put forward to explain this phenomenon [GK22]. Among them, the best known is the *manifold assumption*. This assumption relies on a well-known mathematical concept named the *concentration of measure* phenomenon [Led01, Mat02]. It states that a lipschitz function that depends on many independent random variables is essentially constant [Tal96, Don00]. In other words, this means that a high-dimensional data space lies on or near low-dimensional submanifold of the original space because of the redundancy between variables. Furthermore, there exists a homeomorphism to switch between these two spaces. The dimension of the low-dimensional submanifold represents thus the number of implicit parameters that are sufficient to characterize the dataset. It is called *intrinsic dimension* of the dataset. This dimension plays a key role in the performance of algorithms relying on

high dimensional input data [Ver04, BWS16]. In this regard, it has been shown that neural networks are indeed capable of reducing a high-dimensional problem to an underlying low-dimensional problem by learning local coordinate transformations of the high-dimensional data [ALMZ19]. Regarding autoencoders, recent studies confirm the ability of neural networks to overcome HDLSS problem by showing empirically that VAEs achieve better results than more conventional methods such as PCA [MFHM19, MF19]. Consequently, autoencoders (and especially VAEs) appear to be a sound alternative to LDA or other classical algorithms like PCA.

Another argument for choosing VAEs is that generative models such as VAEs or cVAEs can be more relevant to consider as dimensionality reduction techniques than discriminative models *i.e.* classical autoencoders, because they provide more interpretable models [Jeb03, ZBC+23]. Thus, by using VAEs or cVAEs to perform a dimensionality reduction, the process remains just as interpretable as classical algorithms such as PCA or LDA.

In the next section, we propose a suitable and justified generative model called cVAE-OSM that is based on cVAE and converges towards the Optimal Statistical Model regardless of the leakage model considered (see Theorem 2).

3.3 cVAE-OSM architecture

In this section, we present cVAE-OSM architecture *i.e.* an explainable cVAE architecture that converges towards the Optimal Statistical Model.

Overview of the contribution. As stated in [ZBC+23], given $\mathbf{T} \in \mathbb{R}^D$ and $Y^{k^*} \in \mathbb{F}_2^S$, the goal of a cVAE in the SCA context is to compute a θ-parametric conditional distribution $\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \theta)$ which is an approximation of the true conditional distribution $\mathbb{P}(\mathbf{T} \,|\, Y^{k^*})$. As cVAEs are *latent variable models* [Bis98], the computation of this parametric conditional distribution is done by marginalizing over the latent representation $\mathbf{Z}$:

$$\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \theta) = \int_{\mathbf{z} \in \mathcal{Z}} \mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \mathbf{z}, \theta)\, \mathbb{P}(\mathbf{z} \,|\, Y^{k^*}, \theta)\, d\mathbf{z}. \tag{4}$$

Due to its intractability, Equation 4 cannot be directly computed and is instead approximated via a *Monte-Carlo* method. To do so the encoder, whose purpose is to characterize latent spaces that are representative of input traces, estimates $\mathbb{Q}(\mathbf{Z} \,|\, \mathbf{T}, Y^{k^*}, \phi)$, an approximation of the intractable conditional probability distribution $\mathbb{P}(\mathbf{Z} \,|\, \mathbf{T}, Y^{k^*}, \theta)$. Finally, given $\mathbf{Z}$, the decoder computes the conditional probability distribution $\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \mathbf{z}, \theta)$ such that the new generated trace $\mathbf{T}'$ is drawn from the same distribution as $\mathbf{T}$.

Given a trace $\mathbf{T}$ as input, our proposed cVAE aims at (optimally) reducing $\mathbf{T}$ to its corresponding monovariate trace $\tilde{\mathbf{T}}$. This reduction is done such that the multivariate trace output $\mathbf{T}'$, which is reconstructed from $\tilde{\mathbf{T}}$, will be similar to $\mathbf{T}$ *i.e.* they are both drawn from the same distribution. In order to explicitly characterize $\tilde{\mathbf{T}}$ in the latent space, we propose the following cVAE depicted in Figure 1. Its architecture can be decomposed in three parts which are detailed below: *encoder*, *reparametrization trick* and *decoder*. Indeed, given a raw trace $\mathbf{T} \in \mathbb{R}^D$, the goal of the encoder is to optimally reduce $\mathbf{T}$ to a monovariate trace $\tilde{\mathbf{T}} \in \mathbb{R}$ such that the latent space $\mathcal{Z}$ is characterized by the parameters μ_ϕ and σ_ϕ^2 of an approximation of the distribution followed by $\tilde{\mathbf{T}}$ namely a Gaussian distribution of parameters $\mu = D$ and $\sigma^2 = 2D$ (see Corollary 1). Once the parameters are estimated and in order to reconstruct $\mathbf{T}'$, we sample, by using the reparametrization trick, D points from the estimated distribution of the latent space such that these points characterize the noise induced in traces. Then, the decoder generates $\mathbf{T}' \in \mathbb{R}^D$ such that $\mathbf{T}'$ follows the same distribution as $\mathbf{T}$.

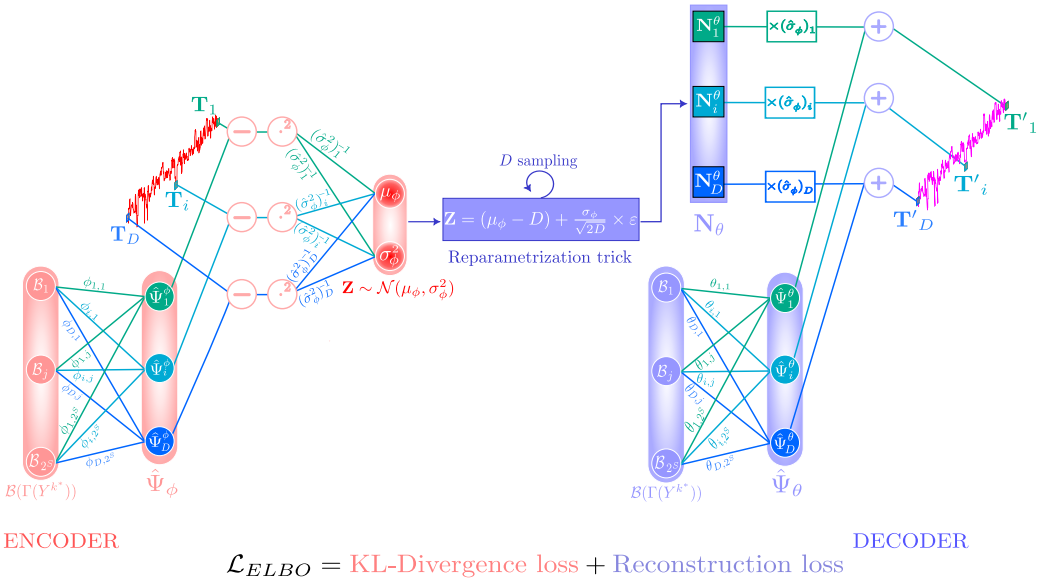


Figure 1: Proposed cVAE architecture.

Encoder. Let $\mathbf{T} \in \mathbb{R}^D$ be a D-dimensional trace such that $\mathbf{T} = \Psi(Y^{k^*}) + \mathbf{N}$ as defined in Section 2.1. The encoder $E_\phi : \mathbb{R}^D \times \mathbb{F}_2^S \to \mathbb{R} \times \mathbb{R}$ aims, given the multivariate trace $\mathbf{T}$, at approximating parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$ of the distribution followed by the monovariate trace $\tilde{\mathbf{T}}$. In this regard, we construct an encoder that explicitly computes the optimal dimensionality reduction (see Equation 3) by using linear layers. The encoder consists first of a linear layer whose purpose is to correctly estimate the deterministic part of the multivariate trace $\mathbf{T}$ *i.e.* its objective is to model a $\hat{\Psi}_\phi$ function that correctly estimates for each time sample each α_{ij} (see Equation 2). Since Ψ corresponds at each sample to a weighted sum of monomials, we design a linear network of a single fully connected layer of D neurons. This linear layer takes as input the decomposition of the sensitive value into 2^S monomials *i.e.* $\Gamma(Y^{k^*})$. Let us consider the ith $(i \in [1, D])$ neuron, by using the aforementioned architecture, its output is thus:

$$\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i = \varrho\Big(\sum_{j=1}^{2^S} \phi_{ij} \cdot \left(\Gamma(Y^{k^*})\right)_j\Big), \tag{5}$$

where ϕ_{ij} are trainable parameters that represent monomials contribution and ϱ corresponds to the activation function. In our case, to fit the leakage model formula defined in Equation 2, we consider the identity function. As the weights ϕ_{ij} converge correctly towards α_{ij} during the cVAE training (see Section 4.1.2 for weights convergence), we obtain an exact characterization of the deterministic part of $\mathbf{T}$.

Once, the characterization of the deterministic part is done by the first linear layer, we perform the optimal dimensionality reduction by using a second linear layer. To do so, for each sample, we subtract it to the real sample *i.e.* sample from the trace $\mathbf{T}$, square the result and use it as inputs for our second linear layer which consists of a single fully connected linear layer composed of two neurons. The purpose of this second neural network is thus to estimate $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}^2_\phi$, by performing the optimal dimensionality reduction as defined in Equation 3, through the use of the ELBO loss (see Section 4.1). Its output is

thus:

$$\begin{cases} \boldsymbol{\mu}_\phi & = \mathbb{E}\left[\varrho\left(\sum_{i=1}^{D} \frac{1}{(\hat{\sigma}_\phi^2)_i} \cdot \left(\mathbf{T}_i - \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\right)^2\right)\right] \\ \boldsymbol{\sigma}_\phi^{\mathbf{2}} & = \mathbb{V}\left[\varrho\left(\sum_{i=1}^{D} \frac{1}{(\hat{\sigma}_\phi^2)_i} \cdot \left(\mathbf{T}_i - \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\right)^2\right)\right] \end{cases} \tag{6}$$

where ϱ is the activation function and corresponds to the identity function to fit the optimal dimensionality reduction defined in Equation 3 and $(\hat{\boldsymbol{\sigma}}_\phi^{\mathbf{2}})_i$ are trainable parameters that represent the variance at each sample. As the weights $\frac{1}{(\hat{\boldsymbol{\sigma}}_\phi^{\mathbf{2}})_i}$ converge correctly towards $\frac{1}{\boldsymbol{\sigma}^{\mathbf{2}}}$ during the cVAE training (see Section 4.1.2 for weights convergence), the latent space is constructed so that it exactly characterizes the optimal dimensionality reduction $\tilde{\mathbf{T}}$ given a former trace $\mathbf{T}$. The encoder E_ϕ is therefore trained to return the ϕ-parametric variables $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}_\phi^{\mathbf{2}}$ of the distribution $\mathbb{Q}(\mathbf{Z} \,|\, \mathbf{T}, Y^{k^*}, \phi)$ which converge towards parameters $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\sigma}_\theta^{\mathbf{2}}$ of the intractable distribution $\mathbb{P}(\mathbf{Z} \,|\, \mathbf{T}, Y^{k^*}, \theta)$ that characterizes the latent variables $\mathbf{Z}$. According to the Corollary 1 and since $\mathbf{Z}$ describes exactly $\tilde{\mathbf{T}}$, this intractable distribution takes the form of a Gaussian distribution $\mathcal{N}(D, 2D)$.

Reparametrization trick [KW14]. Once traces are optimally reduced to a single sample, we need to reconstruct the traces $\mathbf{T}'$ from the latent variables $\mathbf{Z} \in \mathcal{Z}$ to ensure that the dimensionality reduction $\tilde{\mathbf{T}}$ done by the encoder is indeed the optimal one. Following the construction of the encoder, the latent space can be defined as the optimal dimensionality reduction space related to a trace $\mathbf{T}$. In [KW14], Kingma *et al.* identify a limitation for the estimation of the parameters which characterize the latent space. Indeed, the randomness of latent variables $\mathbf{Z}$ implies that they are non-differentiable which induces that the backpropagation cannot be performed. To circumvent this problem, Kingma *et al.* propose as solution the *reparametrization trick*. It consists of rewriting the sampling of $\mathbf{Z}$ in a such way that the derivative can still be computed with respect to $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}_\phi^{\mathbf{2}}$. For this, they first introduce a random variable $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1)$ and then compute $\mathbf{Z}$:

$$\mathbf{Z} = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \times \boldsymbol{\varepsilon} \quad \sim \mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi^{\mathbf{2}}).$$

However, such trick must be modified to being consistent with our proposal. Indeed, starting from the fact that the latent variables estimated by the encoder characterize the optimal dimensionality reduction of $\mathbf{T}$, the idea is to generate, a D-dimensional noise vector $\mathbf{N}_\theta \sim \mathcal{N}_D(\mathbf{0^D}, \mathbf{I_D})$ by sampling iteratively D samples from $\mathcal{Z}$ in such a way that each sample follows $\mathcal{N}(0, 1)$. To do so, we modify the reparametrization trick used in [KW14, ZBC+23] such that a new latent variable $\mathbf{Z}$ is now sampled from $\mathcal{N}(0, 1)$ instead of $\mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi^{\mathbf{2}})$ which converge, in our case, towards $\mathcal{N}(D, 2D)$. Therefore, we rewrite the reparametrization trick h as follows to satisfy $\mathbf{Z} \sim \mathcal{N}(0, 1)$:

$$\begin{array}{rccc} h & : & \mathbb{R} \times \mathbb{R} & \to & \mathbb{R}^D \\ & & (\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi) & \mapsto & \mathbf{Z} = (\boldsymbol{\mu}_\phi - D) + \frac{\boldsymbol{\sigma}_\phi}{\sqrt{2D}} \times \boldsymbol{\varepsilon}. \end{array}$$

In the rest of the paper, we refer to this specific reparametrization trick as the function h.

Decoder. The decoder $D_\theta : \mathbb{R}^D \times \mathbb{F}_2^S \to \mathbb{R}^D$ aims at maximizing the conditional probability distribution $\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \mathbf{z}, \theta)$. This maximization is done by reconstructing the traces $\mathbf{T}' \sim \mathcal{N}_D(\boldsymbol{\mu_\theta}, \boldsymbol{\Sigma_\theta})$ under the parameters θ such that $\mathbf{T}$ and $\mathbf{T}'$ must follow the same distribution namely $\mathcal{N}_D(\Psi(Y^{k^*}), \boldsymbol{\Sigma})$. Once $\mathbf{N}_\theta \sim \mathcal{N}_D(\mathbf{0^D}, \mathbf{I_D})$ is sampled, each sample $(\mathbf{N}_\theta)_i$ is multiplied by its corresponding standard deviation $(\hat{\boldsymbol{\sigma}}_\theta)_i$. Note that this standard

deviation $(\hat{\boldsymbol{\sigma}}_\theta)_i$ is the same as the one estimated by the encoder $(\hat{\boldsymbol{\sigma}}_\phi)_i$. We therefore obtain $\mathbf{N}_\theta \sim \mathcal{N}_D(\mathbf{0^D}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}$ defines a diagonal covariance matrix.

Then, the decoder models a linear layer, as for the encoder, to characterize the deterministic part $\hat{\Psi}_\theta(Y^{k^*})$ so that the reconstructed trace $\mathbf{T}'$ verifies

$$\mathbf{T}' = \hat{\Psi}_\theta(Y^{k^*}) + \mathbf{N}_\theta.$$

This linear layer is composed of a fully connected layer of D neurons and takes as input the decomposition of the sensitive value Y^{k^*} into 2^S monomials, *i.e.* $\Gamma(Y^{k^*})$, and outputs for the ith $(i \in [1, D])$ neuron:

$$\big(\hat{\Psi}_\theta(Y^{k^*})\big)_i = \varrho\Big(\sum_{j=1}^{2^S} \theta_{ij} \cdot \big(\Gamma(Y^{k^*})\big)_j\Big), \tag{7}$$

where ϱ corresponds to the identity function and θ_{ij} are trainable parameters that represent monomials contribution. Finally, we sum $\hat{\Psi}_\theta(Y^{k^*})$ and $\mathbf{N}_\theta$ in order to reconstruct $\mathbf{T}'$. As the weights θ_{ij} converge correctly towards α_{ij} during cVAE training (see Section 4.1.2 for weights convergence), the deterministic part of $\mathbf{T}$ is exactly characterized. Hence, this ensures that $\mathbf{T}' \sim \mathcal{N}_D(\Psi(Y^{k^*}), \boldsymbol{\Sigma})$.

Further notations related to cVAE-OSM. In the rest of this paper, we denote as $\mathsf{cVAE}_{\phi,\theta}$ the function resulting from cVAE-OSM architecture *i.e.* given a trace $\mathbf{T}$ and a sensitive value Y^{k^*}, $\mathsf{cVAE}_{\phi,\theta}(\mathbf{T}, Y^{k^*}) = D_\theta\big(h\big(E_\phi(\mathbf{T}, Y^{k^*})\big), Y^{k^*}\big) = \mathbf{T}'$. Moreover, to ease the reading of the paper, especially when it comes to weights convergence, we refer to $\{\phi_{ij}\}$ and $\{\theta_{ij}\}$ estimations as $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ estimations respectively for $i \in [1, D]$ and $j \in [1, 2^S]$. Similarly, through misuse of language, we refer to $\{\alpha_{ij}\}$ estimation as Ψ estimation for $i \in [1, D]$ and $j \in [1, 2^S]$. Finally, we distinguish in the rest of the paper $\mathbf{Z}$ and $\tilde{\mathbf{T}}$ as they refer to similar but slightly different notions. Indeed, $\mathbf{Z}$ characterizes random variable which is sampled from estimated distribution followed by the optimal dimensionality reduction of $\mathbf{T}$. While $\tilde{\mathbf{T}}$ corresponds to the exact computation of dimensionality reduction of $\mathbf{T}$. It can be computed using weights estimated by cVAE-OSM namely $\{\phi_{ij}\}$ or $\{\theta_{ij}\}$ for coefficients of the deterministic part and $\boldsymbol{\Sigma}_\phi$ for variance.

Neural network complexity. As we presented throughout this section, particularly in the last paragraphs, cVAE-OSM architecture is explicable and deterministic. This offers a real advantage over discriminative models. Firstly, because the lack of explainability and interpretability of the latter leads both attackers and evaluators to use them as black boxes. Secondly, the use of discriminative models requires an attacker/evaluator to manage a large number of hyperparameters related to architecture design. This can therefore be problematic, especially if the attacker/evaluator has a finite time frame to extract the secret information.

Moreover, as cVAE-OSM architecture is deterministic, we can provide theoretical neural network complexity bounds. In fact, given D the dimension of traces, S the highest achievable degree of bits' interactions and $d_{\max} \in [0, S]$ the maximal degree of bits' interactions considered *i.e.* the maximal degree of monomials representing bits' interactions considered, E_ϕ and D_θ require $D\sum_{i=0}^{d_{\max}} \binom{S}{i}$ weights each to estimate $\hat{\Psi}_\phi$ (see Equation 5) and $\hat{\Psi}_\theta$ (see Equation 7) respectively. Additionally, to compute the mean $\boldsymbol{\mu}_\phi$ and variance $\boldsymbol{\sigma}^2_\phi$ of monovariate traces $\tilde{\mathbf{T}}$, E_ϕ needs D weights for each parameter. Indeed, since these weights are related to the computation of the optimal dimensionality reduction (see Equation 6), there is no need to include the bias in the weights. Hence, the total number of weights required to construct our model amounts to $2D(\sum_{i=0}^{d_{\max}} \binom{S}{i} + 1)$. We can again reduce the complexity by remarking that the same D weights are used in

E_ϕ to estimate $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}^2_\phi$, as we can see in Equation 6. Further details are provided in Section 5.1, paragraph "*Implementation trick*".

Proposition 1. ***cVAE-OSM architecture complexity.*** *The final complexity of cVAE-OSM amounts to* $D(1 + 2\sum_{i=0}^{d_{\max}} \binom{S}{i})$ *weights.*

Compared to generative models proposed in the state-of-the-art, we observe a significant gain in complexity. Indeed, generative models have an $\mathcal{O}(D^2)$ complexity [ZBC+23]. By performing a dimensionality reduction, this initial $\mathcal{O}(D^2)$ complexity drops down now to an $\mathcal{O}(D)$ complexity. Furthermore, as this dimensionality reduction is performed without any loss of information, this gain in complexity doesn't affect cVAE-OSM performance, meaning that our model provides an attack performance level equivalent to state-of-the-art generative models.

Hence, through this section, we present an explicable model with a simplified architecture preventing unnecessary weights estimation that performs a dimensionality reduction without loss of information such as LDA while avoiding its practical limitations like HDLSS/SSS problem. Moreover, our model architecture is designed to perform optimal SCA (see Definition 4). In the following section, we focus on cVAE-OSM learning strategy, which ensures our model to properly converge towards the optimal statistical model.

4 Learning and key recovery strategy

This section is dedicated to the interpretability aspect of cVAE-OSM. First, we prove in Section 4.1 that cVAE-OSM is an optimal statistical model for SCA. To do so, we first ensure that our model converges towards optimal statistical model by designing a customized loss in Section 4.1.1. Then, we provide in Section 4.1.2 proofs of convergence for all weights included in cVAE-OSM, which guarantees that our model is an optimal statistical model. Finally, we present in Section 4.2 an end-to-end profiled attack strategy using cVAE-OSM.

4.1 Optimization and weights convergence

4.1.1 Optimization

In this section, we aim at describing the optimization process employed to properly train cVAE-OSM. As explained in Section 3.3, the goal of a cVAE-OSM is to correctly approximate the true unknown conditional distribution $\mathbb{P}(\mathbf{T}|Y^{k^*})$ by computing the θ-parametric distribution $\mathbb{P}(\mathbf{T}|Y^{k^*}, \theta)$. To estimate it, the cVAE-OSM has to optimize the set of parameters $\Theta = \{\phi, \theta\}$ such that it maximizes the conditional marginal log-likelihood $\log\big(\mathbb{P}(\mathbf{T}|Y^{k^*}, \theta)\big)$.

Theorem 3. *Conditional marginal log-likelihood [ZBC+23]. For all choice of encoder* E_ϕ *and its associated trainable parameters* ϕ*, the conditional marginal log-likelihood verifies:*

$$\begin{aligned}\log\big(\mathbb{P}(\mathbf{T}|Y^{k^*}, \theta)\big) = \mathbb{E}_{\mathbf{z}\sim\ E_\phi}\Big[\log\big(\mathbb{P}(\mathbf{T}, \mathbf{z}\,|\,Y^{k^*}, \theta)\big) - \log\big(\mathbb{Q}(\mathbf{z}\,|\,\mathbf{T}, Y^{k^*}, \phi)\big)\Big] \\ + \mathcal{D}_{KL}\Big(\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T}, Y^{k^*}, \phi)\;||\;\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T}, Y^{k^*}, \theta)\Big).\end{aligned} \tag{8}$$

where $\mathbb{E}_{\mathbf{z}\sim E_\phi}$ *denotes the expectation over the distribution of the random variable* $\mathbf{z}$*, which is sampled from* E_ϕ*.*

Proof. see [ZBC+23, Proof of Theorem 1]. □

However maximizing such a quantity is not possible due to the intractability of $\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T}, Y^{k^*}, \theta)$. Consequently, in [KW14], Kingma *et al.* propose a variational lower

bound on the marginal likelihood to circumvent this issue. This lower bound is then generalized by Sohn *et al.* [SLY15].

Theorem 4. *Variational lower bound on the conditional marginal likelihood [SLY15]. For all choice of encoder E_ϕ and its associated trainable parameters ϕ, the variational lower bound on the conditional marginal likelihood* $\log\big(\mathbb{P}(\mathbf{T}|Y^{k^*},\theta)\big)$ *is given as:*

$$\log\big(\mathbb{P}(\mathbf{T}|Y^{k^*},\theta)\big) \geq \mathbb{E}_{\mathbf{z}\sim\ E_\phi}\big[\log\big(\mathbb{P}(\mathbf{T}\,|\,Y^{k^*},\mathbf{z},\theta)\big)\big] - \mathcal{D}_{KL}\big(\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)\;||\;\mathbb{P}(\mathbf{Z})\big). \quad (9)$$

Proof. see [SLY15, Section 4] and [KW14, Section 2.2]. □

The equality between Equation 8 and Equation 9 holds if and only if the distribution $\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)$ computed by the encoder E_ϕ (see Section 3.3) corresponds perfectly to the intractable conditional distribution $\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\theta)$. When this condition is met, the latent space $\mathcal{Z}$ characterizes exactly the optimal reduction dimensionality of $\mathbf{T}$. Thus, based on this lower bound, we can now define the empirical risk of cVAE-OSM.

Definition 5. Empirical Risk when considering the ELBO loss [KW14]. Let $\mathcal{T} = \{\mathbf{t}_1,\ldots,\mathbf{t}_Q\}$ be a set of Q side channel traces and $\mathcal{Y} = \{y_1^{k^*},\ldots,y_Q^{k^*}\}$ the set of associated sensitive values. The *empirical risk* optimizing $\mathsf{cVAE}_{\phi,\theta}$ is defined as

$$\begin{aligned}\hat{\mathcal{R}}(\mathsf{cVAE}_{\phi,\theta}) &= \frac{1}{Q}\sum_{q=1}^{Q}\mathcal{L}_{ELBO}\big(\mathbf{t}_q,y_q^{k^*};\phi,\theta\big)\\ &= \frac{1}{Q}\sum_{q=1}^{Q}\underbrace{\mathcal{D}_{KL}\big(\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{t}_q,y_q^{k^*},\phi)\;||\;\mathbb{P}(\mathbf{Z})\big)}_{\text{KL-Divergence loss}} - \underbrace{\mathbb{E}_{\mathbf{z}\sim E_\phi}\big[\log\big(\mathbb{P}(\mathbf{t}_q\,|\,y_q^{k^*},\mathbf{z},\theta)\big)\big]}_{\text{Reconstruction loss}}\end{aligned}$$

where, given $(\mathbf{t}_q,y_q^{k^*})$, $\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{t}_q,y_q^{k^*},\phi)$ and $\mathbb{P}(\mathbf{t}_q\,|\,y_q^{k^*},\mathbf{z},\theta)$ are respectively computed by the encoder E_ϕ and decoder D_θ.

The ELBO loss aims to optimize Θ so that cVAE-OSM can approximate the true unknown conditional probability distribution $\mathbb{P}(\mathbf{T}\,|\,Y^{k^*})$. As we can see in Definition 5, this loss can be decomposed into two "sub-losses": the *KL-divergence loss* and the *reconstruction loss*.

KL-divergence loss. This term plays the role of a regularizer in the ELBO loss because its purpose is to counterbalance the reconstruction loss in a such way as to avoid a trivial reconstruction. The KL-divergence loss which impacts only the encoder E_ϕ aims at optimizing ϕ in order to minimize the KL-divergence between $\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)$ and $\mathbb{P}(\mathbf{Z})$. Indeed, as explained in Section 3.3, the goal of E_ϕ is to correctly approximate the intractable conditional distribution $\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\theta)$ which is involved in the expression of the conditional marginal log-likelihood $\log\big(\mathbb{P}(\mathbf{T}\,|\,Y^{k^*},\theta)\big)$ (see Theorem 3). Hence, the maximization of $\log\big(\mathbb{P}(\mathbf{T}\,|\,Y^{k^*},\theta)\big)$ highly depends on the good matching between $\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)$ and $\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\theta)$. However as explained earlier, due to the intractability of $\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\theta)$, we cannot directly maximize the conditional marginal log-likelihood by computing $\mathcal{D}_{KL}\big(\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)\;||\;\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\theta)\big)$. Fortunately, to circumvent this issue, it has been shown that an equality is obtained between the conditional marginal log-likelihood and the variational lower bound (see Theorem 4) when $\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)$ corresponds to $\mathbb{P}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\theta)$ [KW14, SLY15]. Consequently, as we can see in Equation 9, to reach this equality and thus maximize the conditional marginal log-likelihood, we can instead compute $\mathcal{D}_{KL}\big(\mathbb{Q}(\mathbf{Z}\,|\,\mathbf{T},Y^{k^*},\phi)\;||\;\mathbb{P}(\mathbf{Z})\big)$ which appears in the variational lower bound expression. Since the latent space $\mathcal{Z}$ in cVAE-OSM characterizes the optimal dimensionality reduction $\tilde{\mathbf{T}}$ of traces $\mathbf{T}$, according to Corollary 1, $\mathbf{Z}$ follows a normal

distribution of parameters $\boldsymbol{\mu} = D$ and $\boldsymbol{\sigma^2} = 2D$. Therefore, during cVAE-OSM training, to maximize $\log\big(\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \theta)\big)$, the KL-divergence loss has to force the ϕ-parametric distribution $\mathbb{Q}(\mathbf{Z} \,|\, \mathbf{T}, Y^{k^*}, \phi)$ such that it fits a Gaussian distribution $\mathcal{N}(D, 2D)$. Taking these elements into account, it is now possible to define the KL-divergence loss of cVAE-OSM.

Proposition 2. ***KL-divergence loss of cVAE-OSM.*** *Let* $\mathbf{T}$ *be a* D*-dimensional side channel trace and* Y^{k^*} *its associated sensitive value, the KL-divergence loss of cVAE-OSM verifies:*

$$D_{KL}\big(\mathbb{Q}(\mathbf{Z}|\mathbf{T}, Y^{k^*}, \phi) \,||\, \mathbb{P}(\mathbf{Z})\big) = \frac{1}{2}\left(\log\left(\frac{2D}{\boldsymbol{\sigma}_\phi^{\mathbf{2}}}\right) + \frac{\big(\boldsymbol{\mu}_\phi - D\big)^2 + \boldsymbol{\sigma}_\phi^{\mathbf{2}}}{2D} - 1 \right) \tag{10}$$

where $\boldsymbol{\mu}_\phi$ *and* $\boldsymbol{\sigma}_\phi$ *correspond to the parameters estimated by the encoder* E_ϕ*.*

Proof. The result is obtained by expanding the KL-divergence loss expression:

$$\begin{aligned}
&D_{KL}\big(\mathbb{Q}(\mathbf{Z}|\mathbf{T}, Y^{k^*}, \phi) \,||\, \mathbb{P}(\mathbf{Z})\big) = \mathbb{E}_{\mathbf{z}\sim E_\phi}\Big[\log\big(\mathbb{Q}(\mathbf{z}|\mathbf{T}, Y^{k^*}, \phi) - \log\big(\mathbb{P}(\mathbf{z})\big)\Big] \\
&= \mathbb{E}_{\mathbf{z}\sim E_\phi}\left[\log\left(\frac{1}{\boldsymbol{\sigma}_\phi\sqrt{2\pi}}\right) - \frac{1}{2}\left(\frac{\mathbf{z} - \boldsymbol{\mu}_\phi}{\boldsymbol{\sigma}_\phi}\right)^2 - \log\left(\frac{1}{\sqrt{2D}\sqrt{2\pi}}\right) + \frac{1}{2}\left(\frac{\mathbf{z} - D}{\sqrt{2D}}\right)^2 \right] \\
&= \log\left(\frac{\sqrt{2D}}{\boldsymbol{\sigma}_\phi}\right) - \frac{1}{2}\left(\frac{\mathbb{E}_{\mathbf{z}\sim E_\phi}\big[(\mathbf{z} - \boldsymbol{\mu}_\phi)^2\big]}{\boldsymbol{\sigma}_\phi^{\mathbf{2}}}\right) + \frac{1}{2}\left(\frac{\mathbb{E}_{\mathbf{z}\sim E_\phi}\big[(\mathbf{z} - D)^2\big]}{2D}\right) \\
&= \frac{1}{2}\left(\log\left(\frac{2D}{\boldsymbol{\sigma}_\phi^{\mathbf{2}}}\right) + \frac{\big(\boldsymbol{\mu}_\phi - D\big)^2 + \boldsymbol{\sigma}_\phi^{\mathbf{2}}}{2D} - 1 \right).
\end{aligned}$$

□

Reconstruction loss. While optimizing parameters ϕ through its training, the cVAE-OSM also optimizes concomitantly the parameters θ thanks to the reconstruction loss. This loss which affects the decoder D_θ aims at maximizing under the parameters θ the conditional marginal log-likelihood $\log\big(\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \mathbf{z}, \theta)\big)$ given a latent representation $\mathbf{z} \in \mathcal{Z}$. In other words, this term encourages D_θ to optimize its parameters θ in such a way as the trace $\mathbf{T}'$ which is reconstructed by D_θ given the latent variable $\mathbf{z}$ is similar to $\mathbf{T}$ *i.e.* $\mathbf{T}$ and $\mathbf{T}'$ follow the same distribution. As a reminder, since we assume that the traces are composed of a deterministic part $\Psi(Y^{k^*})$ and a white Gaussian noise, the distribution followed by the traces is then assumed to be a multivariate Gaussian where the covariance matrix is diagonal. Hence, based on this assumption, we can now compute the reconstruction loss.

Proposition 3. ***Reconstruction loss of cVAE-OSM.*** *Let* $\mathbf{T}$ *be a* D*-dimensional side channel trace and* Y^{k^*} *its associated sensitive value, the Reconstruction loss of cVAE-OSM verifies:*

$$\mathbb{E}_{\mathbf{z}\sim E_\phi}\big[-\log\big(\mathbb{P}(\mathbf{T} \,|\, Y^{k^*}, \mathbf{z}, \theta)\big)\big] = \mathbb{E}_{\mathbf{z}\sim E_\phi}\left[\sum_{i=1}^{D} \frac{1}{2}\left(\log(2\pi(\boldsymbol{\sigma}_\theta^{\mathbf{2}'})_i) + \frac{\big(\mathbf{T}_i - (\boldsymbol{\mu}_\theta')_i\big)^2}{(\boldsymbol{\sigma}_\theta^{\mathbf{2}'})_i} \right) \right] \tag{11}$$

where $(\boldsymbol{\mu}_\theta')_i$ *and* $(\boldsymbol{\sigma}_\theta^{\mathbf{2}'})_i$ *are respectively the mean and variance at the* i^{th} *sample of* $\mathbf{T}'$*, the traces reconstructed by the decoder* D_θ*.*

Proof. Since the reconstructed traces $\mathbf{T}'$ are assumed to be drawn from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_\theta'$ and diagonal covariance matrix $\boldsymbol{\Sigma}_\theta'$ the result can be deduced straightforwardly. □

The expectation term in the reconstruction loss can be approximated via a *Monte Carlo sampling* method. Nevertheless, due to computational efficiency, we only sample one latent variable $\mathbf{z}$ during training.

Now that we defined how the optimization is done during cVAE-OSM training, we propose in the following section weights convergence proofs that ensure that the proposed cVAE-OSM converges towards the optimal statistical model and thus to indeed perform an optimal reduction dimensionality.

4.1.2 Weights convergence

Using the ELBO loss, proposed in Definition 5, allows cVAE-OSM to converge to the optimal statistical model (see Theorem 2) by appropriately optimizing parameters ϕ of the encoder E_ϕ and parameters θ of the decoder D_θ. First, as a reminder, we consider a D-dimensional side channel trace $\mathbf{T}$ as a random vector that is drawn from a multivariate Gaussian distribution of parameters $\boldsymbol{\mu} = \Psi(Y^{k^*})$ and $\boldsymbol{\Sigma} = \text{diag}[\boldsymbol{\sigma}_1^2, \ldots, \boldsymbol{\sigma}_D^2]$. Let $\{l_1, \ldots, l_{np}\}$ be a set of PoIs, then, for each sample $i \in [1, D]$, $\mathbf{T}_i$ is thus defined as:

$$\mathbf{T}_i = \begin{cases} \left(\Psi(Y^{k^*})\right)_i + \mathbf{N}_i \sim \mathcal{N}\Big(\left(\Psi(Y^{k^*})\right)_i, \boldsymbol{\sigma}_i^2\Big) & \text{if } i \in \{l_1, \ldots, l_{np}\} \\ \mathbf{N}_i \sim \mathcal{N}(0, \boldsymbol{\sigma}_i^2) & \text{otherwise.} \end{cases} \tag{12}$$

Encoder. As explained in Section 4.1.1, the trainable parameters ϕ, namely $\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i$ and $(\hat{\boldsymbol{\sigma}}_\phi^2)_i$ which are involved in E_ϕ, are affected by the KL-divergence loss term (see Proposition 2). Indeed, to minimize this term, we need to fit, during the training of cVAE-OSM, the ϕ-parametric distribution $\mathbb{Q}(\mathbf{Z} \,|\, \mathbf{T}, Y^{k^*}, \phi)$ to the distribution followed by the latent variable $\mathbf{Z}$ namely the Gaussian distribution $\mathcal{N}(D, 2D)$. To achieve this *i.e.* a KL-divergence close or equal to zero, it is necessary that the encoder E_ϕ optimizes its parameters $\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i$ and $(\hat{\boldsymbol{\sigma}}_\phi^2)_i$ such that they converge respectively to $\left(\Psi(Y^{k^*})\right)_i$ and $\boldsymbol{\sigma}_i^2$. If so, the correct convergence of weights ensures that the latent variable $\mathbf{Z} \in \mathcal{Z}$ accurately characterizes the optimal dimensionality reduction of $\mathbf{T}$ *i.e.* $\tilde{\mathbf{T}}$.

Theorem 5. ***Convergence of encoder parameters.*** *Let $\mathbf{T}$ be a side channel trace and Y^{k^*} its associated sensitive value such that $\mathbf{T}$ is defined as in Equation 12 and its dimension D is sufficiently large. During its training, the cVAE-OSM optimizes encoder parameters ϕ so as to minimize the KL-divergence loss. When this optimal configuration is reached, we have for all samples $i \in [1, D]$, $\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i = \left(\Psi(Y^{k^*})\right)_i$ and $(\hat{\boldsymbol{\sigma}}_\phi^2)_i = \boldsymbol{\sigma}_i^2$.*

Proof. According to Theorem 2, a side channel trace $\mathbf{T}$ defined as in Equation 12 has an optimal dimensionality reduction $\tilde{\mathbf{T}}$ which equals to:

$$\tilde{\mathbf{T}} = \sum_{i=1}^{D} \left(\frac{\mathbf{T}_i - \sum_{j=1}^{2^S} \alpha_{ij} \left(\Gamma(Y^{k^*})\right)_j}{\boldsymbol{\sigma}_i} \right)^2.$$

Considering the fact that $\mathbf{T} \sim \mathcal{N}_D(\Psi(Y^{k^*}), \boldsymbol{\Sigma})$ we therefore have, as stated in Corollary 1, $\tilde{\mathbf{T}} \sim \mathcal{N}(D, 2D)$ for large D (heuristically $D > 100$). As explained in Section 3.3, cVAE-OSM is designed such that $\mathbf{Z} \in \mathcal{Z}$ is sampled from an estimation of $\tilde{\mathbf{T}}$ distribution. Hence, during training, encoder E_ϕ parameters force, via the KL-divergence loss, $\{\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\}_{1 \leq i \leq D}$ and $\{(\hat{\boldsymbol{\sigma}}_\phi)_i\}_{1 \leq i \leq D}$ weights to converge towards quantities that ensure $\mathbf{Z}$ follows a Gaussian distribution $\mathcal{N}(D, 2D)$ given

$$\mathbf{Z} = \sum_{i=1}^{D} \left(\frac{\mathbf{T}_i - \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i}{(\hat{\boldsymbol{\sigma}}_\phi)_i} \right)^2.$$

Now, let us define $\mathbf{V}_i = \dfrac{\mathbf{T}_i - \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i}{(\hat{\boldsymbol{\sigma}}_\phi)_i}$. As $\mathbf{Z}$ follows a Gaussian distribution $\mathcal{N}(D, 2D)$, by definition, the KL-divergence loss forces therefore implicitly $\mathbf{V}_i \sim \mathcal{N}(0,1)$ for large D (see Corollary 1).
Based on these results, we can now prove that KL-divergence loss minimization forces $\{\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\}_{1\leq i\leq D}$ and $\{(\hat{\boldsymbol{\sigma}}_\phi)_i\}_{1\leq i\leq D}$ weights to converge towards $\{\left(\Psi(Y^{k^*})\right)_i\}_{1\leq i\leq D}$ and $\{(\boldsymbol{\sigma}_\phi)_i\}_{1\leq i\leq D}$ respectively.
This proof will be divided in two cases. Each one corresponds to whether the sample considered is a PoI or not.
Case 1: $\mathbf{T}_i = \left(\Psi(Y^{k^*})\right)_i + \mathbf{N}_i$ *i.e.* $\mathbf{T}_i$ is a PoI. By replacing $\mathbf{T}_i$ by its expression in $\mathbf{V}_i$, we obtain: $\mathbf{V}_i = \frac{\left(\Psi(Y^{k^*})\right)_i + \mathbf{N}_i - \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i}{(\hat{\boldsymbol{\sigma}}_\phi)_i} \sim \mathcal{N}(0,1)$. $\mathbf{N}_i$ can now be expressed as a function of $\mathbf{V}_i$:

$$\mathbf{N}_i = (\hat{\boldsymbol{\sigma}}_\phi)_i \mathbf{V}_i - \left(\Psi(Y^{k^*})\right)_i + \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i,$$

Its probability distribution is thus:

$$\begin{aligned}
\mathbf{N}_i &\sim \mathcal{N}\Big(\mathbb{E}\big[(\hat{\boldsymbol{\sigma}}_\phi)_i \mathbf{V}_i - \left(\Psi(Y^{k^*})\right)_i + \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\big], \mathbb{V}\big[(\hat{\boldsymbol{\sigma}}_\phi)_i \mathbf{V}_i - \left(\Psi(Y^{k^*})\right)_i + \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\big]\Big) \\
&\sim \mathcal{N}\Big(\mathbb{E}\big[(\hat{\boldsymbol{\sigma}}_\phi)_i \mathbf{V}_i - \left(\Psi(Y^{k^*})\right)_i + \left(\hat{\Psi}_\phi(Y^{k^*})\right)_i\big], \mathbb{V}\big[(\hat{\boldsymbol{\sigma}}_\phi)_i \mathbf{V}_i\big]\Big) \\
&\sim \mathcal{N}\Big(\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i - \left(\Psi(Y^{k^*})\right)_i, (\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\phi)_i\Big).
\end{aligned}$$

Considering the fact that $\mathbf{N}_i \sim \mathcal{N}(0, \boldsymbol{\sigma}^{\mathbf{2}}_i)$, we therefore have $\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i = \left(\Psi(Y^{k^*})\right)_i$ and $(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\phi)_i = \boldsymbol{\sigma}^{\mathbf{2}}_i$.
Case 2: $\mathbf{T}_i = \mathbf{N}_i$ *i.e.* $\mathbf{T}_i$ is not a PoI. Following the same reasoning as for the first case, one can prove that $\left(\hat{\Psi}_\phi(Y^{k^*})\right)_i = \left(\Psi(Y^{k^*})\right)_i = 0$ and $(\hat{\boldsymbol{\sigma}^{\mathbf{2}}}_{\boldsymbol{\phi}})_i = \boldsymbol{\sigma}^{\mathbf{2}}_i$ when the time sample $\mathbf{T}_i$ is not considered as a PoI.

□

Remark 2. One can observe during SCA experiments (see Section 5) that the convergence stated in Theorem 5 can be obtained even for small D (*e.g.* $D \sim 10$).

Decoder. The trainable parameters θ *i.e.* $\left(\hat{\Psi}_\theta(Y^{k^*})\right)_i$ and $(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i$ of the decoder D_θ are affected by the reconstruction loss term (see Proposition 3) during the training of cVAE-OSM. Indeed, D_θ aims at reconstructing a trace $\mathbf{T}' \in \mathbb{R}^D$ given the optimal dimensionality reduction $\tilde{\mathbf{T}} \in \mathbb{R}$ of the trace $\mathbf{T} \in \mathbb{R}^D$. To do so, it seeks to find the parameters θ that minimize the reconstruction loss. It turns out that the configuration in which the reconstruction loss is minimized happens when D_θ optimizes the θ-parameters in such a way that $\left(\hat{\Psi}_\theta(Y^{k^*})\right)_i$ and $(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i$ correspond to $\left(\Psi(Y^{k^*})\right)_i$ and $\boldsymbol{\sigma}^{\mathbf{2}}_i$.

Theorem 6. ***Convergence of decoder parameters.*** *Let* $\mathbf{T}$ *be a side channel trace and* Y^{k^*} *its associated sensitive value such that* $\mathbf{T}$ *is defined as in Equation 12. During its training, cVAE-OSM optimizes decoder parameters* θ *in a such way that in an optimal configuration i.e. a configuration that minimizes the reconstruction loss, for all samples* $i \in [1, D]$*, we have* $\left(\hat{\Psi}_\theta(Y^{k^*})\right)_i = \left(\Psi(Y^{k^*})\right)_i$ *and* $(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i = \boldsymbol{\sigma}^{\mathbf{2}}_i$.

Proof. Let $\mathbf{T}$ be a side channel trace defined as in Equation 12. This proof will be divided in two cases. Each one corresponds to whether the sample considered is a PoI or not.
As explained previously, regarding to the decoder, its parameters θ are only affected by the reconstruction loss. Let $\mathbf{T}'$ be the trace reconstructed by the decoder. This trace is drawn from a multivariate Gaussian distribution of parameters $\boldsymbol{\mu_\theta} = \hat{\Psi}_\theta(Y^{k^*})$ and $\boldsymbol{\Sigma_\theta} = \mathrm{diag}[(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_1, \ldots, (\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_D]$.
Case 1: $\mathbf{T}_i = \left(\Psi(Y^{k^*})\right)_i + \mathbf{N}_i$ *i.e.* $\mathbf{T}_i$ is a PoI.

By replacing $\mathbf{T}_i$ in the reconstruction loss by its expression, we obtain:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim E_\phi}\left[\sum_{i=1}^{D}\frac{1}{2}\left(\log(2\pi(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i)+\frac{\left(\mathbf{T}_i-(\boldsymbol{\mu}_{\boldsymbol{\theta}})_i\right)^2}{(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i}\right)\right]\\
=&\,\mathbb{E}_{\mathbf{z}\sim E_\phi}\left[\sum_{i=1}^{D}\frac{1}{2}\left(\log(2\pi(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i)+\frac{\left(\left(\Psi(Y^{k^*})\right)_i+\mathbf{N}_i-\left(\hat{\Psi}_\theta(Y^{k^*})\right)_i\right)^2}{(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i}\right)\right].
\end{aligned}$$

As stated in [Yu20], this equation is minimized if and only if $\frac{\left(\left(\Psi(Y^{k^*})\right)_i+\mathbf{N}_i-\left(\hat{\Psi}_\theta(Y^{k^*})\right)_i\right)^2}{(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i}=1$. This means that the minimum is reached if and only if $(\boldsymbol{\mu}_{\boldsymbol{\theta}})_i=\boldsymbol{\mu}_i$ and $(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i=(\boldsymbol{\sigma}^{\mathbf{2}})_i$ where $\boldsymbol{\mu}_i$ and $(\boldsymbol{\sigma}^{\mathbf{2}})_i$ correspond respectively to the true mean and variance of traces at sample i. In this case, we therefore have

$$\begin{cases}
(\boldsymbol{\mu}_{\boldsymbol{\theta}})_i=\left(\hat{\Psi}_\theta(Y^{k^*})\right)_i &=\mathbb{E}\left[\left(\Psi(Y^{k^*})\right)_i+\mathbf{N}_i\right]=\left(\Psi(Y^{k^*})\right)_i+\mathbb{E}\left[\mathbf{N}_i\right]=\left(\Psi(Y^{k^*})\right)_i=\boldsymbol{\mu}_i\\
(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i &=\mathbb{V}\left[\left(\Psi(Y^{k^*})\right)_i+\mathbf{N}_i\right]=\mathbb{V}\left[\mathbf{N}_i\right]=\boldsymbol{\sigma}^{\mathbf{2}}_i.
\end{cases}$$

Case 2: $\mathbf{T}_i=\mathbf{N}_i$ *i.e.* $\mathbf{T}_i$ is not a PoI.
Following the same reasoning as for the first case, we prove that $(\boldsymbol{\mu}_{\boldsymbol{\theta}})_i=\left(\hat{\Psi}_\theta(y_q^{k^*})\right)_i=\mathbb{E}\left[\mathbf{N}_i\right]=0=\boldsymbol{\mu}_i$ and $(\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_\theta)_i=\mathbb{V}\left[\mathbf{N}_i\right]=\boldsymbol{\sigma}^{\mathbf{2}}_i$ when the time sample $\mathbf{T}_i$ is not considered as a PoI.

□

Now that we have proved that the $\mathsf{cVAE}_{\phi,\theta}$ function resulting from cVAE-OSM architecture proposed in Section 3.3, converges actually to the optimal statistical model, we introduce in the following section a strategy using cVAE-OSM to conduct an end-to-end profiled attack.

4.2 cVAE-OSM based profiled attack strategy

Since cVAE-OSM converges to an optimal statistical model (see Section 4.1), we conduct, to recover the correct key k^*, a profiled attack using as distinguishing rule the optimal dimensionality reduction based rule instead of the ELBO loss. Compared to [ZBC+23], the following strategy avoids one important issue of the former work which consisted in sampling latent variable in order to compute the distinguishing rule. Based on Theorem 2, given a set $\mathcal{T}$ of Q D-dimensional side channel traces and a set of associated plaintexts $\mathcal{P}$, this rule is defined as follows:

$$\mathcal{DR}_{opt}(\mathcal{T},\mathcal{P})=\arg\min_k\sum_{q=1}^{Q}\tilde{\mathbf{t}}_q(k). \tag{13}$$

where $\tilde{\mathbf{t}}_q(k)$ corresponds to monovariate trace associated with $\mathbf{t}_q$ for a given key hypothesis $k\in\mathcal{K}$ (see Theorem 2).

Indeed, by definition, using Equation 13 as a distinguishing rule, allows us to conduct the best attack possible in terms of success probability maximization since $\mathcal{DR}_{opt}$ is the best rule among all known ones to distinguish the correct key k^* [HRG14]. The attack is a two-stage process, its sketch is described below.

1. **cVAE-OSM training and PoIs selection phase**
   This first step aims at training the model on profiling set $\mathcal{T}_{prof}=\{\mathbf{t}_1,...,\mathbf{t}_N\}\in\mathbb{R}^{N\times D}$ and $\mathcal{Y}_{prof}=\{y_1,...,y_N\}$ the set of associated sensitive values in which each y_i is

computed as $y_i = f(p_i, k_i)$ with f a cryptographic function mixing p_i and k_i the associated plaintext and key. These estimations are then used to select PoIs for the key recovery phase. To do so, the attacker/evaluator must:

1.1. Choose a basis $\mathcal{B}$ to decompose sensitive values $\mathcal{Y}_{prof}$ (see Section 5.1.1 for an example of basis choice).

1.2. Decompose $\mathcal{Y}_{prof}$ in the full basis $\mathcal{B}$ *i.e.* all degrees of bits' interaction are considered. We denote this decomposition as $\mathcal{B}\big(\Gamma(\mathcal{Y}_{prof})\big)$.

1.3. Train cVAE-OSM using $\mathcal{T}_{prof}$ and $\mathcal{B}\big(\Gamma(\mathcal{Y}_{prof})\big) \in \mathbb{R}^{N\times 2^S}$ with 2^S the number of monomials representing all bits' interactions.

1.4. Once cVAE-OSM is properly trained, the attacker has at its disposal encoder (resp. decoder) weights $\hat{\Psi}_\phi \in \mathbb{R}^{D\times 2^S}$ and $\mathbf{\Sigma}_\phi$ (resp. $\hat{\Psi}_\theta \in \mathbb{R}^{D\times 2^S}$) which correspond to leakage model and variance (resp. leakage model) estimation. He can thus visualize the hypothetical leakage model thanks to $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$.

1.5. Optional: to enhance the key recovery phase, the attacker can perform, by using $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$, a leakage assessment to both automatically select relevant bit interactions (*i.e.* select the most informative parts from the $\mathcal{B}$-basis decomposition) and PoIs. PoIs selection methods are proposed in Section 5.2.1.

2. **Key recovery phase**
Once the attacker has at its disposal leakage model and variance estimations, he can for each key hypothesis $k \in \mathcal{K}$:

2.1. Compute sensitive values set $\mathcal{Y}_{att} = \{y_1^a, ..., y_Q^a\}$ where $y_i^a = f(p_i^a, k)$, given the set of known attack plaintexts set $\mathcal{P}_{att} = \{p_1^a, ..., p_Q^a\}$.

2.2. Decompose $\mathcal{Y}_{att}$ in the full basis $\mathcal{B}$ chosen in the previous phase such that $\mathcal{B}\big(\Gamma(\mathcal{Y}_{att})\big) \in \mathbb{R}^{Q\times 2^S}$.

2.3. Given the set of Q attack traces $\mathcal{T}_{att} = \{\mathbf{t}_1^a, ..., \mathbf{t}_Q^a\}$, the attacker computes for each trace its optimal dimensionality reduction

$$\tilde{\mathbf{t}}_q^a(k) = \sum_{i=1}^{D} \frac{\left((\mathbf{t}_q^a)_i - \sum_{j=1}^{2^S} \hat{\alpha}_{ij} \left(\mathcal{B}\big(\Gamma(y_q^a)\big) \right)_j \right)^2}{\hat{\boldsymbol{\sigma}}_i^{\mathbf{2}}}$$

where $\hat{\alpha}$ and $(\hat{\boldsymbol{\sigma}}_i^{\mathbf{2}})_{1\le i\le D}$ are weights estimated by cVAE-OSM during its training that characterize coefficients of leakage model and variance. It corresponds to either weights of $\hat{\Psi}_\phi$ *i.e.* $\{\phi_{ij}\}_{1\le i\le D, 1\le j\le 2^S}$ or weights of $\hat{\Psi}_\theta$ *i.e.* $\{\theta_{ij}\}_{1\le i\le D, 1\le j\le 2^S}$ for $\hat{\alpha}$ and $\mathbf{\Sigma}_\phi$ for estimated variance.

2.4. The attacker can thus compute the score $\mathcal{S}_k = \sum_{q=1}^{Q} \tilde{\mathbf{t}}_q^a(k)$ associated with the key hypothesis k.

2.5. Optional: to enhance the attack, the attacker can perform steps 2.3 and 2.4 considering the basis and PoIs automatically extracted in 1.5. By performing the attack on a subset of samples and a subset of bits' interaction, he can mitigate the impact of uninformative samples and bits' interactions that may degrade the attack.

Assuming that the profiling phase is correctly performed (*i.e.* leakage model and variance of traces are correctly estimated), by definition, $\mathcal{S}_k$ should be minimized when $k = k^*$. Therefore, following Equation 13, the most likely key candidate is $\hat{k} = \arg\min_k \mathcal{S}_k$.

Since Equation 13 derives directly from optimal distinguisher [HRG14], the attacker is ensured to perform an optimal attack provided that the profiling phase is carried out correctly. Indeed, this therefore means that he has at its disposal a good estimation of leakage model and variance of traces, as demonstrated in Section 4.1.2. Moreover, as previously explained, using $\mathcal{DR}_{opt}$ as distinguishing rule instead of ELBO loss based distinguishing rule as done in [ZBC+23] (which also aims at maximizing the optimal distinguisher) presents an advantage. Indeed, it avoids additional management of a hyperparameter to recover secret key k^*, namely N_z the number of sampling over the latent variable *i.e.* the number of generated traces (see [ZBC+23] for more details).

Proposition 4. ***cVAE-OSM attack complexity.*** *The key recovery phase using cVAE-OSM is performed in constant time.*

Hence, our contribution highly reduces the time complexity related to the key recovery phase when using generative models as we drop from an $\mathcal{O}(N_z)$ attack complexity [ZBC+23] down to an $\mathcal{O}(1)$ complexity. The following section is dedicated to the validation of all these theoretical results with experiments.

5 Experimental results on cVAE-OSM

In this section, we aim to experimentally validate theoretical claims made on cVAE-OSM in the previous sections. First, we assess model properties in Section 5.1 using simulations. Then, we evaluate relevance of this model in the context of attacks in Section 5.2 using publicly available datasets. Finally, we summarize all results obtained in previous sections and conclude in Section 5.3.

5.1 Experimental results on simulations

This first section aims to validate theoretical claims made in previous sections using simulated traces. We present our setup in Section 5.1.1. The following sections namely Section 5.1.2, Section 5.1.3, Section 5.1.4 and Section 5.1.5 are dedicated to experiments carried out. For each of these sections, a brief description of the experiment is given, followed by a summary of obtained results.

5.1.1 Experimental setup

Implementation tricks. We implement our model in Python using Tensorflow [AAB+15] and Keras [C+15] libraries. We point out that when implementing cVAE-OSM, some implementation tricks must be considered to satisfy the theoretical results pointed in Section 3.3 and Section 4. For more details, our code is publicly available in the following GitHub repository: https://github.com/Sbsm157/cVAE-OSM.

Simulations. Let us denote the sensitive variable Y^{k^*} as $Y^{k^*} = Sbox[p \oplus k^*] \in \mathbb{F}_2^8$ and $(Y^{k^*})_b$ as the bth bit of Y^{k^*}. We also denote $\mathbf{N}_i \sim \mathcal{N}(0, \boldsymbol{\sigma}_i^2)$ as a Gaussian noise at the sample i and $\{l_1, \ldots, l_{np}\}$ as a set of np PoIs indexes contained in a trace $\mathbf{T}$. To assess claims made on cVAE-OSM in previous sections, we simulate traces which follow 4 scenarios corresponding to different leakage models:

- **Scenario 1**: Hamming weight leakage model – In this first scenario, we assume that the leakage model Ψ is defined by the Hamming Weight leakage model. Therefore, we define the ith sample of the trace $\mathbf{T}$ as:

$$\mathbf{T}_i = \begin{cases} \Psi_{HW} = \sum_{b=0}^{7} (Y^{k^*})_b + \mathbf{N}_i & \text{if } i \in \{l_1, \ldots, l_{np}\} \\ \\ \mathbf{N}_i & \text{otherwise.} \end{cases}$$

- **Scenario 2**: Independent bit leakage model − In this case, we assume that each bit leaks independently from the others. The ith sample of $\mathbf{T}$ is thus defined as:

$$\mathbf{T}_i = \begin{cases} \Psi_{IBL} = \sum_{b=0}^{7} \alpha_b \cdot (Y^{k^*})_b + \mathbf{N}_i & \text{if } i \in \{l_1, \ldots, l_{np}\} \\ \\ \mathbf{N}_i & \text{otherwise.} \end{cases}$$

  where $\alpha = [1, 0.5, 1, 0, 2, 0.5, 0.75, 0.25]$.

- **Scenario 3**: Multivariate leakage model − In this scenario, we consider a configuration in which the leakage also depends on bits' interaction. To simplify the equations, we exceptionally denote here $(Y^{k^*})_b$ *i.e.* the bth bit of the sensitive value Y^{k^*} as Y_b. Hence, we define the ith sample of $\mathbf{T}$ as:

$$\mathbf{T}_i = \begin{cases} \Psi_{ML} = 1 \cdot Y_0 + 2 \cdot Y_5 + 0.5 \cdot Y_6 + 0.25 \cdot r(Y_1 Y_3) \\ +1 \cdot r(Y_2 Y_4) + 0.15 \cdot r(Y_4 Y_7) + 1 \cdot r(Y_0 Y_5 Y_6) + 0.4 \cdot r(Y_1 Y_5 Y_7) \\ +0.3 \cdot r(Y_1 Y_6 Y_7) + 0.6 \cdot r(Y_2 Y_3 Y_4 Y_6) + 0.2 \cdot r(Y_3 Y_4 Y_5 Y_7) & \text{if } i \in \{l_1, \ldots, l_{np}\} \\ +0.1 \cdot r(Y_0 Y_1 Y_2 Y_5 Y_6) + 0.25 \cdot r(Y_0 Y_1 Y_2 Y_5 Y_7) \\ +0.5 \cdot r(Y_0 Y_1 Y_2 Y_5 Y_6 Y_7) + 0.75 \cdot r(Y_0 Y_1 Y_2 Y_3 Y_5 Y_6 Y_7) \\ +1 \cdot r(Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7) + \mathbf{N}_i \\ \mathbf{N}_i & \text{otherwise.} \end{cases}$$

  where r is a function that characterizes bits' interaction (see Equation 1).

- **Scenario 4**: Multiple leakage model − We decide to consider in this last scenario a configuration in which leakage model varies for all PoIs. Indeed, if the sample $\mathbf{T}_i$ is a PoI, then the leakage model associated with it is either Hamming weight, Independent bit or Multivariate. The aim of this scenario is to assess the cVAE-OSM ability to correctly characterize for each PoI the correct leakage model. Consequently, we define each sample $\mathbf{T}_i$ as:

$$\mathbf{T}_i = \begin{cases} \Psi_{HW} & \text{if } i \in \{l_1, \ldots, l_{np}\} \text{ and } i \equiv 0 \mod 3 \\ \Psi_{IBL} & \text{if } i \in \{l_1, \ldots, l_{np}\} \text{ and } i \equiv 1 \mod 3 \\ \Psi_{ML} & \text{if } i \in \{l_1, \ldots, l_{np}\} \text{ and } i \equiv 2 \mod 3 \\ \mathbf{N}_i & \text{otherwise.} \end{cases}$$

The different weights for the three last scenarios are arbitrarily chosen such that they vary across different bits and their interactions. For each of these scenarios, we simulate traces with different levels noise *i.e.* low, medium and high noise. This noise is either an isotropic white Gaussian noise *i.e.* $\mathbf{\Sigma} = \boldsymbol{\sigma}^2 \mathbf{I_D}$ or a non-isotropic white Gaussian noise *i.e.* $\mathbf{\Sigma} = \text{diag}[\boldsymbol{\sigma}_1^2, \ldots, \boldsymbol{\sigma}_D^2]$. For both configurations, we set $\boldsymbol{\sigma}$ such that the highest peak of the resulting SNR (Signal-to-Noise Ratio) for each of these noise levels does not exceed 16, 2.5 and 0.1 for low, medium and high noise respectively. Thus, we choose for all scenarios $\boldsymbol{\sigma} = 1$ and $\boldsymbol{\sigma} = 5$ for medium and high noise respectively. For low noise, we pick $\boldsymbol{\sigma} = 0.4$ for scenarios n°1, 2 and 4 and $\boldsymbol{\sigma} = 0.5$ for scenario n°3. To generate a non-isotropic white Gaussian noise for all 3 noise levels, we set each element of the diagonal matrix $\mathbf{\Sigma}$ by randomly picking an item from a set of values *i.e.*

$\boldsymbol{\sigma} \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2\}$ for low noise, $\boldsymbol{\sigma} \in \{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ for medium noise and $\boldsymbol{\sigma} \in \{5, 6, 7, 8, 9, 10, 11, 12, 13\}$ for high noise.

Our model is trained using raw traces (*i.e.* no preprocessing). Regarding hyperparameters used to train cVAE-OSM models, we choose Adam as optimizer [KB14] and set respectively the batch size, learning rate and number of epochs to $\{4, 8, 32, 128\}$, $\{0.1, 0.001, 0.0001\}$ and $\{50, 100, 200\}$. Finally, to ease leakage model Ψ interpretability, we use, as in [ZBC+23, Section 4], Guilley *et al.* orthonormal basis [GHMR17] which aims at clearly describe bits' interactions. Indeed, in their work they propose to orthonormalize the conventional monomial basis by applying a Gram-Schmidt process. In this way, this orthonormalized basis uncorrelates basis vectors, resulting in a clearer interpretability of the contribution of each monomial *i.e.* bits' interaction on Ψ leakage model.

5.1.2 Convergence of cVAE-OSM weights: leakage model and variance extraction

As explained in Section 3.3 and Section 4.1, the model namely the encoder E_ϕ and decoder D_θ optimize their parameters ϕ and θ respectively such that they extract from traces the leakage models $\hat{\Psi}_\phi$, $\hat{\Psi}_\theta$ and variance $\boldsymbol{\Sigma}_\phi$. The first experiment assesses cVAE-OSM ability to correctly extract Ψ and $\boldsymbol{\Sigma}$ during its training. We generate for each scenario $20,000$ traces containing $D = 150$ samples such that 10% are PoIs. Then, we train our model on theses traces considering a full basis (*i.e.* the size of the orthonormal basis is equal to 256) and visualize trainable parameters.

Isotropic noise. When considering Scenario 4 with a high isotropic Gaussian noise, the estimated $\hat{\Psi}_\phi$ (Figure 2a) and $\hat{\Psi}_\theta$ (Figure 2b) converge towards the same leakage model which is similar to the true leakage model Ψ (Figure 2c). Table 1a, which reports $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ mean squared error (MSE), confirms these findings, as the related MSE values for $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ are relatively low. Overall, looking at the MSE values in this table, we notice that for all noise levels and scenarios, cVAE-OSM is able to extract the leakage model. However, a difference in term of MSE value is noticeable between leakage models estimated by E_ϕ and D_θ (values for E_ϕ are larger than those for D_θ), especially for high noise level. This difference is visually confirmed when comparing Figure 2a and Figure 2b for example. This leads us to believe that the deterministic part estimated by the decoder $\hat{\Psi}_\theta$ seems to "converge better" than the one estimated by the encoder namely $\hat{\Psi}_\phi$. This difference in convergence between encoder and decoder weights could be explained by the fact that during cVAE-OSM training, encoder weights are impacted by the minimization of KL-divergence loss (see Section 4.1). Based on the KL-divergence loss defined for this model in Proposition 2, this loss and thus encoder forces monovariate traces $\tilde{\mathbf{T}}$ to follow a normal distribution which is an approximation of their real distribution *i.e.* a chi-squared distribution (see Corollary 1). This approximation of distribution during KL-divergence loss minimization could explain encoder weights convergence towards an approximation of Ψ coefficients. All in all, these observations suggest that during cVAE-OSM training, E_ϕ and D_θ are able to extract the leakage model, even when the leakage model is not the same for all samples and when noisy configurations are considered.

Regarding $\boldsymbol{\Sigma}$ estimation, we summarize results obtained when considering isotropic noise in Table 1b. Here again, we observe that, for any noise level and scenario, cVAE-OSM approximation for $\boldsymbol{\Sigma}$ does not exceed a 5% margin of error. This suggests that E_ϕ is able to extract $\boldsymbol{\Sigma}$ from traces. Hence, it appears that during its training, cVAE-OSM can indeed extract and converge towards true leakage model Ψ and variance $\boldsymbol{\Sigma}$, when considering an isotropic noise.

Non-Isotropic noise. In order to provide the most comprehensive assessment possible, we carry out experiments to also determine whether our model is able to correctly extract

Table 1: All scenarios, Isotropic noise – Convergence of $\hat{\Psi}_\phi$, $\hat{\Psi}_\theta$ and $\mathbf{\Sigma}_\phi$ according to noise levels.

		$\hat{\Psi}_\phi$ MSE	$\hat{\Psi}_\theta$ MSE
Scenario 1	Low noise	0.016	0.006
	Medium noise	0.071	0.015
	High noise	1.51	0.40
Scenario 2	Low noise	0.014	0.004
	Medium noise	0.07	0.013
	High noise	1.12	0.26
Scenario 3	Low noise	0.048	0.034
	Medium noise	0.095	0.039
	High noise	1.37	0.04
Scenario 4	Low noise	0.03	0.02
	Medium noise	0.08	0.02
	High noise	1.76	0.31

(a) Mean Squared Error (MSE) between $\hat{\Psi}_\phi$ (resp. $\hat{\Psi}_\theta$) and the true leakage model for all scenarios.

		$\mathbf{\Sigma}_\phi$
Scenario 1	Low noise	2.49%
	Medium noise	2.17%
	High noise	3.7%
Scenario 2	Low noise	2.62%
	Medium noise	2.12%
	High noise	2.4%
Scenario 3	Low noise	2.31%
	Medium noise	2.04%
	High noise	4.88%
Scenario 4	Low noise	2.52%
	Medium noise	1.95%
	High noise	3.5%

(b) Approximation errors between $\mathbf{\Sigma}_\phi$ and the true $\mathbf{\Sigma}$ for all scenarios.

Table 2: Scenario 4 Non isotropic noise– Convergence of $\hat{\Psi}_\phi$, $\hat{\Psi}_\theta$ and $\mathbf{\Sigma}_\phi$ according to noise levels.

	$\hat{\Psi}_\phi$ MSE	$\hat{\Psi}_\theta$ MSE
Low noise	0.07	0.02
Medium noise	0.61	0.03
High noise	2.81	0.6

(a) Mean Squared Error (MSE) between $\hat{\Psi}_\phi$ (resp. $\hat{\Psi}_\theta$) and the true leakage model Ψ.

	$\mathbf{\Sigma}_\phi$
Low noise	2.77%
Medium noise	4%
High noise	3.95%

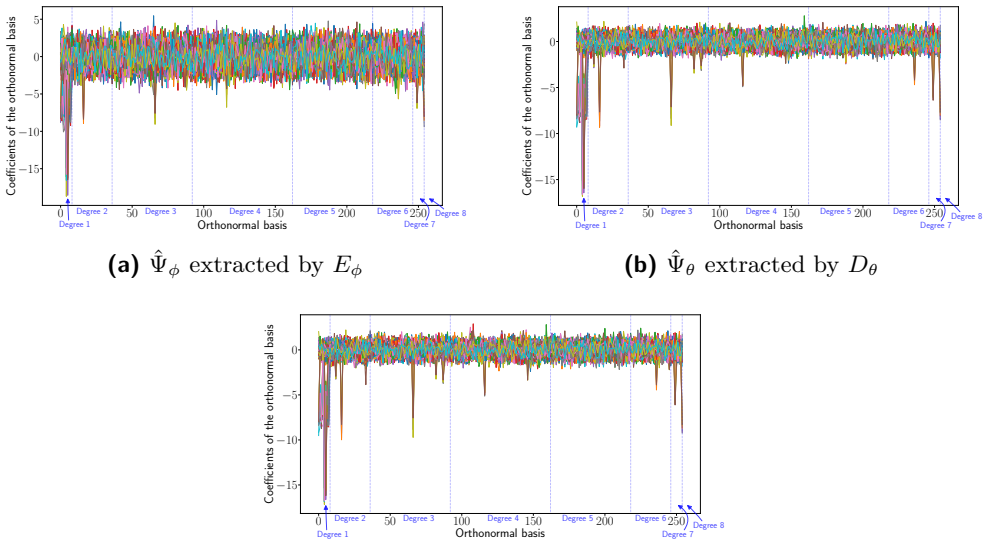
(b) Approximation errors between $\mathbf{\Sigma}_\phi$ and the true $\mathbf{\Sigma}$.

the leakage model and variance in a non-isotropic noise case. To do so, we proceed in a similar way to what we did previously for isotropic noise, considering in this case a non-isotropic noise. The results obtained are in Table 2. We only report in this table the results for the fourth scenario as it is the more complex one and the results found for the other scenarios are similar to those found in Table 1. As we can see, the same observations hold true for traces with isotropic Gaussian noise. Indeed, according to MSE values reported in Table 2a, cVAE-OSM is able to correctly extract leakage model Ψ since MSE values associated to Ψ_ϕ and Ψ_θ are low, regardless of the noise level considered. For $\mathbf{\Sigma}$ estimation, here again, our model seems to correctly estimate it as the approximation errors of $\mathbf{\Sigma}_\phi$ do not exceed a 4% margin of error, for all noise level.

Finally, through these experiments, it appears that during its training, cVAE-OSM can indeed extract and converge towards true leakage model Ψ and (isotropic/non-isotropic) variance $\mathbf{\Sigma}$, even in a noisy configuration. In the following experiments, the results presented will only concern the last scenario *i.e.* the fourth one, unless otherwise indicated, as it is the most complex dataset and the related leakage model is a merge of leakage models of the three first scenarios.

5.1.3 Optimal dimensionality reduction

As stated in Section 3.3, cVAE-OSM is designed to extract leakage model and variance from traces by computing an optimal dimensionality reduction. In this experiment, we evaluate cVAE-OSM ability to perform a single-point compression without information loss. We simulate $256,000$ traces, following Scenario 4, containing $D = \{10, 50, 150\}$ samples (such that 10% are PoIs) for different noise levels $\boldsymbol{\sigma} = \{0.1, 0.2, 0.4, 0.8, 1, 2, 4, 10, 25\}$. We



(a) $\hat{\Psi}_\phi$ extracted by E_ϕ

(b) $\hat{\Psi}_\theta$ extracted by D_θ

(c) True leakage model Ψ

Figure 2: Scenario 4 Isotropic high noise − Visualization of the leakage models $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ for all samples extracted by cVAE-OSM, excluding the bias.

then compute Mutual Information (MI) which reflects the amount of information contained in traces and Perceived Information (PI) which corresponds to information extracted by a statistical model from traces [RSVC+11]. Given a secret key variable K and a trace variable $\mathbf{T}$, MI and PI metrics are defined as:

$$MI(K;\mathbf{T}) = H(K) + \sum_{k\in\mathcal{K}} \mathbb{P}(k) \cdot \sum_{\mathbf{t}\in\mathcal{T}} \mathbb{P}(\mathbf{t}|k) \cdot \log_2 \mathbb{P}(k|\mathbf{t})$$
$$PI(K;\mathbf{T}) = H(K) + \sum_{k\in\mathcal{K}} \mathbb{P}(k) \cdot \sum_{\mathbf{t}\in\mathcal{T}} \mathbb{P}(\mathbf{t}|k) \cdot \log_2 \mathsf{m}(k|\mathbf{t})$$

where $H(K) = \log_2(|\mathcal{K}|)$ is the entropy of K and m is a $\mathbb{P}(k|\mathbf{t})$ estimation made by a statistical model. However, since the true $\mathbb{P}(k|\mathbf{t})$ distribution is unknown, MI and PI metrics are approximated by using the sampling process introduced in [BHM+19, Equations 6 and 11]. In the following, the estimated MI and PI will be respectively denoted $\widehat{MI}$ and $\widehat{PI}$. Note that the computation of $\widehat{MI}$ is only possible in our context because the traces considered are simulated. By comparing $\widehat{MI}$ and $\widehat{PI}$, we can quantify information loss due to single-point trace compression, and thus assess if cVAE-OSM achieves to perform an optimal dimensionality reduction *i.e.* a data compression without loss of information. Indeed if MI and PI are similar, this means that parameters estimated by cVAE-OSM are sufficient to accurately describe traces (leakage model and noise). One can therefore assume that the dimensionality reduction performed by cVAE-OSM is done without information loss. Figure 3 summarizes results found. We can observe that for any $D = \{10, 50, 150\}$, $\widehat{MI}$ scores and $\widehat{PI}$ scores are equivalent. These results are in line with previous experimental conclusions: cVAE-OSM can indeed extract leakage model and noise of traces. Thus, this also suggests that dimensionality reduction done in cVAE-OSM is performed without any information loss which seems beneficial from an attack perspective.

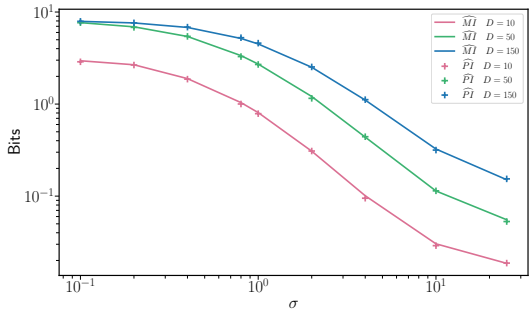


Figure 3: Evolution of estimated Mutual Information $\widehat{MI}$ and estimated Perceived Information $\widehat{PI}$ as a function of σ.

Table 3: Scenario 4 Isotropic high noise − Convergence of $\hat{\Psi}_\phi$, $\hat{\Psi}_\theta$ and $\mathbf{\Sigma}_\phi$ as a function of number of traces N used for training.

	$\hat{\Psi}_\phi$ MSE	$\hat{\Psi}_\theta$ MSE
$N = 2D$	0.94	0.00045
$N = D$	1.31	0.05
$N = D/2$	1.79	0.44
$N = D/3$	2.06	0.83

(a) Mean Squared Error (MSE) between $\hat{\Psi}_\phi$ (resp. $\hat{\Psi}_\theta$) and the true leakage model Ψ.

	$\mathbf{\Sigma}_\phi$
$N = 2D$	3.01%
$N = D$	2.99%
$N = D/2$	3.06%
$N = D/3$	3.39%

(b) Approximation errors between $\mathbf{\Sigma}_\phi$ and $\mathbf{\Sigma}$.

5.1.4 Curse of dimensionality: HDLSS/SSS problem

As mentioned in Section 3.2 one advantage to use a cVAE to perform the optimal dimensionality reduction is that a cVAE unlike conventional techniques, *e.g.* LDA, does not suffer from practical limitations namely *High-Dimension Low Sample Size* (HDLSS) or *Small Sample Size* (SSS) problem. As a reminder, HDLSS or SSS problem arises whenever the number of traces per class is smaller than the number of samples [TGIH17]. In this section, we want to evaluate cVAE-OSM ability to overcome HDLSS/SSS problem. To do so, we simulate, for each noise level, N traces per class of $D = 150$ samples following an isotropic Gaussian noise and then train our model with various numbers of traces $N = \{2D, D, D/2, D/3\}$. In Table 3, we report results obtained for Scenario 4 when considering a high noise level. When observing MSE found in Table 3a, we can see that the lower the number of traces, the higher the MSE, while remaining reasonable *i.e.* the model is still able to extract the deterministic part. Indeed, in our context, a MSE of 2 does not highly impact the extraction of Ψ (see Figure 2a which corresponds to a MSE of 1.76). Hence, this means that our model is still able to extract both $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ leakages models when the number of traces $N < D$. Regarding $\mathbf{\Sigma}_\phi$ estimation, we observe that for any N considered, approximation errors are about 3% (see Table 3b), which lead us to conclude that cVAE-OSM is able to converge toward a good approximation of $\mathbf{\Sigma}$. Hence, according to these results, cVAE-OSM is able to converge similarly even when considering a limited number of traces. In this context, cVAE-OSM seems to overcome HDLSS/SSS problem and thus can be considered as a good alternative to LDA.

5.1.5 Practical issues

While performing an attack in practice, one can face several issues such as the number of samples D to be processed, attack efficiency with respect to PoIs selection or the impact

Table 4: Scenario 4 Isotropic high noise − Mean Squared Error (MSE) between $\hat{\Psi}_\phi$ (resp. $\hat{\Psi}_\theta$) extracted by cVAE-OSM and the true leakage model Ψ as a function of number of PoIs.

	$\hat{\Psi}_\phi$ MSE	$\hat{\Psi}_\theta$ MSE
0.1% PoIs	0.48	0.0042
1% PoIs	0.48	0.0067
50% PoIs	0.59	0.11
100% PoIs	0.67	0.22

of basis size choice. In this section, we want to assess if these issues influence cVAE-OSM capability to extract leakage model and variance from traces.

Impact of PoIs in leakage model extraction. In this experiment, we evaluate the impact of uninformative samples on leakage model extraction. To do so, we simulate $20,000$ traces of $D = 1,000$ samples such that $\{0.1\%, 1\%, 50\%, 100\%\}$ are PoIs. Theses traces follow Scenario 4 with an isotropic Gaussian noise and a high noise level. Table 4 shows mean squared errors obtained after $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ extraction by encoder E_ϕ (resp. decoder D_θ) once cVAE-OSM is trained for each PoIs' configuration. According to these results, even when considering a noisy configuration, we observe low MSE. This therefore means $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ converge towards the same leakage model namely the true leakage model Ψ, regardless of the number of PoIs. Hence, it appears that uninformative samples do not influence the convergence of model weights since cVAE-OSM is able to estimate the true leakage model Ψ. However, although uninformative samples do not influence leakage model extraction, we will see in Section 5.2 that these uninformative samples impact attacks efficiency.

Impact of D on convergence of cVAE-OSM weights. In Remark 1, we approximate the $\chi^2(D)$ distribution followed by monovariate traces $\tilde{\mathbf{T}}$ (*i.e.* single-point compressed traces) to a $\mathcal{N}(D, 2D)$ normal distribution on the grounds that side-channel traces have usually $D > 100$ samples. Then, we design cVAE-OSM architecture and KL-divergence loss based on this approximation. In this section, we want to evaluate the impact of D on the model. More specifically, as this approximation is used to define KL-divergence loss (see Proposition 2) and this latter is used for encoder E_ϕ weights estimation, we consequently want to assess the impact of D on convergence of E_ϕ weights. We simulate $20,000$ traces following an isotropic Gaussian noise and containing $D = \{5; 25; 50; 150; 1,000\}$ samples such that 10% are PoIs. We then train our model on these traces. Results obtained for the configuration in which Scenario 4 and a high noise are considered are reported in Table 5. As we can see in this table, values taken by D impact convergence of E_ϕ weights: it seems more difficult to extract the deterministic part Ψ and the variance $\mathbf{\Sigma}$ for small D. Indeed, for $D = 5$, $D = 25$ and $D = 50$, the deterministic parts $\hat{\Psi}_\phi$ extracted by E_ϕ are noisier and less accurate (leading to higher MSE) than those extracted when $D = 150$ and $D = 1,000$ for example (MSE are lower). A similar observation can be made concerning the variance $\mathbf{\Sigma}_{\boldsymbol{\phi}}$ estimated by E_ϕ: the smaller D, the greater the approximation error. These results emphasize the fact that for small D, it is not possible to accurately approximate a χ^2-distribution with a normal distribution and that this approximation subsequently induces errors during the estimation of leakage model Ψ and variance $\mathbf{\Sigma}$. This is thus in line with Remark 1, namely that to ensure convergence of E_ϕ weights, D must be greater or equal to 100. However, as we can see in Figure 3, these results must be qualified. Indeed, when $D = 10$ and $D = 50$, PI estimations are consistent with those of MI meaning that there is no information loss during leakage model and variance extractions made by E_ϕ. Thus, this implies that cVAE-OSM is able to extract an accurate estimation of the true leakage model and variance even when $D < 100$.

Table 5: Scenario 4 Isotropic high noise – Convergence of $\hat{\Psi}_\phi$ and $\mathbf{\Sigma}_\phi$ as a function of samples number D considered.

	$\hat{\Psi}_\phi$ MSE	$\mathbf{\Sigma}_\phi$
$D = 5$	2.35	11.27%
$D = 25$	1.59	4.53%
$D = 50$	1.81	4.15%
$D = 150$	1.76	3.5%
$D = 1,000$	1.11	3%

Table 6: Scenarios 1 and 2 Isotropic high noise – Convergence of $\hat{\Psi}_\phi$, $\hat{\Psi}_\theta$ and $\mathbf{\Sigma}_\phi$ according to basis size.

		$\hat{\Psi}_\phi$ MSE	$\hat{\Psi}_\theta$ MSE
Scenario 1	Basis size = 9	1.11	0.23
	Basis size = 256	1.51	0.41
Scenario 2	Basis size = 9	0.63	0.09
	Basis size = 256	1.12	0.26

(a) Mean Squared Error (MSE) between $\hat{\Psi}_\phi$ (resp. $\hat{\Psi}_\theta$) and the true leakage model for the three scenarios.

		$\mathbf{\Sigma}_\phi$
Scenario 1	Basis size = 9	3.9%
	Basis size = 256	3.7%
Scenario 2	Basis size = 9	2.98%
	Basis size = 256	2.4%

(b) Approximation errors between $\mathbf{\Sigma}_\phi$ and the true $\mathbf{\Sigma}$ for scenarios 1 and 2.

Impact of basis size. As explained previously, to conduct the previous experiments, we train our model considering a full basis size in order to also take into account hypothetical high degree bits interactions. In this experiment, we want to assess the impact of basis size choice on the convergence of our model. Thus, in the following, we focus only on Scenarios 1 and 2 *i.e.* Hamming weight and Independent Bit leakage models since among the 4 scenarios (see Section 5.1.1), these are the only configurations in which there are no bit interactions. Training is done considering different choice of basis size: since labels used for training are bytes, we choose a first basis size equal to 9 (including the bias). This corresponds to a configuration in which we consider no interaction between bits. The second basis size chosen is 256 (including the bias), meaning that we also consider high degree of bits interactions. We report results obtained in Table 6 for a high noise level. If we refer to Table 6b, we observe a slight decrease in $\mathbf{\Sigma}$ approximation when a smaller base is chosen. Generally speaking, we can consider that $\mathbf{\Sigma}$ approximation remains of the same order and thus that basis size choice for estimating $\mathbf{\Sigma}$ has little impact on the model. This is somewhat less the case for the leakage model approximation. Looking at Table 6a, we see that choosing a smaller basis size reduces MSE between $\hat{\Psi}_\phi$ (resp. $\hat{\Psi}_\theta$) and the true leakage model down by a factor of about 2. Basis size therefore seems to have a slight impact on leakage model estimation. As a conclusion, choosing a suitable basis size seems to give a better estimate of leakage model coefficients. However, even with a poor choice of basis size, our model is still able to extract the true leakage model. Therefore, it appears that our model is robust to a poor choice of basis size. Our model is thus generic in terms of the choice of basis size, as assuming a full basis by default does not compromise the extraction of a good estimation of leakage model and variance by our model.

5.2 Experimental results on real datasets

Now that we have assessed cVAE-OSM properties on simulations, we evaluate its relevance in the context of attacks using publicly available datasets. To do so, we present in Section 5.2.1 our setup and in Section 5.2.2, Section 5.2.3, Section 5.2.4 and Section 5.2.5 results obtained for each of these datasets.

5.2.1 Setup

Presentation of datasets. To assess cVAE-OSM relevance for attacks, we evaluate our model on diversified datasets that allow us to cover a wide range of use cases. More specifically, we select 4 different datasets whose implementations correspond to either conventional symmetric cryptography with AES [DR02] or post-quantum cryptography with NTRU [HPS98, CDH+19] cryptosystem. To ensure that we cover a wide range of use cases, the chosen datasets correspond to different types of implementations: unprotected software implementation on smart cards, unprotected hardware implementation on FPGA, protected software implementation with first-order masking.

- **DPA contest v4.2** is a dataset which corresponds to an AES-128 software implementation with first-order masking and a shuffling countermeasure on an Atmel ATMega-163 smart-card [TP08, BBD+14]. The dataset is available at https://dpacontest.telecom-paris.fr/v4. In our case, we consider this implementation as unprotected since we assume that the attacker has at its disposal mask and shuffling offsets values. We target the first round SBox operation. Knowing the shuffling offset value *offset* and mask value m, we associate to each trace $\mathbf{t}$ its corresponding label $SBox(p_0 \oplus k^*) \oplus m[offset + 1]$ where p_0 denotes the first byte of the processed plaintext.

- **AES_HD_Ext** is an extension of AES_HD dataset introduced in [PHJ+18] and is available at https://github.com/AISyLab/AES_HD_Ext. This dataset is an unprotected AES-128 hardware implementation on Xilinx Virtex-5 FPGA. In this implementation, the authors target the register writing in the last round such that for each leakage trace $\mathbf{t}$, the associated label is computed in the following way: $SBox^{-1}(c_j \oplus k^*) \oplus c_{j'}$, where c_j and $c_{j'}$ are two ciphertext bytes such that the relation between j and j' is given through the inverse ShiftRows operation of AES. The authors choose $j = 12$ and $j' = 8$.

- **ASCAD v1** is a dataset introduced in [BPS+20] which corresponds to a protected AES-128 software implementation, running over an 8-bit AVR microcontroller namely an ATMega8515. The countermeasure consists in a first-order boolean masking. The whole database is available at https://github.com/ANSSI-FR/ASCAD. In this implementation, for each leakage trace $\mathbf{t}$, we target the third byte of the first round SBox operation such that the associated sensitive value is $SBox(p_3 \oplus k^*)$. However, since this implementation is protected, $\mathbf{t}$ is related to the manipulation of the value $SBox(p_3 \oplus k^*) \oplus r_3$ where r_3 is a mask. In our experiments, we consider two versions: **ASCAD v1-F** (fixed key for profiling) and **ASCAD v1-R** (random key for profiling).

- **SCANTRU** is a dataset publicly available at https://github.com/ANSSI-FR/scantru and introduced in [BEM+23]. It consists in a protected software implementation on a STM32F407 discovery board of a lattice-based key encapsulation mechanism (KEM). More specifically, the KEM chosen is NTRU HPS variant with parameters $n = 509$ and $q = 2^{11} = 2048$ [CDH+19]. In this implementation, the authors target an operation within the KEM decapsulation routine namely a polynomial product occurring during `Decrypt` algorithm between the secret key $f \in \mathbb{Z}_3[X]/(X^{509} - 1)$ (*i.e.* a polynomial of $n = 509$ coefficients in $\{-1, 0, 1\}$[a]) and ciphertext $c \in \mathbb{Z}_{2^{11}}[X]/(X^{509} - 1)$. Two countermeasures are integrated in this implementation: a blind decryption countermeasure which consists in masking ciphertexts in order to prevent chosen ciphertext attacks (CCA) and a shuffling countermeasure which consists in rotating the secret polynomial f when executing the product $c \times f$, resulting in the randomization of secret coefficients order and therefore operations order. In our experiments, the blind

[a] For the sake of simplicity, we note that coefficients of f are in $\{-1, 0, 1\}$ instead of $\{0, 1, 2047\}$. See [CDH+19, BEM+23] for more details.

decryption countermeasure does not impact our attack scenario since the knowledge of ciphertexts c is not required to retrieve coefficients of the secret f. Moreover, we assume that the rotation is known by the attacker. Consequently we consider this implementation as unprotected. Hence, knowing the value of rotation r, each leakage trace $\mathbf{t}$ corresponds to the processing of $c_i \times f_{(n-i+r) \bmod n}$ where c_i corresponds to the ith coefficient of an unknown ciphertext c and $f_{(n-i+r) \bmod n}$ is the corresponding rotated coefficient of secret key f.

Configuration. We conduct our experiments on a laptop equipped with 8GB RAM and a 2.6 GHz Intel Core i7-9750H processor. We use the same implementation as the one described in Section 5.1.1. We train our model using Adam as optimizer and set respectively the batch size, learning rate and number of epochs to $\{2, 8, 32\}$, $\{0.01, 0.001, 0.0001\}$ and $\{10, 200\}$. As for simulations, to ease leakage model interpretability, we use Guilley *et al.* orthonormal basis [GHMR17]. For each experiment, we consider a maximal bits' interaction number *i.e.* a full orthonormal basis. For all datasets, the attack consists in following steps described in Section 4.2 *i.e.* training on profiling traces (full traces) then, after an automatic PoIs selection, maximizing the optimal distinguisher (see Definition 3) to recover the key. Based on deterministic part coefficients estimated by both encoder and decoder, we define a way to select PoIs *i.e. sum method*. Given estimated deterministic part coefficients, we sum for each sample the absolute values of its associated coefficients excluding the bias and define as PoIs samples with the highest sum of coefficients. By summing coefficients, one can have for each sample a "score" corresponding to its total contribution in the leakage model (*i.e.* how informative each sample is overall). The idea is that the higher the contribution, the more likely the sample is a PoI. When applying this method, we exclude bias because this value is essentially constant for all samples. It consequently cannot be used to determine whether or not a sample can be considered as a PoI. Regarding preprocessing, we consider either raw traces or one of the following preprocessing methods:

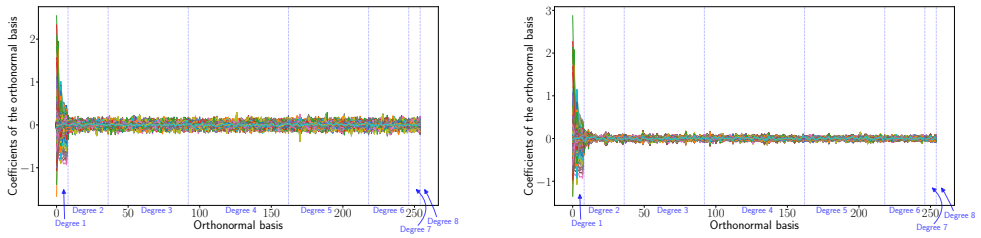
- ▷ **Maximum absolute value scaling**: this technique is used to scale each sample by its maximum absolute value. Hence, given a diagonal matrix $B \in \mathbb{R}^{D \times D}$ in which each element corresponds to a scaling factor, instead of having $\mathbf{T} \sim \mathcal{N}_D(\Psi(Y^{k^*}), \mathbf{\Sigma})$ where $\Psi(Y^{k^*}) \in \mathbb{R}^D$ and $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$, $\mathbf{T}$ follows after this scaling $\mathcal{N}_D(\boldsymbol{\mu}', \mathbf{\Sigma}')$ where $\boldsymbol{\mu}' = B\Psi(Y^{k^*})$, $\mathbf{\Sigma}' = B\mathbf{\Sigma}B^T$. When training our model with these preprocessed traces, cVAE-OSM will extract an estimation of $B\Psi(Y^{k^*})$ and $B\mathbf{\Sigma}B^T$ instead of $\Psi(Y^{k^*})$ and $\mathbf{\Sigma}$. Therefore, this type of preprocessing does not alter mathematical prerequisites needed for cVAE-OSM convergence as well as the key extraction phase.

- ▷ **Bias suppression**: this method is used to suppress traces bias *i.e.* traces modulation. It is done by computing the empirical mean $\hat{\boldsymbol{\mu}} \in \mathbb{R}^D$, which we subsequently subtract from the traces. Therefore, $\mathbf{T} \sim \mathcal{N}_D(\boldsymbol{\mu}', \mathbf{\Sigma})$ where $\boldsymbol{\mu}' = \Psi(Y^{k^*}) - \hat{\boldsymbol{\mu}}$ instead of following $\mathcal{N}_D(\Psi(Y^{k^*}), \mathbf{\Sigma})$. Consequently, cVAE-OSM will estimate $\Psi(Y^{k^*}) - \hat{\boldsymbol{\mu}}$ instead of $\Psi(Y^{k^*})$. Here again, as for maximum absolute value scaling, this preprocessing does not modify prerequisites required for model convergence as well as the key extraction phase.

We consider the guessing entropy as attack metric. As a reminder, this metric is defined as the expected rank of the true key k^* [SMY09]. In the following sections, we present for each dataset the chosen configuration and the results obtained.

5.2.2 Results obtained on DPA Contest v4 dataset

Configuration. To attack **DPA contest v4** dataset, we first apply on profiling traces a maximum absolute value scaling. Then, we train our model on $25,000$ profiling traces of dimension $D = 4,000$ for 200 epochs using Adam optimizer, a batch size and a learning



(a) $\hat{\Psi}_\phi$ extracted by E_ϕ **(b)** $\hat{\Psi}_\theta$ extracted by D_θ

Figure 4: DPA contest v4 – Visualization of the leakage models $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ for all samples extracted by cVAE-OSM, excluding the bias.

Table 7: DPA contest v4 – Average minimum traces to key disclosure.

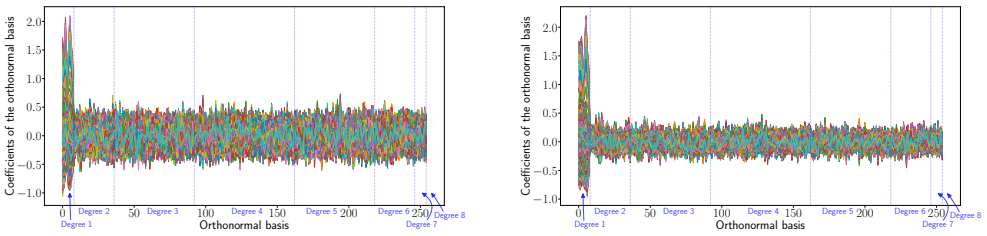
Template attacks	Stochastic attacks	CNN	cVAE-SA [ZBC+23]	cVAE-OSM (This work)
4 [KPH+19]	4	3 [ZBHV20]	4	7 (E_ϕ and D_θ)

rate of respectively 8 and 0.001 in order to retrieve leakage model and variance estimations. Once training is done, as we can see in Figure 4, we visualize leakage models extracted by the cVAE-OSM to select a subset of bits' interactions. In our case we consider interactions of up to 2 bits. Based on this subset, we select 200 PoIs using the sum method. The results obtained by averaging 10 independent shuffled experiments are reported in Table 7.

Results. As we can see in Table 7, our model achieves a rank of 1 for 7 traces whether using the deterministic part extracted by E_ϕ or D_θ. These results are not surprising and are in line with results shown in Figure 4 *i.e.* both deterministic parts estimated by E_ϕ and D_θ converge towards the same coefficients values as stated in Section 4.1.

In addition, Table 7 summarizes the number of traces required to achieve a constant rank of 1 for different types of attacks namely conventional profiled attacks and DLSCA, including our attack, when considering **DPA contest v4** dataset. Results of conventional profiled attacks (template and stochastic attacks) correspond to results obtained by performing these attacks on the 50 samples that maximize the Signal-to-Noise Ratio (SNR) and considering for stochastic attacks Guilley *et al.* orthornormal basis [GHMR17]. As we can see in this table, when comparing results obtained by our model with those obtained by conventional profiled attacks, we remark that our model performs similarly to template and stochastic attacks. The slight difference in the number of traces *i.e.* 7 traces for cVAE-OSM against 4 traces for conventional profiled attacks can be explained by the fact that, unlike template and stochastic attacks, the attack phase for cVAE-OSM is performed using 200 PoIs. This intuition is confirmed, indeed, when performing template and stochastic attacks with 200 PoIs, the number of traces required amounts then to 6 traces.

Regarding DLSCA using generative models, our model achieves results comparable to cVAE-SA [ZBC+23]. The slight difference in number of traces can be explained, by the fact that the attack is carried out with fewer constraints for our model in comparison with the attack conducted in [ZBC+23]. In their paper, Zaid *et al.* explain that their profiling and attack steps are done on the 50 samples that maximize the SNR, considering a number of sampling over the latent space $N_z = 100$ (*i.e.* generation of 100 synthetic traces is necessary) to correctly approximate $\mathbb{P}(\mathbf{T}|P,k)$ given a plaintext P and a key hypothesis k. While we, on the other hand, consider a less restrictive attack strategy as we conduct a profiling step using the entire traces (4,000 samples) and then carry out the attack on 200 PoIs without the need of generating traces to estimate $\mathbb{P}(\mathbf{T}|P,k)$ (see Proposition 4) *i.e.*



(a) $\hat{\Psi}_\phi$ extracted by E_ϕ **(b)** $\hat{\Psi}_\theta$ extracted by D_θ

Figure 5: AES_HD_Ext – Visualization of the leakage models $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ for all samples extracted by cVAE-OSM, excluding the bias.

Table 8: AES_HD_Ext – Average minimum traces to key disclosure.

Template attacks	Stochastic attacks	MLP	cVAE-SA [ZBC+23]	cVAE-OSM (This work)
5,300	5,100	902[1][RPP22]	-	1,400 (E_ϕ and D_θ)

[1] This result is not averaged over 10 independent experiments.

our time complexity for the key recovery phase is divided by 100. Consequently, our model achieves comparable results with a less restrictive strategy than in [ZBC+23]. Regarding discriminative models' performances, 3 traces are needed to achieve a constant rank of 1. As for previous results, performances are quite similar, with the advantage for our model, unlike CNN, of being fully explicable and interpretable. Hence, through this experiment, we deduce that when it comes to attacking a high-SNR unprotected implementation, our model performs just as well as conventional profiled attacks or DLSCA.

5.2.3 Results obtained on AES_HD_Ext dataset

Configuration. Regarding **AES_HD_Ext** dataset, we first reduce bias included in the set of 250,000 profiling traces of dimension $D = 1,250$ by first applying the bias suppression preprocessing method and then adding an arbitrary constant. In our case, we set this constant to 5. Then, we train for 10 epochs our model on these traces using Adam optimizer, a batch size of 32 and a learning rate of 0.001. By visualizing leakage model extracted by cVAE-OSM (see Figure 5), we only consider coefficients associated with single bits. Based on this subset, we select 100 PoIs using the sum method. Results obtained by averaging 10 independent shuffled experiments are presented in Table 8.

Results. As for **DPA contest v4** dataset, the same observations hold true for **AES_HD_Ext** dataset: our model performs similarly regardless of the extracted deterministic part used (see Table 8), the number of traces required to reach a constant rank of 1 is about 1,400. In addition, Table 8 summarizes number of traces needed to achieve a constant rank of 1 for conventional profiled attacks and DLSCA[b]. The results for conventional profiled attacks are achieved by applying the same configuration as for our model. This therefore means that these results are obtained by averaging 10 attacks on the 100 samples with the highest SNR value and considering for stochastic attacks, Guilley *et al.* orthonormal basis.

As we can see in Table 8, template and stochastic attacks require respectively 5,300 and 5,100 traces to reach a constant rank of 1, whereas cVAE-OSM needs 1,400 traces to

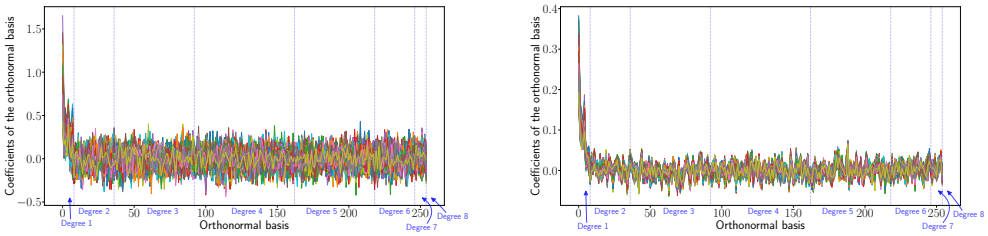
[b]The results here only take into account those obtained on **AES_HD_Ext** dataset and not on AES_HD dataset. Indeed, although these two datasets are unprotected implementations of AES on FPGA, as they come from two different acquisition campaigns, it leads to different deterministic part coefficients. Consequently, this makes the comparison of attack performances on these two datasets difficult.

converge towards a rank of 1. Interestingly, by using our model instead of conventional profiled attacks, the attack is improved by a factor of about 3.7. These results are surprising, as in the case of a white Gaussian noise, cVAE-OSM and conventional attacks correspond both to a Gaussian naive Bayes estimator. We identify two possible explanations that could justify this gap in performance between our model and conventional profiled attacks. The first explanation concerns the basis size considered during the attack phase. Indeed, cVAE-OSM allows greater flexibility during the attack phase because, as explained in Section 4.2, the attacker can select a subset of coefficients during the attack phase to enhance attack performance. This is not the case for stochastic attacks as they are performed using the full basis. Although this first reason is relevant, it does not fully explain this difference performance. Indeed, when considering a full basis *i.e.* 256 coefficients, cVAE-OSM needs 3,000 and 1,400 traces to converge towards a rank of 1 using $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ respectively. Altogether, it seems that E_ϕ in particular is affected by uninformative coefficients if the full basis is considered. Another possible explanation lies in the way the profiling phase is carried out (since this is the only aspect that differs from conventional profiled attacks, apart from the basis size considered). Indeed, we assume a better estimation of deterministic part coefficients is made by our model in comparison with conventional profiled attacks because the profiling is carried out taking into account the entire set of trace samples. In fact, the need to reduce the dimensionality of traces by selecting a certain number of samples with the highest SNR values before performing conventional profiled attacks may imply a loss of information which could potentially affect parameters estimations and thus degrade attack performances. This situation cannot occur when using our model, as the estimation of these same parameters is carried out via a dimensionality reduction that is without loss of information (see Section 3.1 and Section 3.3 for theoretical results and Section 5.1.3 for experimental results). Hence, these first results highlight the benefits of using cVAE-OSM model over conventional profiled attacks. Indeed, by using our model, the attacker is assured that parameters estimations are carried out using all of the information contained in traces, unlike conventional profiled attacks.

Regarding DLSCA, discriminative models namely Multilayer Perceptron (MLP) performs better than cVAE-OSM model. Indeed, MLP reaches a rank of 1 for about 900 traces *i.e.* 500 fewer traces than our model. Nevertheless, the result for discriminative models must be put into perspective, as it corresponds to the best attack *i.e.* this result is not averaged in [RPP22]. By way of comparison, without averaging the attacks, our best attack retrieves the key within 110 traces. Hence, it is difficult to determine which model, *i.e.* a discriminative or a generative model, is best suited. However, based on all observations made and the fact that discriminative models should perform better than generative models [NJ01], it appears that the best discriminative model has not yet been identified for this dataset. Nevertheless, the high number of parameters that must be taken into account to design this discriminative model can constitute a non-negligible limitation for an attacker/evaluator. Hence, using cVAE-OSM proves to be a sound alternative for an evaluator as its architecture is deterministic and it enables him to carry out an end-to-end interpretable attack without incurring a major loss of performance.

5.2.4 Results obtained on ASCAD dataset

Preprocessing. Unlike cVAE-OSM, discriminative models are able to directly perform high-order attacks. Indeed, this type of model carries out a non-linear complex recombination of PoIs in a way that is difficult to interpret. Indeed, a masked implementation consists in splitting a sensitive value Y^{k^*} in several shares such that given n_s the number of shares, $Y^{k^*} = Y_0^{k^*} \star Y_1^{k^*} \star ... \star Y_{n_s}^{k^*}$ for some field operation $\star$. Thus, to retrieve Y^{k^*}, an attacker has to recombine all these shares by performing a high-order attack. By choosing a generative approach, cVAE-OSM is able to extract from traces leakage model and variance and consequently to characterize $\mathbb{P}(\mathbf{T}, Y^{k^*})$ in an interpretable way. Hence, to properly



(a) $\hat{\Psi}_\phi$ extracted by E_ϕ **(b)** $\hat{\Psi}_\theta$ extracted by D_θ

Figure 6: ASCAD v1-F – Visualization of the leakage models $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ for all samples extracted by cVAE-OSM, excluding the bias.

Table 9: ASCAD v1 – Average minimum traces to key disclosure.

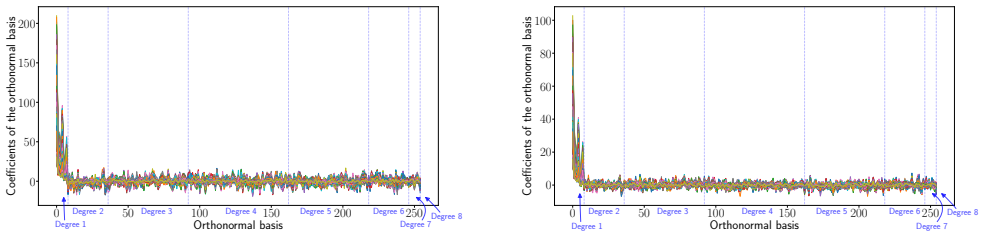
	Template attacks	Stochastic attacks	CNN	cVAE-SA [ZBC+23]	cVAE-OSM (This work)
ASCAD v1-F	327	247	87 [PWP22]	194	201 (E_ϕ and D_θ)
ASCAD v1-R	1,020	988	78 [WPP22]	250	300 (E_ϕ and D_θ)

perform high-order attacks without loss of model interpretability, we can consider two ways to perform a high-order attack: the attacker can either train several models *i.e.* one model for each share and recombine models' output probabilities using convolution product [LPR+14, Section 6; MCLS22, Section 2.1] or he can train a single model by preprocessing traces beforehand. This preprocessing consists in combining together through a certain recombination technique samples related to the processing of all the shares. In this way, these reconstructed traces correspond to the processing of "unmasked" values Y^{k^*}. Several recombination techniques have been introduced in SCA literature such as product combining [CJRR99], absolute difference combining [Mes00] and optimal product combining, also known as centered product [PRB09]. In our case, we consider the second attack path[c]. Therefore, we perform a preliminary step to preprocess traces for both **ASCAD v1** datasets. This preprocessing consists in combining 13 samples related to each share processing, *i.e.* masked value $SBox(p_3 \oplus k^*) \oplus r_3$ and mask r_3, using optimal product combining technique [PRB09]. This results in reconstructed traces of $D = 169$ samples which we will use for profiling and attack.

Configuration. Regarding **ASCAD v1-F** (resp. **ASCAD v1-R**) dataset, we apply on traces maximum absolute value scaling (resp. no scaling). Then, we train our model with $50,000$ (resp. $200,000$) profiling traces of dimension $D = 169$ for 200 (resp. 10) epochs using Adam optimizer, a batch size and a learning rate set to 8 (resp. 2) and 0.001 (resp. 0.01). Once training is done, we visualize in Figure 6 (resp. Figure 7) deterministic parts and consider bits' interaction at most of degree 2. Based on this subset of bits' interaction, we then select 13 PoIs using the sum method to carry out the attack. Results obtained by averaging 10 independent shuffled experiments for **ASCAD v1-F** and **ASCAD v1-R** datasets are summarized in Table 9.

Results. As for **DPA contest v4** and **AES_HD_Ext** datasets, cVAE-OSM seems to perform similarly whether considering the deterministic part extracted by E_ϕ or D_θ *i.e.* it converges towards a rank of 1 for 201 traces (see Table 9). Same observations can be made for **ASCAD v1-R** dataset, the choice of deterministic part during the attack phase does not impact the attack. The model reaches a rank of 1 for about 300 traces.

[c]This choice is purely arbitrary; the first attack path is also totally feasible.



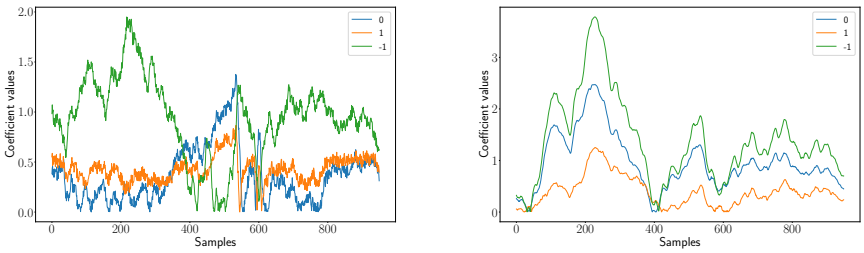
(a) $\hat{\Psi}_\phi$ extracted by E_ϕ **(b)** $\hat{\Psi}_\theta$ extracted by D_θ

Figure 7: ASCAD v1-R – Visualization of the leakage models $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ for all samples extracted by cVAE-OSM, excluding the bias.

Table 9 reports number of traces required to achieve a constant rank of 1 for both **ASCAD v1** datasets when considering different types of attacks, from conventional profiled attacks to DLSCA. Results of conventional profiled attacks obtained for **ASCAD v1-F** and **ASCAD v1-R** datasets are found by averaging 10 template and stochastic attacks on the 13 samples that maximize the SNR of the same recombined traces as those used for cVAE-OSM (see paragraph Configuration). As for previous attacked datasets, we perform stochastic attacks using Guilley *et al.* basis. As we can see, our model performs for both datasets better than conventional profiled attacks. Indeed, for **ASCAD v1-F** dataset, the number of required traces to reach a constant rank of 1 is 201 traces for our model whereas it amounts to 327 and 247 traces respectively for template and stochastic attacks. For **ASCAD v1-R** dataset, our model needs about 300 traces to reach a rank of 1 while template and stochastic attacks need respectively 1,020 and 988 traces. The same explanations as those given in Section 5.2.3 could be envisaged for this dataset. Hence, these new results support those obtained for previous datasets and seem to confirm that cVAE-OSM can outperform conventional profiled attacks as no information loss occurs during leakage model and variance estimations. cVAE-OSM is therefore a valuable alternative to them.

Regarding DLSCA, we can see that our model performs similarly as state-of-the-art generative models: we note a difference of traces number to reach a constant rank of 1 of 7 and 50 traces respectively for **ASCAD v1-F** and **ASCAD v1-R** datasets. This slight difference can be explained by several facts. First, we assume that by performing the attack on traces with a reduced dimension namely $D = 64$ (combination of 8 samples of each share using optimal combining method) for both datasets, they reduce the impact of noisy samples and thus reach more quickly a constant rank of 1. Moreover, as for **DPA contest v4** dataset, our attack is performed with fewer constraints than in [ZBC+23]. Indeed, they need to generate $N_z = 1,000$ synthetic traces to correctly approximate $\mathbb{P}(\mathbf{T}|P, k)$, which is not our case.

However, regarding discriminative models, we remark that CNNs reach better performance than cVAE-OSM. Indeed CNNs retrieve secret key within 87 and 78 traces for **ASCAD v1-F** and **ASCAD v1-R** datasets respectively whereas our model retrieves the secret key within 201 and 300 traces respectively. These results are explained by the fact that discriminative approaches, which directly estimate $\mathbb{P}(Y^{k^*}|\mathbf{T})$ by automatically recombining samples, converge to lower asymptotic errors than generative models [NJ01]. Consequently, when dealing with a masked implementation an attacker should consider a discriminative model to achieve the best possible performance since it can automatically design a custom sample recombination function adapted to the traces considered. However, the achievement of such a performance is mitigated by the lack of interpretability of the model and also by the number of parameters to be considered to both design a non-



(a) $\hat{\Psi}_\phi$ extracted by E_ϕ **(b)** $\hat{\Psi}_\theta$ extracted by D_θ

Figure 8: SCANTRU – Visualization of the leakage models $\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ for all samples extracted by cVAE-OSM.

deterministic architecture not easily justifiable (number of layers, type of layers, activation function...) and train the model (loss, learning rate, batch size...) to ensure a good approximation of $\mathbb{P}(Y^{k^*}|\mathbf{T})$. Hence, our model can be considered as a good alternative to these models since it provides a good trade-off between performance and interpretability.

5.2.5 Results obtained on NTRU dataset

Configuration. By choosing this dataset, we aim to assess cVAE-OSM ability to also attack implementations of asymmetric cryptography algorithms. To this end, we train our model on $50,700$ profiling traces of $D = 950$ samples, on which we apply bias suppression preprocessing method. Then, we train our model for 10 epochs using Adam optimizer and set respectively the batch size and learning rate to 2 and 0.0001. As the number of classes is equal to 3 and the associated values are encoded on a maximum of 11 bits, Guilley *et al.* orthonormal basis cannot be used due to the lack of representation of the remaining $2,045$ classes. We therefore train the neural network by applying one-hot encoding to labels. Once cVAE-OSM is trained, we decide to divide the attack into two stages as in [BEM+23] but using a single model. First, we want to retrieve the -1 coefficients because class -1 is more discriminating than other classes and then, discriminate 1 from 0 coefficients. To ease the readability of this process, interested readers can refer to the SNR computation that has been conducted on this dataset in [BEM+23, Figure 3]. Consequently, the PoI selection for the two stages differ. To select the first set of PoIs, we use the following method: we select samples which maximize the ratio between the value of the coefficient associated with class -1 and the sum of the value of coefficients associated with all classes. Hence, it should be easier to discriminate the -1 class by focusing on these samples. Regarding the second subset of PoIs, we apply the following technique: we compute for each sample the squared Euclidean distance between coefficients associated with classes 0 and 1. Then, we select samples that maximize this distance. Roughly speaking, with this selection method, we aim at picking samples for which the difference in behavior in terms of electromagnetic radiation emitted for classes 0 and 1 is the greatest. For both subsets, we select 35 PoIs and compute success rates for each of the 3 classes. The results presented in the following are obtained by averaging 100 independent shuffled experiments.

Results. Table 10 reports for all three coefficients the number of traces required to achieve a success rate of 100% for both conventional profiled attacks and DLSCA[d]. Interestingly, as we can see, a successful attack is achieved within about 460 traces regardless of the deterministic part considered. These results are surprising since as we can see in Figure 8,

[d] To the best of our knowledge, our work is the first profiled DLSCA on this dataset.

Table 10: SCANTRU – Average minimum traces to achieve a success rate of 100%.

	Template attacks	Stochastic attacks	cVAE-OSM (This work)
Coefficient 0	86	157	324 (E_ϕ) / 186 (D_θ)
Coefficient 1	3,131	484	460 (E_ϕ and D_θ)
Coefficient −1	28	33	14 (E_ϕ) / 114 (D_θ)

$\hat{\Psi}_\phi$ and $\hat{\Psi}_\theta$ are quite different, although they seem to grant large coefficient values to the same samples, which may explain why attack performance are similar.

Regarding stochastic and templates attacks, the results obtained in Table 10 for both attacks are found by averaging 100 attacks on the 35 samples with the highest SNR value. Stochastic attacks are performed using one-hot encoding. To lead to a successful attack, template attack for coefficient 1 are done using the 10 samples that maximize the SNR. As we can see in Table 10, our model achieve better performance than conventional profiled attacks. Indeed, the secret key is recovered within 460 traces whereas template and stochastic attacks achieve a successful attack within respectively $3,131$ and 484 traces. By taking a closer look at the success rates found for each coefficient, we can see that, as expected for the three attacks, coefficient -1 is retrieved with few traces. We remark that the attack requiring the fewest traces is the one using our model. Regarding coefficient 0 and 1, as expected, the discrimination is more difficult (see [BEM+23, Figure 3]) for all three attacks. Indeed, although for all three attacks, coefficient 0 is quickly found, the performance of the attack is degraded by the fact that coefficient 1 is difficult to predict, meaning that the model is biased (a high number of traces is required to ensure that coefficient 1 is not misclassified as 0). As we can see, here again, cVAE-OSM is the fastest to retrieve coefficient 1, enabling a successful attack. Indeed, it requires 460 traces to correctly predict 1 while stochastic and template attacks need respectively $3,131$ and 484 traces. The same explanations as those provided for **AES_HD_Ext** dataset to justify why cVAE-OSM outperforms conventional profiled attacks can be applied here too. Hence, through this experiment, we assess and validate cVAE-OSM relevance for attacking public key algorithm.

5.3 Conclusion on experimental investigations and discussions

Through this section, we experimentally assess theoretical claims made on cVAE-OSM in Section 3. As presented in Section 5.1 and Section 5.2, experimental part on simulations and various datasets validates all the theoretical results highlighted in previous sections. In particular, we experimentally verify on simulations that cVAE-OSM estimation of the deterministic part and variance converge well towards the true deterministic part and variance of traces even when considering a wrong basis size. Moreover, through simulations, we validate that cVAE-OSM overcomes HDLSS/SSS problem and that the dimensionality reduction made by our model is performed without information loss. Regarding attacks on publicly available datasets, results obtained corroborate the observations achieved with simulations. More specifically, these experiments allow us to assess the versatility of our model. Based on the results, it appears that cVAE-OSM is suitable for attacking different types of cryptographic algorithms. Interestingly, we remark that with regards to conventional profiled SCA, our model achieves better performance. Such out-performances compared with conventional attacks are not expected, as in the case of a white Gaussian noise, cVAE-OSM and conventional attacks correspond both to a Gaussian naive Bayes estimator. We assume that this gain in performance stems from leakage models and variances estimation process. Indeed, we assume that cVAE-OSM model provides the most accurate coefficients estimation possible, because the entire trace is considered in the estimation process, meaning that there is no information loss. By contrast, this is not necessarily the case with conventional profiled attacks as a PoIs selection is done

Table 11: Comparison of cVAE-SA [ZBC+23] and cVAE-OSM models regarding architecture ($O(D^2)$ and $O(D)$ respectively) and attack ($O(N_z)$ and $O(1)$ respectively) complexities.

	DPA contest v4		**AES_HD**[1]**/AES_HD_Ext**		**ASCAD v1-F** and **v1-R**		**SCANTRU**	
	cVAE-SA [ZBC+23]	cVAE-OSM (This work)	cVAE-SA [ZBC+23]	cVAE-OSM (This work)	cVAE-SA [ZBC+23]	cVAE-OSM (This work)	cVAE-SA [ZBC+23]	cVAE-OSM (This work)
Trainable parameters	$34,064,000$	$2,052,000$	$3,770,000$	$641,250$	$144,326$	$86,697$	$1,814,500$	$6,650$
Attack complexity[2]	$100 \times \lvert\mathcal{K}\rvert \times Q$	$\lvert\mathcal{K}\rvert \times Q$	$1,000 \times \lvert\mathcal{K}\rvert \times Q$	$\lvert\mathcal{K}\rvert \times Q$	$1,000 \times \lvert\mathcal{K}\rvert \times Q$	$\lvert\mathcal{K}\rvert \times Q$	–	$\lvert\mathcal{K}\rvert \times Q$

[1] Dataset considered in [ZBC+23].
[2] Number of operations required to retrieve the correct key $k^* \in \mathcal{K}$ considering Q attack traces.

beforehand.

In addition, through these experiments, we also validate that our model performs just as well as state-of-the-art DLSCA generative models namely cVAE-SA [ZBC+23], albeit with a much less restrictive attack configuration. In particular, our model addresses several cVAE-SA limitations: the number of parameters is reduced, no preprocessing step is required and the attack is free from the generation of N_z synthetic traces to reach a good rank convergence. Table 11 summarizes architecture and attack complexity for cVAE-SA and cVAE-OSM models considering whole traces. As we can see, for an identical configuration *i.e.* same input traces dimension, our model leads always to a better architecture complexity since $D(1 + 2\sum_{i=0}^{d_{\max}} \binom{S}{i})$ trainable parameters are required (see Proposition 1) against $2D((D+1) + 1 + \sum_{i=0}^{d_{\max}} \binom{S}{i})$ for cVAE-SA model [ZBC+23]. Consequently, due to the huge number of trainable paramaters, a preprocessing step that consists in a PoIs selection is necessary when using cVAE-SA to reduce its complexity (see previous sections). This preprocessing step can be problematic from an attack performance perspective as it may induce information loss during estimations of deterministic part and variance of traces. In contrast, this is not the case with our model since it considers whole traces and performs automatically the PoIs selection with a reasonable number of trainable parameters. Finally, regarding attack complexity, cVAE-OSM leads to a faster key recovery time consumption than cVAE-SA [ZBC+23] as the attack strategy proposed in Section 4.2 does not require to generate traces to correctly approximate $\mathbb{P}(\mathbf{T}|P,k)$ for a given plaintext P and key hypothesis $k \in \mathcal{K}$. Indeed, when using cVAE-OSM, $\mathbb{P}(\mathbf{T}|P,k)$ is approximated by directly computing for all key hypotheses and attack traces the formula presented in Theorem 2 (see Section 4.2) using leakage model and variance estimated by the model during training. This is not the case when considering cVAE-SA since for each key hypothesis $k \in \mathcal{K}$ and attack trace tested, the evaluator/attacker must deal with an additional hyperparameter to correctly approximate $\mathbb{P}(\mathbf{T}|P,k)$ namely N_z, the number of synthetic traces to generate. This implies to perform $N_z \times |\mathcal{K}| \times Q$ operations to retrieve the correct key $k^* \in \mathcal{K}$.

Finally, regarding DLSCA discriminative models, we observe through our experiments that as expected, these models can lead to better attack performances compared to cVAE-OSM [NJ01]. In particular, for masking implementations, discriminative models seem more appropriate from an attack perspective. However, although they offer better performances, their lack of explainability and interpretability and the difficulty of designing their architectures can represent real limitations for an attacker/evaluator. This is not the case for cVAE-OSM, as it has a deterministic architecture and is both fully interpretable and explicable. Consequently, it represents an interesting alternative to these models as it provides a good trade-off between performance and explainability/interpretability. However, some limitations of our model compared to DLSCA discriminative models still remain and are worth pointing out. The first limitation concerns traces desynchronization handling. Indeed, as cVAE-OSM requires synchronized traces, a preliminary preprocessing step is necessary to correctly use our model. There is no such limitation for DLSCA discriminative models as desynchronization is automatically handled by the model itself.

Moreover, unlike DLSCA discriminative models, cVAE-OSM cannot automatically defeat masking countermeasures as it directly estimates $\mathbb{P}(\mathbf{T}|P,k)$. This consequently leads an attacker/evaluator to either perform a preprocessing step on traces or recombine output probabilities in order to cancel masking countermeasure as in Section 5.2.4. Hence, future works should be geared towards developing a solution that will enable cVAE-OSM to automatically manage both desynchronization and masking countermeasures, while maintaining its interpretability and explainability. Finally, another limitation of our model concerns generative models approach itself. In contrast to discriminative DLSCA models that directly estimate $\mathbb{P}(P,k|\mathbf{T})$, generative models estimate $\mathbb{P}(\mathbf{T}|P,k)$ and $\mathbb{P}(P,k|\mathbf{T})$ is then retrieved using Bayes' theorem. To both correctly estimate $\mathbb{P}(\mathbf{T}|P,k)$ and gain in interpretability/explainability, an assumption on the noise distribution contained in $\mathbf{T}$ is done (Gaussian noise assumption) in cVAE-OSM. This consequently restricts the genericity of our model. Future works could thus focus on relaxing the Gaussian noise assumption, allowing for a more generic DLSCA generative model.

6 Conclusion

In this paper, we address a major limitation of state-of-the-art generative models in DLSCA, namely the complexity of the architecture [ZBC+23] by incorporating in our proposed generative model an optimal dimensionality reduction *i.e.* without loss of information. To do so, we propose a dimensionality reduction that derives from [BGH+15] and allows our neural network to remain interpretable and explicable. Based on this dimensionality reduction, we propose a suitable deterministic architecture and design a customized loss which ensures that all architecture weights converge towards interpretable and predefined values, leading cVAE-OSM to converge towards the optimal statistical model and thus to carry out optimal attacks [HRG14]. Indeed, when assessing our neural network with a wide range of real use-cases (symmetric and asymmetric/post-quantum cryptography implementations), we observe that our model performs better than conventional profiled attacks. Throughout this paper, we see that generative models are increasingly seen as a serious, interesting and reliable alternative to discriminative models in DLSCA, as they can be leveraged to develop a fully end-to-end interpretable and explainable deep learning model for SCA, with a reasonable architecture complexity. However, they still have numerous limitations, especially when it comes to dealing with desynchronization and mounting higher-order attacks. Hence, it could be interesting to evaluate the impact of adding convolution or Long Short-Term Memory (LSTM) layers in the presence of desynchronized traces. Other possibilities include the design of a loss that can manage desynchronization and/or masking countermeasure.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments which helped to improve this work.

References

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever,

Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[AARR03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The em side—channel(s). In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 29–45, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[ALMZ19] Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6109–6119, 2019.

[BBD+14] Shivam Bhasin, Nicolas Bruneau, Jean-Luc Danger, Sylvain Guilley, and Zakaria Najm. Analysis and improvements of the dpa contest v4 implementation. In Rajat Subhra Chakraborty, Vashek Matyas, and Patrick Schaumont, editors, *Security, Privacy, and Applied Cryptography Engineering*, pages 201–218, Cham, 2014. Springer International Publishing.

[Beh18] Eric Beh. Exploring how to simply approximate the p-value of a chi-squared statistic. *Austrian Journal of Statistics*, 47(3):63–75, May 2018.

[BEM+23] Luk Bettale, Julien Eynard, Simon Montoya, Guénaël Renault, and Rémi Strullu. Security assessment of ntru against non-profiled sca. In Ileana Buhan and Tobias Schneider, editors, *Smart Card Research and Advanced Applications*, pages 248–268, Cham, 2023. Springer International Publishing.

[BGH+15] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is more - dimensionality reduction from a theoretical perspective. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 22–41. Springer, 2015.

[BHH78] G.E.P. Box, W.G. Hunter, and J.S. Hunter. *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building.* Wiley Series in Probability and Statistics. Wiley, 1978.

[BHM+19] Olivier Bronchain, Julien M. Hendrickx, Clément Massart, Alex Olshevsky, and François-Xavier Standaert. Leakage certification revisited: Bounding model errors in side-channel security evaluations. In *Advances in Cryptology – CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I*, page 713–737, Berlin, Heidelberg, 2019. Springer-Verlag.

[Bis98] Christopher M. Bishop. *Latent Variable Models*, pages 371–403. Springer Netherlands, Dordrecht, 1998.

[BKG20] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *CoRR*, abs/2003.05991, 2020.

[BL04] Peter J. Bickel and Elizaveta Levina. Some theory for Fisher's linear discriminant function, 'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989 – 1010, 2004.

[Bot22] Cristian-Alexandru Botocan. One network to rule them all. an autoencoder approach to encode datasets. Cryptology ePrint Archive, Paper 2022/890, 2022. https://eprint.iacr.org/2022/890.

[BPS+20] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. Deep learning for side-channel analysis and introduction to ascad database. *Journal of Cryptographic Engineering*, 10(2):163–188, 2020.

[BWS16] Pratik Prabhanjan Brahma, Dapeng Wu, and Yiyuan She. Why deep learning works: A manifold disentanglement perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10):1997–2008, 2016.

[C+15] François Chollet et al. Keras. https://keras.io, 2015.

[Car10] Claude Carlet. *Boolean Functions for Cryptography and Error-Correcting Codes*, page 257–397. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2010.

[CDH+19] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M Schanck, Peter Schwabe, William Whyte, and Zhenfei Zhang. Ntru: algorithm specifications and supporting documentation (2019). *URL: https://csrc. nist. gov/projects/post-quantum-cryptography/round-2-submissions. Citations in this document*, 1, 2019.

[CDP15] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Enhancing dimensionality reduction methods for side-channel attacks. In Naofumi Homma and Marcel Medwed, editors, *Smart Card Research and Advanced Applications - 14th International Conference, CARDIS 2015, Bochum, Germany, November 4-6, 2015. Revised Selected Papers*, volume 9514 of *Lecture Notes in Computer Science*, pages 15–33. Springer, 2015.

[CDP17] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Convolutional neural networks with data augmentation against jitter-based countermeasures. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, pages 45–68, Cham, 2017. Springer International Publishing.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[CK13] Omar Choudary and Markus G. Kuhn. Efficient template attacks. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 253–270. Springer, 2013.

[CRR03] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 13–28, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[Doe21] Carl Doersch. Tutorial on variational autoencoders, 2021.

[Don00] David Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 01 2000.

[DR02] Joan Daemen and Vincent Rijmen. *The design of Rijndael*, volume 2. Springer, 2002.

[Fis36] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, September 1936.

[Fuk90] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition (2nd Ed.)*. Academic Press Professional, Inc., USA, 1990.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[GHMR17] Sylvain Guilley, Annelie Heuser, Tang Ming, and Olivier Rioul. Stochastic side-channel leakage analysis via orthonormal decomposition. In *Innovative Security Solutions for Information Technology and Communications: 10th International Conference, SecITC 2017, Bucharest, Romania, June 8–9, 2017, Revised Selected Papers 10*, pages 12–27. Springer, 2017.

[GK22] Philipp Grohs and Gitta Kutyniok. *Mathematical aspects of deep learning*. Cambridge University Press, 2022.

[Goo17] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2017.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *International algorithmic number theory symposium*, pages 267–288. Springer, 1998.

[HRG14] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good is not good enough - deriving optimal distinguishers from communication theory. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 55–74. Springer, 2014.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[IUH22] Akira Ito, Rei Ueno, and Naofumi Homma. Perceived information revisited: New metrics to evaluate success rate of side-channel attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):228–254, Aug. 2022.

[IUH24] Akira Ito, Rei Ueno, and Naofumi Homma. Perceived information revisited ii: Information-theoretical analysis of deep-learning based side-channel attacks. *Cryptology ePrint Archive*, 2024.

[Jeb03] T. Jebara. *Machine Learning: Discriminative and Generative*. The Springer International Series in Engineering and Computer Science. Springer US, 2003.

[JSLH22] Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8, 01 2022.

[JWHT13] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R.* Springer Texts in Statistics. Springer, 2013.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[KKH21] Donggeun Kwon, Heeseok Kim, and Seokhie Hong. Non-profiled deep learning-based side-channel preprocessing with autoencoders. *IEEE Access*, 9:57692–57703, 2021.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. pages 104–113, 1996. URL: `http://www.cryptography.com/timingattack/paper.html`.

[KPH+19] Jaehun Kim, Stjepan Picek, Annelie Heuser, Shivam Bhasin, and Alan Hanjalic. Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):148–179, 2019.

[KW14] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[Led01] Michel Ledoux. *The concentration of measure phenomenon.* Number 89. American Mathematical Soc., 2001.

[LPR+14] Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems – CHES 2014*, pages 35–54, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[LYW+22] Qi Lei, Zijia Yang, Qin Wang, Yaoling Ding, Zhe Ma, and An Wang. Autoencoder assist: An efficient profiling attack on high-dimensional datasets. In Cristina Alcaraz, Liqun Chen, Shujun Li, and Pierangela Samarati, editors, *Information and Communications Security - 24th International Conference, ICICS 2022, Canterbury, UK, September 5-8, 2022, Proceedings*, volume 13407 of *Lecture Notes in Computer Science*, pages 324–341. Springer, 2022.

[Mat02] Jiri Matousek. *Lectures on Discrete Geometry.* Springer-Verlag, Berlin, Heidelberg, 2002.

[MCLS22] Loïc Masure, Valence Cristiani, Maxime Lecomte, and François-Xavier Standaert. Don't learn what you already know: Scheme-aware modeling for profiling side-channel analysis against masking. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(1):32–59, Nov. 2022.

[MDP19] Loïc Masure, Cécile Dumas, and Emmanuel Prouff. A comprehensive study of deep learning for side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(1):348–375, Nov. 2019.

[Mes00] Thomas S. Messerges. Using second-order power analysis to attack dpa resistant software. In *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.

[MF19] Mohammad Mahmud and Xianghua Fu. Unsupervised classification of high-dimension and low-sample data with variational autoencoder based dimensionality reduction. pages 498–503, 07 2019.

[MFHM19] Mohammad Mahmud, Xianghua Fu, Joshua Huang, and Md Masud. *High-Dimensional Limited-Sample Biomedical Data Classification Using Variational Autoencoder: 16th Australasian Conference, AusDM 2018, Bahrurst, NSW, Australia, November 28–30, 2018, Revised Selected Papers*, pages 30–42. 02 2019.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering*, pages 3–26, Cham, 2016. Springer International Publishing.

[MTA07] J.S. Marron, Michael Todd, and Jeongyoun Ahn. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102:1267–1271, 02 2007.

[NJ01] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[OGRW11] Hayrettin Okut, Daniel Gianola, Guilherme JM Rosa, and Kent A Weigel. Prediction of body mass index in mice using dense molecular markers and a regularized neural network. *Genetics research*, 93(3):189–201, 2011.

[OWR+13] Hayrettin Okut, Xiao-Liao Wu, Guilherme JM Rosa, Stewart Bauck, Brent W Woodward, Robert D Schnabel, Jeremy F Taylor, and Daniel Gianola. Predicting expected progeny difference for marbling score in angus cattle using artificial neural networks and bayesian regression models. *Genetics Selection Evolution*, 45(1):1–13, 2013.

[PBA21] Servio Paguada, Lejla Batina, and Igor Armendariz. Toward practical autoencoder-based side-channel analysis evaluations. *Comput. Networks*, 196:108230, 2021.

[PHJ+18] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(1):209–237, Nov. 2018.

[PPM+23] Stjepan Picek, Guilherme Perin, Luca Mariot, Lichao Wu, and Lejla Batina. Sok: Deep learning-based physical side-channel analysis. *ACM Comput. Surv.*, 55(11):227:1–227:35, 2023.

[PRB09] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Transactions on computers*, 58(6):799–811, 2009.

[PWP22] Guilherme Perin, Lichao Wu, and Stjepan Picek. Exploring feature selection scenarios for deep learning-based side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):828–861, 2022.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986.

[RPP22] Azade Rezaeezade, Guilherme Perin, and Stjepan Picek. To overfit, or not to overfit: Improving the performance of deep learning-based SCA. In Lejla Batina and Joan Daemen, editors, *Progress in Cryptology - AFRICACRYPT 2022: 13th International Conference on Cryptology in Africa, AFRICACRYPT 2022, Fes, Morocco, July 18-20, 2022, Proceedings*, volume 13503 of *Lecture Notes in Computer Science*, pages 397–421. Springer Nature Switzerland, 2022.

[RSVC+11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*, pages 109–128. Springer, 2011.

[SA08] François-Xavier Standaert and Cédric Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2008.

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, pages 30–46, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3483–3491, 2015.

[Smi46] Cedric A. B. Smith. Some examples of discrimination. *Annals of Eugenics*, 13(1):272–282, January 1946.

[SMY09] François-Xavier Standaert, Tal G Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28*, pages 443–461. Springer, 2009.

[Tal96] Michel Talagrand. A new look at independence. *The Annals of Probability*, 24(1):1 – 34, 1996.

[TGIH17] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI communications*, 30(2):169–190, 2017.

[TP08] Research Group of Telecom ParisTech. Dpa contest. https://dpacontest.telecom-paris.fr/home/, 2008.

[Ver04] Michel Verleysen. Learning high-dimensional data. *Limitations and Future Trends in Neural Computation*, 2, 02 2004.

[WP20] Lichao Wu and Stjepan Picek. Remove some noise: On pre-processing of side-channel measurements with autoencoders. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):389–415, 2020.

[WPP22] Lichao Wu, Guilherme Perin, and Stjepan Picek. The best of two worlds: Deep learning-assisted template attack. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(3):413–437, 2022.

[Yu20] Ronald Yu. A tutorial on vaes: From bayes' rule to lossless compression. *CoRR*, abs/2006.10273, 2020.

[ZBC+23] Gabriel Zaid, Lilian Bossuet, Mathieu Carbone, Amaury Habrard, and Alexandre Venelli. Conditional variational autoencoder based on stochastic attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(2):310–357, 2023.

[ZBD+20] Gabriel Zaid, Lilian Bossuet, François Dassance, Amaury Habrard, and Alexandre Venelli. Ranking loss: Maximizing the success rate in deep learning side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(1):25–55, Dec. 2020.

[ZBHV20] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Methodology for efficient CNN architectures in profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):1–36, 2020.

[ZZN+20] Jiajia Zhang, Mengce Zheng, Jiehui Nan, Honggang Hu, and Nenghai Yu. A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(3):73–96, Jun. 2020.