arXiv:2301.10802v1 [cs.NE] 25 Jan 2023

NASCTY：利用神经进化攻击侧信道泄漏以生成卷积神经网络

Fiske Schijlen[1],Lichao Wu[1],和 Luca Mariot[2]

[1]代尔夫特理工大学，Mekelweg 5，2628 CD，代尔夫特，荷兰，
fiske.schijlen@hotmail.com, l.wu-4@tudelft.nl[2]语义学、网络安全与服务（SCS），
特文特大学，Drienerlolaan 5 ，
7522NB 恩斯赫德，荷兰，
l.mariot@utwente.nl

2023年1月27日

摘要

侧信道分析（SCA）可以通过利用设备产生的泄漏来获取与密钥相关的信息。研究人员最近发现，神经网络（NN）能够执行强大的剖析式侧信道分析，即使目标设备采取了对策也不例外。本文探讨了利用神经进化攻击侧信道迹线并生成卷积神经网络（NASCTY-CNN）的有效性。这是一种新颖的遗传算法方法，通过对架构超参数应用遗传算子，自动生成用于侧信道分析的卷积神经网络。结果表明，对于采用掩码防护的去同步化泄漏，我们能够达到接近最先进方法的性能，这说明类似的神经进化方法为进一步研究提供了坚实的途径。最后，构建出的神经网络之间的共性揭示了 NASCTY 如何构建有效架构并应对所采用的对策。

关键词 侧信道分析、遗传算法、神经网络、神经架构搜索

1 引言

密码算法是现代生活中无处不在的一部分，因为它们使我们能够保护敏感数据的机密性和完整性。然而，这类算法的实现（即使其在数学上是安全的）有时仍会泄露有关安全资产的信息，例如通过功耗 [10] 或电磁辐射 [16, 12]。攻击者可以针对泄漏进行侧信道分析 （SCA），利用这些泄漏来获取密钥或其部分内容。

假设攻击者拥有一台与目标设备完全相同的设备副本，剖析侧信道分析便成为最强大的攻击方法之一。这种攻击利用副本生成的功耗轨迹，构建一个用于剖析与

密钥相关中间数据或密钥本身相对应的泄漏模式的模型。随后，剖析模型可以从目标设备生成的迹线中恢复密钥。如今，神经网络（NN）凭借强大的攻击能力，已成为最受欢迎的剖析模型选项之一，即使泄漏迹线受到对策保护也是如此 [3, 8, 26, 28, 24]。这种攻击通常称为基于深度学习的侧信道分析（DL-SCA） [15]。

在实践中，应用 DL-SCA 面临的最大障碍之一，是神经网络架构的设计及其超参数的优化。神经网络的 架构指其内部组成部分，例如神经元及其之间的连接。在侧信道分析研究中，神经网络架构往往是通过经验决定的，因此即使使用相同的数据集，也会得到不同的架构 [3, 28, 24]。事实上，面对数量庞大的超参数组合，要找到最优架构并非易事。更糟的是，所选架构在攻击不同数据集或不同实现时可能无法迁移。因此，如果能有一种成熟且自动化的方法，为给定数据集构建适用于侧信道分析的架构，将会大有帮助。虽然也有其他自动设计侧信道分析神经网络的方法，但它们也各有问题。例如，Rijsdijk 等人采用强化学习，生成了性能出色的神经网络，但作者仍需从对待设计架构的一般性描述着手 [18]。此外，强化学习方法计算成本高昂，需要使用 GPU 集群，并花费数天进行调优。另一方面，Wu等人使用贝叶斯优化来寻找用于侧信道分析的神经网络架构 [25]。这种方法比强化学习快得多，同时能取得相近的结果（就攻击性能而言）。不过，作者仍需为贝叶斯优化选择代理模型和采集函数，这可能再次显著增加超参数调优的难度。此外，上述两种方法都依赖于从迭代中积累的经验，而“所选模型是否为全局最优？”这个问题很难回答。

本文提出一种遗传算法（GA），作为上述方法的替代方案，用于 DL-SCA 中的超参数调优任务。一般而言，遗传算法是一种用途广泛的超参数调优元启发式算法，可对候选超参数向量（在遗传算法术语中称为个体）种群进行优化。遗传算法通过应用重组和变异等遗传算子、使用选择方法筛选种群，并根据适应度函数评估个体，来模拟这些向量的自然进化。该过程经过多代迭代后，将最后一代中表现最佳的超参数向量作为解决方案。原则上，遗传算法能够为从不同密码实现中采集的泄漏构建稳健模型。

本研究的主要贡献如下：

1. 我们提出一种基于遗传算法的方法，用于调优剖析侧信道分析中神经网络的超参数。我们的方法可自动运行、易于扩展，并能够生成多种神经网络。
2. 我们分析了使用该方法构建的高性能架构的组成部分，从而深入揭示卷积神经网络超参数选项对侧信道分析的有效性。

本文其余部分的组织如下：第2节介绍剖析式侧信道分析、神经网络、遗传算法以及我们所使用的数据集。第3节概述相关工作。第4节详细介绍我们提出的新方法。第5节详细介绍实验设置，并

报告所得结果。第6节讨论所获得的架构，最后在第7节总结全文。

2 背景

本节介绍与侧信道分析、神经网络和遗传算法相关的所有必要背景概念，这些概念构成了我们工作的基础。由于全面概述这些主题显然超出了本文的范围，因此有必要在此进行介绍。读者可在 Picek 等人近期发表的知识系统化论文 [15] 中找到更多信息。

2.1 剖析侧信道分析

剖析 攻击由剖析阶段和攻击阶段组成，分别类似于监督学习机器学习中的训练阶段和测试阶段。在剖析阶段，攻击者利用克隆设备的泄漏构建模型，以映射泄漏与相应标签（即密钥相关中间数据）之间的关系。在攻击阶段，攻击者遍历所有可能的密钥候选值，并获取其标签对应的输出概率。对每条功耗轨迹重复这一过程，并将分配给每个密钥候选值的概率取对数后求和，即可得到一个对数概率向量，用于确定每个候选值为正确密钥的可能性。式（1）给出了计算密钥候选值k' 在N 条攻击轨迹上的对数概率的过程：

$$P_{log}(k') = \sum_{i=0}^{N-1} \log(P(l(p_i, k'))), \tag{1}$$

其中，$l(\cdot)$表示生成目标中间数据的密码运算，$P(\cdot)$是剖析模型赋予的概率，而 p_i 表示用于泄漏分析的明文i。

攻击性能通过密钥排名 指标评估，如下所示：

$$\mathsf{KR} = \left|\{k' | P_{\log}(k') > P_{\log}(k)\}\right|. \tag{2}$$

直观地说，密钥排名是指正确密钥值的正确可能性低于它的密钥候选值数量。当正确密钥以最高可能性被预测出来，或者其排名进入少数几个可能性最高的候选值之列、可通过穷举攻击找到时，攻击即为成功。在本文中，我们讨论多次实验运行所获得的平均密钥排名；在这种情况下，该指标通常称为猜测熵 [20]。此外，与相关工作中的常见做法一样，我们只评估针对单个密钥字节的攻击性能（这称为部分猜测熵）；为简便起见，我们仍将其称为猜测熵。一个常见假设是，攻击单个密钥字节也能揭示攻击其他密钥字节所需的平均工作量 [27, 18]。

2.1.1 对策

侧信道分析对策旨在减轻密码运算过程中产生的信息泄漏。在本研究中，我们考虑了布尔掩码和去同步化对策。掩码对策将敏感中间数据

值拆分为不同份额，以降低密钥依赖性 [12]。例如，在所考虑的数据集 [3], 中，按照实现方式，在高级加密标准加密第一轮的加轮密钥操作后应用随机掩码r。随后，中间值计算如下：

$$Z = \mathsf{SBOX}(p \oplus k) \oplus r. \tag{3}$$

去同步化是一种常见的隐藏对策，会在泄漏中引入时间随机性。在实践中，可以通过加入时钟抖动和插入随机指令来实现这类效果。我们通过对每条功耗轨迹进行随机移位来模拟这种效果，移位上限为 [3, 26]。

2.2 神经网络

侧信道分析可视为一项分类任务，旨在将输入泄漏映射到与目标标签相对应的类别。这项任务可以通过神经网络 （NN）来完成。神经网络本质上是由神经元层组成的非线性函数，这些神经元有时也称为节点。神经元的输出定义如下：

$$y = \phi(\sum_{i=i}^{n} w_i x_i + b_k), \tag{4}$$

该值的计算方式是：将前一层中输入该神经元的 $x1,\ldots,x_n$ 与各自对应的权重 $w1,\ldots,w_n$ 相乘，加上与神经元 k 对应的偏置 值 b_k，最后用激活函数 ϕ对结果进行变换。激活函数是非线性的来源，通常能够提高训练阶段的效率。在侧信道分析中，神经网络常用的两种激活函数是修正线性单元 （ReLU）和缩放指数线性单元 （SELU）。

训练神经网络时，会通过梯度下降更新每个神经元的权重和偏置，以最小化损失函数。在多分类问题中，常见的损失函数是分类交叉熵（CCE）。交叉熵用于衡量两个分布之间的差异。最小化类别真实分布与神经网络建模的分布之间的交叉熵，可以提高其预测效果：

$$CCE(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{l}\sum_{j=1}^{c} y_{i,j} \cdot \log(\widehat{y_{i,j}}), \tag{5}$$

其中c 和l 分别表示类别数和数据量，y 为真实值，$\hat{}y$为预测值。

在侧信道分析中，神经网络架构的一种主要类型是多层感知机（MLP），其中，一系列由神经元构成的全连接隐藏层之后接一个输出层，该输出层将最终输出值转换为标签预测概率。卷积神经网络（CNN）是侧信道分析中另一种常用的网络类型。它在其第一个全连接层之前置入一个或多个卷积块。此类卷积块由一个卷积层构成，该层尝试在输入数据上计算局部特征，并且其后可选地接一个池化层，用于聚合所得数值，例如通过计算n个数值的平均值。公式（6）形式化地展示了如何将核大小为k的j个卷积滤波器应用于输入 $\{x_i, x_{i+1}, \ldots, x_{i+j}\}$。卷积层重复执行此类卷积，直至遍历全部$n$ 个输入，从而得到$n \cdot j$ 个输入

供全连接层使用。形式化地，该操作可表述如下：

$$\begin{aligned}\mathsf{conv}(\{x_i, x_{i+1}, \ldots, x_{i+j}\}) = \{&c_{0,0} \cdot c_{0,1} \cdot \ldots \cdot c_{0,k-1} \cdot x_i, \\ &c_{1,0} \cdot c_{1,1} \cdot \ldots \cdot c_{1,k-1} \cdot x_{i+1}, \\ &\ldots, \\ &c_{j-1,0} \cdot c_{j-1,1} \cdot \ldots \cdot c_{j-1,k-1} \cdot x_{i+j}\}.\end{aligned} \tag{6}$$

2.3 遗传算法

遗传算法（GA）是一种基于种群的优化算法，通常会借鉴生物进化中的元素 [6]。遗传算法的目标是通过维护一组此类解，并经过若干代的演化来优化某个问题的解。我们将这样的解称为个体 或基因组，它由称为基因的基本构件组成。每一代都通过评估每个基因组的适应度、选择适应度较高的基因组作为繁殖的父代，并对这些父代应用变异和交叉等遗传算子来生成子代；子代构成下一代。

在开始第一代之前，会随机初始化种群中的基因组，以增加多样性。随后不断迭代演化，直到适应度评估预算耗尽，或最佳基因组的适应度值达到预设阈值。每一代都从适应度评估开始：为每个基因组赋予一个适应度值，以衡量对应个体在相关优化问题上的表现。下一步是选择，旨在从种群中淘汰较弱的基因组，使算法倾向于保留能够产生适应度更高基因组的遗传变异。现代遗传算法并非简单地直接选出若干适应度最高的基因组，而是采用更复杂的方法来保持种群的多样性。其中一种方法是锦标赛选择 [13], ，通过从随机选取的一些基因组中举行“锦标赛”，并保留适应度最高者作为父代，以此确定亲本。

选定父代后，遗传算法通常会生成与父代数量相同的子代个体。生成一个子代的基因组，需要对一个或多个父代应用一个或多个遗传算子。这些算子包括变异和交叉 [7], ，不过在非交配遗传算法中可以省略后者。变异只需要一个亲本：将其克隆，并使用预先定义的变异函数进行随机修改，从而生成一个子代。另一方面，交叉是指组合两个或多个亲本的属性以构造一个子代。在本研究中，我们采用多项式变异，这是一种针对实值参数的变异方法，由 Deb 和 Agrawal [4] 提出，适用于具有预定义最小值和最大值边界的变量。该方法以相同的概率将某个变量x 向较低边界x_L或较高边界x_U 变异。变异幅度随后由伪随机数 $0 \leq u < 1$ 和参数 η决定， η 的值越大，变异范围越小。换言之，变异后的值x' 等于$\bar{x} + \bar{\delta}_L (x - x_L)$ 或$\bar{x} + \bar{\delta}_R (x_R - x)$，其中 $\bar{\delta}_L$和 $\bar{\delta}_R$随u和 η 变化，具体定义见公式（7）。

$$\begin{aligned}\bar{\delta}_L &= (2u)^{\frac{1}{1+\eta}} - 1 \\ \bar{\delta}_R &= 1 - (2(1-u))^{\frac{1}{1+\eta}}.\end{aligned} \tag{7}$$

神经进化是指使用进化算法来构建或优化神经网络。在本研究中，我们将使用遗传算法为

侧信道分析构建神经网络架构。在这种情况下，遗传算法中的基因组描述了神经网络的超参数组合。适应度由对网络性能的评估决定。

2.4 数据集

我们使用 ASCAD 数据集 [3], ，其中每条功耗轨迹包含 700 个轨迹点，对应于第三个密钥字节的 S 盒操作。请注意，我们指的是固定密钥 ASCAD 数据集，其中所有 AES 操作都使用相同的加密密钥。这些功耗轨迹受到掩码 对策的保护，因此中间值Z 按照公式 (3) 使用某个随机掩码字节r计算得到。

训练集（35 584 条）和验证集（3 840 条）是从 50 000 条训练功耗轨迹中抽取的均衡样本。选择这两个集合的样本数量，是为了确保它们分别满足各自用途所需的规模。由于我们在使用此方法的所有实验中都采用恒等泄漏模型，因此两个集合的样本数量都是 256 的倍数，即可能的输出标签数量。请注意，虽然我们预计恒等泄漏模型和类别不平衡 [14],不会带来太多问题，但我们仍对类别进行均衡，以减轻任何不良影响。我们使用一万条攻击轨迹来评估攻击性能。最后，我们在该数据集上分别针对启用和未启用去同步对策的情况开展实验。

3 相关工作

现在，我们概述有关针对侧信道分析领域优化神经网络架构的文献，涵盖手动和全自动方法。接下来，我们简要回顾基于神经进化的自动化方法。

3.1 侧信道分析中的网络架构优化

2016 年，Maghrebi 等人提出了几种基于深度学习的方法，并在 DPAv2 数据集 [23]以及采用和未采用一阶掩码的定制 AES 实现上验证了这些方法的有效性 [11]。在采用掩码的实现上，定制神经网络仅用不到 10^3 条训练功耗轨迹便取得了零密钥排名。有趣的是，他们使用遗传算法确定卷积神经网络架构，并将猜测熵作为适应度函数，但并未对其方法进行深入阐述。

Benadjila 等人 [3] 进一步探索了神经网络在侧信道分析领域中的性能。评估了现有文献中若干有前景的架构后，他们认为 VGG-16 [19] 或结构相似的深度卷积神经网络非常适用于侧信道分析。Kim 等人 [8] 结合数据增强提升了攻击性能；事实证明，这是一种有效的方法，能够让卷积神经网络适应多种对策。随后，Zaid 等人提出了高效的卷积神经网络架构 [28] ，在显著缩小神经网络规模的同时实现了最先进的性能。Wouters 等人 [24] 则通过数据预处理策略进一步缩小了网络规模。

除了手动优化神经网络外，近期研究还尝试了完全自动化的网络架构搜索方法。Rijsdijk 等人 [18] 针对侧信道分析定制了 MetaQNN 强化学习算法，以自动寻找卷积神经网络架构。然而，其搜索空间大致局限于我们已知有效的超参数，并未讨论纯多层感知机架构。在他们的工作中，每个神经网络

都使用 Adam 优化器和 SELU 激活函数训练 50 个训练轮次后进行评估。
Wu等人提出了 AutoSCA [25], , 该方法使用 贝叶斯优化 来寻找多层感知机和卷积神经网络的架构超参数。他们的方法取得了良好效果，且主要侧重于寻找更大的架构，使每个全连接层至少包含 100 个神经元。

现有研究对将神经进化应用于侧信道分析的探索仍然很少。Knezevic等人利用遗传编程进化出专用于侧信道分析的自定义激活函数 [9] ，其性能优于广泛使用的 ReLU函数。他们的方法将激活函数编码为一棵树状基因组，其中包含一元和二元运算符，叶节点表示函数的输入。树的初始深度为两到五层，进化过程中深度限制为十二层。进行适应度评估时，他们计算一百折中达到密钥排名为零所需攻击轨迹数量的平均值，并将其加到一减去准确率的结果上。该方法生成了新颖的激活函数，提升了大型、高效多层感知机和卷积神经网络的性能。Acharya等人提出了 InfoNEAT [1], ，这是一种专为侧信道分析定制的增强拓扑神经进化（NEAT）算法 [21] 。他们的方法采用恒等泄漏模型，并使用 NEAT 进化神经网络架构，为 256 个输出类别分别设置一个输出节点。随后，通过堆叠 方法组合得到的 256 个二元网络，将这些网络的输出作为逻辑回归模型的输入。针对从完整数据集中抽取的多个不同折的平衡迹线，分别创建此类堆叠模型；之后，将这些模型的预测概率相加，即可为每条攻击轨迹生成最终预测。

3.2 基于进化的网络架构搜索

进化方法已广泛用于自动化网络架构搜索。Real等人提出了一种用于现代数据集图像分类的方法 [17], , 这项任务需要大型网络。他们提出了一种非交配遗传算法，结合类似 NEAT 的 [21] 变异和层级变异，以细粒度和大规模的方式进化卷积神经网络。具体而言，每个基因组先使用 45 000 个样本进行反向传播训练，再评估其适应度。在这种方法中，子代的基因组保留亲本的权重和偏置，从而使每个网络都能随着时间推移持续训练。

其他成功用于构建图像分类卷积神经网络的神经进化方法还包括深度进化网络结构化表示方法 DENSER [2] 和 EvoCNN [22]。DENSER 使用两级基因型：第一级编码神经网络的超参数，第二级编码层特定变量，例如神经元数量或卷积滤波器的变量。只要能在基因组的第二级中对其进行适当定义，这种结构就能使该算法适用于多层感知机、卷积神经网络及其他类型的网络。此外，DENSER 会先通过反向传播训练神经网络，再在验证集上评估适应度。EvoCNN 的工作方式类似，但会将权重初始化值与架构超参数一同进化。

4 NASCTY

通过神经进化攻击侧信道迹线以生成卷积神经网络（NASCTY-CNN）是一种遗传算法，可调整卷积神经网络的超参数以进行侧信道分析。算法1展示了我们方法的主要流程。本节将

算法1NASCTY-CNN 算法

1：训练_迹线，训练_标签，验证_迹线，验证_标签 ← 采样（*ascad*_数据） *2*：种群 ← 初始化_种群*() 3*：当代数 <最大值_代数时循环 *4*：评估_适应度_值（种群，训练_迹线，训练_标签，训练*p*明文） *5*：亲本 ← 锦标赛_选择（种群） *6*：子 代 ← 生成_子代（亲本） *7*：种群 ← 亲本 ∪ 子 代 *8*：结束循环 *9*：返回种群中适应度最低的基因组

说明基因组结构、种群初始化、适应度评估方法，以及用于生成子代的方法。请注意，训练数据和验证数据只采样一次，这意味着每一世代的适应度评估都使用相同的数据。此外，我们只使用均衡数据样本，即抽取的迹线集合中，对应每个可能输出标签的迹线数量相同。遵循 [17, 2, 22], ，我们还在选择过程中使用大小为 3 的锦标赛，这意味着我们随机选取三个个体，并选择其中适应度最高的个体作为繁殖的潜在亲本。最后，请注意，所有实验均针对固定密钥 ASCAD 数据集中的第三个（掩蔽）密钥字节进行；其中每条迹线包含 700 个轨迹点，每个轨迹点都归一化到 -1 至 1 之间。

4.1 基因组结构

NASCTY 基因组表示一个卷积神经网络，由以下部分组成：一个包含零至五个卷积块的列表；当不存在卷积块时，一个可选的池化层；以及一个包含一至五个全连接层的列表。每个卷积块均由卷积滤波器数量、滤波器尺寸、一个表示是否包含批归一化层的布尔值，以及一个池化层来描述。基因组中的任何池化层均包含池化类型（最大池化或平均池化）、池化尺寸和池化步幅。最后，全连接层仅通过其神经元数量来描述。图 1 展示了基因组结构示例。

将 NASCTY 基因组表示为神经网络时，遵循最先进的架构，我们始终对所有隐藏神经元使用 SELU 激活函数，对卷积块和全连接层使用 He 权重初始化，并对使用 Softmax 激活的输出层使用 Glorot 均匀权重初始化

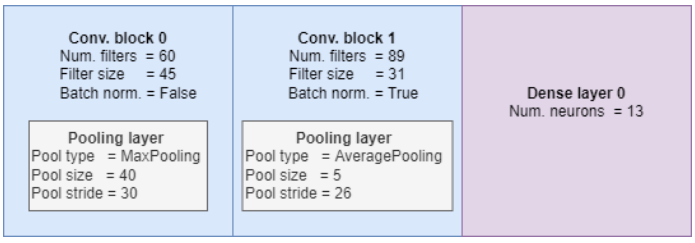


图 1：一个 NASC 中使用的基因组编码示例 TY 算法。

参数	选项
卷积块数量	0 到 5，步长为 1
全连接层数量	1 到 5，步长为 1
卷积滤波器数量	2 至 128，步长为 1
滤波器尺寸	1 至 50，步长为 1
批归一化层	False，True
池化类型	平均、最大值
池化尺寸	2 到 50，步长为 1
池化步幅	2 到 50，步长为 1
稠密神经元数量	1 到 20，步长为 1

表 1：NASCTY 算法中卷积神经网络基因组超参数的范围。

函数。尽管扩大基因组参数空间会增加种群的多样性，但应用先验知识可以加快进化过程。

4.2 种群初始化

我们在表 1 所示的范围内随机初始化所有网络的超参数。这些超参数范围以及基因组结构本身，均参考了先前研究中用于侧信道分析的类 VGG 网络 [3, 8], , 以及基于强化学习的侧信道分析自动化架构搜索方法 [18]。

请注意，可以从参数数量最少的架构开始初始化种群，以缩短所需的评估时间。不过，我们选择完全随机地初始化种群，以避免因种群多样性降低而可能陷入局部最优。

4.3 适应度评估

定义基因组后，使用 Adam 优化器训练相应的卷积神经网络。[1]验证集上的损失值用于适应度评估。通过最小化损失，我们旨在使系统与深度学习侧信道分析领域的相关工作保持一致。当然，也可以考虑其他选项，例如 [25] 中采用的选项。评估网络之前先对其进行训练，目的是更准确地区分其质量。与其他工作一样，我们选择对每个网络训练十个训练轮次 [25, 2, 22]，而且根据 [22] 的方法建议开展的初步实验表明，对于规模相近的网络，这已足以观察到不同质量网络之间显著的 CCE 差异。

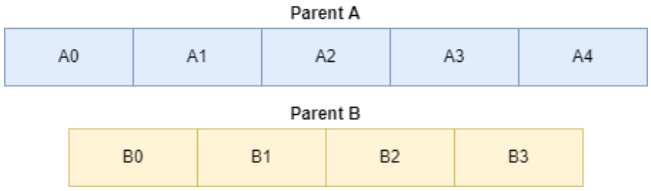
4.4 子代生成

评估完某一世代中每个基因组的适应度后，即可产生下一世代的成员（子代）。其中一半通过对种群进行锦标赛选择，找出适应度高、可作为亲本的基因组。

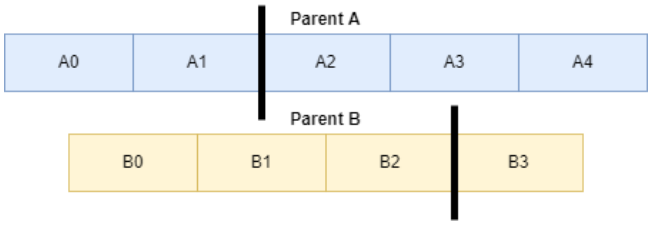
[1]为确保公平比较，每一世代中的所有网络都使用相同的随机种子进行训练。

另一半则通过随机挑选这些亲本配对，并对其进行交叉和变异操作来产生。我们也可以选择只对种群中表现最好的部分成员进行锦标赛选择，以选出亲本。此操作旨在确保种群中保留最优基因组，这一概念称为精英保留，可通过截断比例参数进行调整。

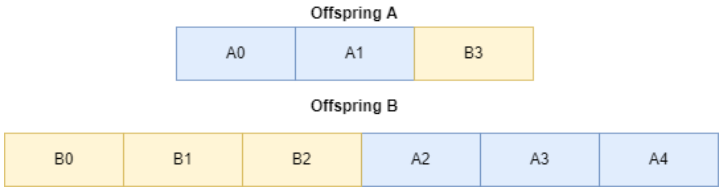
我们的算法使用两种可能的交叉类型之一，即单点交叉或逐参数 交叉，这两种都是遗传算法中常见的交叉策略。我们在第5.2节中评估了这两种方法的性能。为了对两个亲本执行单点交叉，我们分别对亲本的卷积块列表和全连接层列表进行单点交叉。对于卷积块列表或全连接层列表，我们通过在两个亲本相应类型的层列表中随机选取一个截断点来完成此操作。随后，将第一个亲本截断点之前的列表与第二个亲本截断点之后的列表连接起来，生成第一个子代的相应类型列表；将剩余部分连接起来，则生成第二个子代的列表。最后，在不存在卷积块时，将亲本中可选的池化层随机分配给子代，从而完成单点交叉操作。



(a) 每个亲本的层或块列表。



(b) 父代各自的随机截断点。



(c) 交叉后子代中包含的两组层或模块列表。

图2：NASCTY 基因组中卷积模块或全连接层列表进行单点交叉的示意图。

在我们对逐参数交叉的实现中，第一个子代基因组通过遍历父代的卷积块对，并随机继承任一亲本的

卷积块基因来创建。随后，第二个子代的基因组继承这些卷积块剩余的超参数。对于父代的全连接层列表，重复完全相同的过程。在一个亲本拥有比另一个亲本更多的卷积块或全连接层这一典型情况下，多出的单元将未经修改地附加到第一个子代的基因组中。

交叉操作之后，子代以相同的概率通过以下方法之一进行变异：

- 添加一个随机卷积块或全连接层，并为其随机初始化超参数；
- 移除一个随机卷积块或全连接层；
- 以概率 $\frac{1}{n}$通过多项式变异 修改所有超参数，其中n 是基因组中可修改超参数的总数。

多项式变异在第2.3节中有所介绍，适用于具有预定义最小值和最大值边界的变量，因此适合我们的任务：在预定义范围内探索合适的超参数值。此变异方法包含一个参数 η，增大该参数可以降低对给定参数进行修改的程度。建议将 η 的值设为 20 到 100 [5], 之间，但对于未经测试的场景，只有通过实验才能确定 η 的合适值。

5 实验

本节将讨论 NASCTY 的实验评估。首先介绍实验设置，然后展示基于网格搜索的初步调优结果。最后，介绍遗传算法采用表现最佳的参数组合，在 ASCAD 的掩蔽轨迹和去同步轨迹上取得的结果。

5.1 实验设置

为验证我们的方法，我们通过网格搜索优化遗传算法参数，然后评估其在掩蔽的 ASCAD 轨迹，以及不同去同步化级别下掩蔽且去同步的 ASCAD 轨迹上的性能。这些实验旨在确定：

- 遗传算法参数对我们的方法的有效性；
- 我们的自动化方法能否生成性能优于通过试错找到的类似神经网络的神经网络；
- 有助于提升侧信道分析有效性的架构组件。

为考虑变异操作引入的随机性，我们针对每种遗传算法参数配置运行五次实验，并报告最佳结果。对 NASCTY 算法得到的最佳基因组进行评估：训练其对应的神经网络 50 个训练轮次，并计算 [18] 在 100 折上的平均增量密钥排名。

所有实验均使用 52 个并行工作进程执行，每个进程在 Intel E5-2683 v4 CPU 上以约 2.1GHz 的频率运行。借助这些计算资源，

参数	选项
多项式变异 η	20, 40
交叉类型	单点交叉、逐参数
截断比例	0.5, 1.0

表 2：网格搜索 NASCTY 时所考虑的参数值。

上述实验需要 84GB 内存，完成时间最短为四天，最长为七天。运行时间复杂度存在如此显著的差异，是由遗传算法的伪随机性造成的，这导致构建和评估的神经网络大小各不相同。

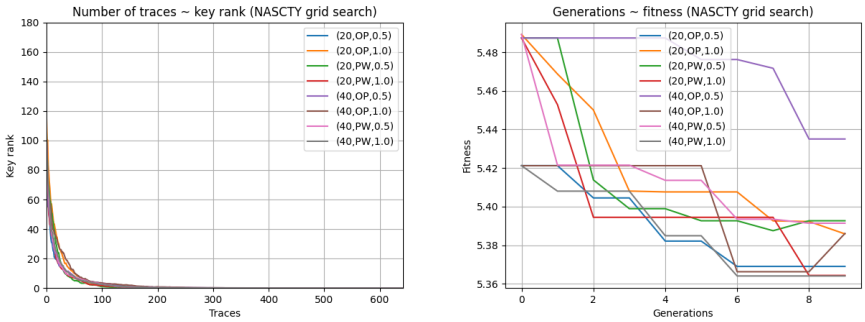
5.2 通过网格搜索调优参数

表 2 汇总了网格搜索所用的遗传算法参数。需要注意的是，增加参数选项、变异策略和交叉策略，可能会带来更好的性能，但若要评估该算法的整体有效性，这些调整就必须与我们当前的策略存在显著差异。每次网格搜索实验中，我们都将种群规模设为 52，以匹配可用并行工作进程的数量，并让遗传算法运行 10 代。此外，为了更准确地观察参数变化的影响，每次实验都使用相同的训练数据、验证数据和初始种群。

我们预计，在这些运行代数相对较少的实验中，较低的多项式变异参数 η [4] 会是更好的选择。具体来说，较小的 η 会导致较大的变异，这有可能更快地找到更好的网络；但 η 取值为 20 仍足够大，可以避免过于冒险的变异 [5]。出于类似原因，较大的截断比例应能带来更好的性能，因为它能在本已较小的种群中保留多样性。不过，与所选的交叉和变异配置相比，其影响可能没有那么显著，因为后两者能更直接地改变种群。最后，我们预计两种交叉策略都能取得良好表现，因为它们都能让算法在预定义的超参数范围内找到有效的架构。图 3 展示了每种参数组合下最佳运行结果所对应的最终网络性能。

图 3a 表明，每种参数组合都能够针对所考虑的 ASCAD 轨迹生成拟合架构。同样，图 3b 所示的最佳运行在各代中的适应度曲线表明，无论所考虑的参数组合为何，最佳基因组的 CCE，即验证损失，都可以在短短十代内得到显著改善。

所有网格搜索实验中，最佳平均增量密钥秩约为 0.50419，这一结果来自多项式变异 η 值为 20、采用单点交叉且截断比例为 1.0 的实验。因此，后续所有实验均采用这些参数。为确定各遗传算法参数对最终性能的影响，我们在保持其他参数为上述观测到的最佳值不变时，一次只修改一个变量，并观察其影响。表 3 展示了此类修改对平均增量密钥秩的影响。从表中可以推断，在我们考察的参数中，只有交叉策略会显著影响最终性能；与逐参数交叉相比，单点交叉效果更好。由于图 3 中采用逐参数交叉的最佳运行结果仍然表现



(a) 100 折上的最佳运行平均性能 (b) 最佳运行的适应度变化过程（越低越好）

图 3：NASCTY 网格搜索结果，对应于每种参数组合所获得的最佳网络。

良好，因此性能差异很可能源于其结果一致性不如采用单点交叉的运行结果。我们推测，单点交叉所体现出的更高一致性，得益于它更善于保留卷积块或全连接层的功能序列。此外，对层列表执行单点交叉，直观上也与我们通过添加或移除整个层来进行变异的策略相辅相成：当有效的添加或移除操作以模块化方式分配给不同子代时，便能更快地识别出来。

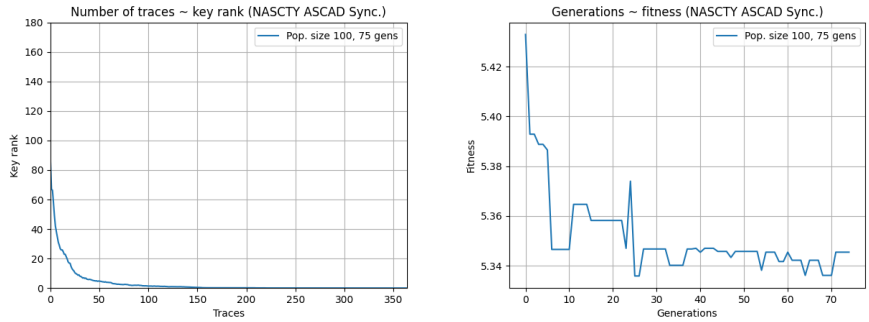
5.3 ASCAD：掩蔽与去同步

其余所有实验均使用通过网格搜索实验找到的性能最佳参数选项进行。此外，我们将这些实验的种群规模设为 100，以充分利用可用资源。与网格搜索实验不同，这些实验不会为 GA 中任何环节涉及的伪随机数设置随机种子，适应度评估过程除外；在该过程中，我们为每个 NN 的训练设置随机种子，以确保公平比较各个基因组。

我们首先在掩蔽的 ASCAD 轨迹上运行 NASCTY 75 代，以评估该算法的整体有效性。然后，我们在同一掩码数据集上进行实验，并按照第 2.1.1 节所述进一步采用去同步化进行保护。具体而言，我们分别在去同步化级别为 10、30 和 50 的情况下进行三组实验，

η	交叉类型	截断比例	平均增量密钥秩
20	单点交叉	1.0	0.50419
40	单点交叉	1.0	0.50880
20	逐参数	1.0	0.89354
20	单点交叉	0.5	0.50966

表 3：NASCTY 网格搜索实验中各参数对最终增量密钥排名的影响。



(a) 最佳神经网络在 100 折中的平均性能　　(b) 最佳运行结果的适应度演进

图 4：在掩蔽的 ASCAD 轨迹上获得的最佳网络所对应的 NASCTY 结果。

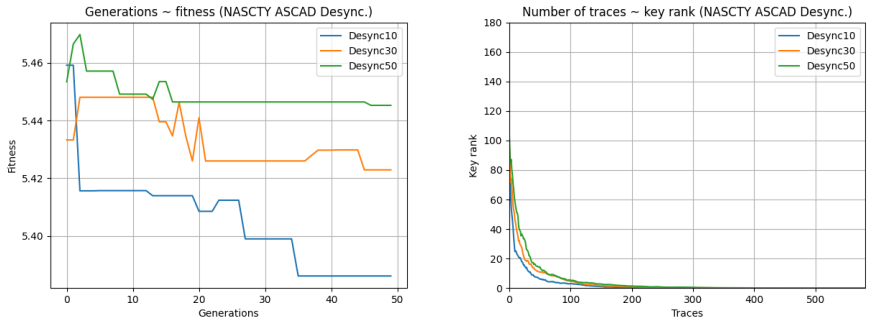
以此确定 NASCTY 能否在不对算法进行额外修改的情况下绕过或缓解这些对策，以及更高的去同步化级别是否会妨碍 NASCTY 找到良好的架构。

图 3b 中的适应度演进趋势表明，增加世代数将提升所观察到的最佳基因组适应度。基于此，我们首先在掩蔽的 ASCAD 轨迹上运行 NASCTY 75 代，种群规模为 100。图 4 展示了此次初始实验的结果。如图 4a 所示，获得的最佳网络平稳收敛。最终，该网络在 314 条攻击轨迹中成功攻破目标，并达到 0.51857 的平均增量密钥秩。最佳适应度值在经过 10 代后持续提升（图 4b），随后在达到第 75 代之前很久便陷入停滞。此外，尽管种群规模和世代数存在差异，最佳网络的性能仍不及我们网格搜索实验所获得的若干神经网络。这种观察到的适应度停滞表明，该算法可能容易陷入局部最优。通常，变异是遗传算法中进行全局搜索的来源，因此我们建议未来的工作评估多项式变异的较低 η 值，或采用扰动性更强的变异策略。

鉴于上一项实验中观察到的适应度进展，我们在经过掩码处理和去同步化的 ASCAD 轨迹上运行 NASCTY 50 代，而非 75 代。去同步化级别为 10、30 和 50 时的结果如图5所示。如图5a所示，去同步化引入的时间随机性会影响算法的性能：随着去同步化级别提高，适应度进展和最终性能均显著下降。尽管如此，结果表明，即便增加了这些对策，NASCTY 仍能找到有效的架构；图5b中评估的网络在去同步化级别分别为 10、30 和 50 时，分别能够在 338、474 或 531 条迹线上达到密钥排名 0。

6 讨论

在同步的 ASCAD 轨迹上表现最佳的一次运行，生成了图6a所示的卷积神经网络架构。该架构有 10470 个可训练参数，依稀类似于 Zaid 等人提出的高效卷积神经网络。相比之下，使用 NASCTY 生成的架构



(a) 最佳运行的适应度变化　　(b) 最佳运行在 100 折中的平均性能

图5：NASCTY 在去同步化级别为 10、30 和 50 的 ASCAD 轨迹上的结果。

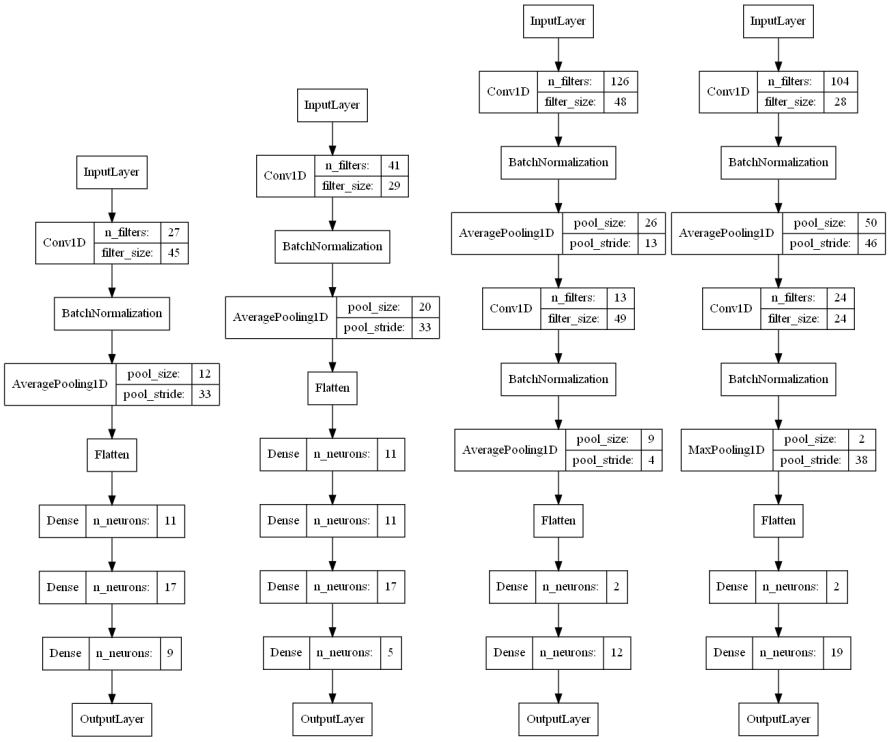
还多了一层全连接层，并包含若干不太直观的组件，例如 27 个尺寸为 45 的卷积滤波器，以及大于池化尺寸的池化步幅。由于 Wouters 等人提出的高效多层感知机仅需两层、每层十个神经元，因此我们推测 NASCTY 可能倾向于纳入不必要的模型复杂度。换言之，NASCTY 对冗余模型复杂度的抑制力度不足。事实上，将去同步化程度从 30 提高到 50 后，最佳架构的可训练参数数量从 90 379 降至 68 427。一种可能的解决方案是在适应度函数中加入模型规模惩罚项。尽管如此，采用当前配置的 NASCTY 足以生成良好的网络架构。表4显示，NASCTY 可与其他最先进的侧信道分析自动化超参数调优方法相媲美，不过最终表现仍逊于它们。

去同步化级别为 10 时对应的架构，在结构和规模上都与同步轨迹对应的架构相似，主要区别在于增加了更多卷积滤波器和另一层全连接层。这两种架构还都包含一个步长大于自身尺寸的池化层，这表明其卷积层提取的特征可能存在冗余或未被正确利用。

针对两个更严重的去同步化级别（30 和 50），NASCTY 架构进一步揭示了 NASCTY 缓解这一对策的方式。如图6c和图6d所示，这两种架构都以一个包含 100 多个滤波器的卷积层开头，滤波器数量显著多于其他架构中的卷积层。此外，两者都包含两个卷积块和两个全连接

方法	**迹线数量至获得平均密钥排名 0**	**可训练参数数量**
RL-SCA [18]	242	1 282
AutoSCA [25]	158	54 752
NASCTY	314	10 470

表4：针对同步的、掩蔽的 ASCAD 轨迹进行侧信道分析的自动化超参数调优方法比较。



(a) 同步　(b) 去同步化 等级 10　(c) 去同步化 等级 30　(d) 去同步化 级别 50

图6：NASCTY 为采用不同去同步化级别保护的掩蔽的 ASCAD 轨迹生成的最佳架构。

层，每个网络中的第一个全连接层都只有两个神经元。滤波器数量较多与现有的去同步化缓解方法一致，但在尝试破解受保护的 ASCAD 轨迹时，使用如此小的全连接层并不常见。最后，平均池化似乎是这些网络的首选池化类型，最大池化仅出现一次。我们强调，针对 50 级去同步化设计的架构，其表现优于 [18]，或与 [28, 24] 中提出的最先进的方法非常接近。

7 结论与未来工作

本文提出了一种用于侧信道分析领域网络架构搜索的遗传算法。在 NASCTY 中，每个基因组编码表示卷积神经网络架构的超参数，并通过验证损失评估基因组的适应度。在产生子代的过程中，我们对父代的层列表进行单点交叉，或采用逐参数交叉，以生成一对子代基因组；随后立即通过添加一层、移除一层，或对其基因应用多项式变异来进行变异。采用这种方法，NASCTY 能够生成与最先进技术相当的架构。NASCTY 的一些网络中存在冗余复杂度和不直观的架构组件，这表明我们的方法很可能还有进一步改进的空间，

这意味着我们仍有尚未探索的潜力，可以达到或超越当前最先进的方法。

此外，NASCTY 在基本不修改其遗传算法参数和实现的情况下，也为采用掩码保护且去同步化级别高达 50 的迹线找到了有效架构。然而，去同步化确实影响了最终网络的性能：NASCTY 为去同步化级别 0、10、30 和 50 生成的网络，分别在 314、338、474 和 531 条攻击轨迹内达到密钥排名 0。我们建议未来的工作评估 NASCTY 对采用其他对策保护的迹线的有效性，以及由此得到的架构模式。观察到的网络架构表明，NASCTY 倾向于通过增加一个卷积层，并增加第一卷积层中的滤波器数量来应对去同步化；去同步化级别为 30 和 50 时生成的架构便是如此。有趣的是，这些网络的全连接部分均以一个包含两个神经元的全连接层开头。此外，一些架构包含步长大于池化层尺寸的池化层，这在其他方法中并不常见，这表明 NASCTY 可能会通过不必要地增加卷积滤波器数量来生成冗余特征。对于同步和轻度去同步的迹线，NASCTY 生成的架构所含可训练参数也多于相关工作中的模型，而后者在同一任务上表现更好 [24], ，这印证了 NASCTY 易于增加冗余模型复杂度这一假设。无论是否存在去同步化，几乎所有池化层都更倾向于采用平均池化而非最大池化。

对于未来的工作，可以探索在适应度评估中引入复杂度惩罚，或在种群初始化时采用最小架构，以缩小网络规模。此外，我们建议尝试更多变异参数，以促进更好的全局搜索，避免陷入局部最优；这种情况似乎往往在算法终止之前就已发生。解决这些缺陷后，我们可以通过扩展搜索空间来发现更出色、更有趣的架构。为此，我们建议在基因组中引入新的超参数，例如每层的激活函数，以及用于更通用应用场景的学习率。最后，我们的方法采用了使用固定密钥的 ASCAD 数据集。未来也值得研究该方法在其他更难的数据集（例如使用随机密钥的 ASCAD 数据集）上的表现。

参考文献

[1] R. Y. Acharya、F. Ganji 和 D. Forte。InfoNEAT：基于信息论的拓扑增广进化方法，用于侧信道分析。*arXiv*预印本 *arXiv:2105.00117*，2021年。

[2] F. Assunção、N. Lourenço、P. Machado 和 B. Ribeiro。DENSER：深度进化网络结构化表示。*Genetic Programming and Evolvable Machines*，20(1)：5–35，2019年。[3] R. Benadjila、E. Prouff、R. Strullu、E. Cagli 和 C. Dumas。深度学习技术在侧信道分析中的研究及 ASCAD数据库简介。*ANSSI，法国 & CEA，LETI，MINATEC*园区，法国。，22：2018，2018年。

[4] K. Deb 和 S. Agrawal。遗传算法中约束处理的小生境惩罚方法。载于 *Artificial Neural Nets and Genetic Algorithms*，第235–243页。施普林格，1999年。

[5] K. Deb 和 D. Deb。实数参数遗传算法变异方案分析。*International Journal of Artificial Intelligence and Soft Computing*，4(1)：1–28，2014年。[6] A. E. Eiben 和 J. E. Smith。进化计算导论。施普林格出版公司，第二版，2015年。[7] J. H. Holland。遗传算法。*Scientific american*，267(1)：66–73，1992年。[8] J. Kim、S. Picek、A. Heuser、S. Bhasin 和 A. Hanjalic。制造一些噪声：释放卷积神经网络在侧信道评估中的威力。*IACR*密码硬件与嵌入式系统汇刊，第148–179页，2019年。[9] K. Knezevic、J. Fulir、D. Jakobovic 和 S. Picek。NeuroSCA：为侧信道分析进化激活函数。*IACR*密码学电子预印本档案，2021：249，2021年。[10] P. C. Kocher、J. Jaffe 和 B. Jun。差分功耗分析。载于第*19*届国际密码学年会密码学进展会议论文集，CRYPTO '99，第388–397页，英国伦敦，1999年。Springer-Verlag。[11] H. Maghrebi、T. Portigliatti 和 E. Prouff。使用深度学习技术破解密码实现。载于国际安全、隐私与应用密码工程会议，第3–26页。施普林格，2016年。[12] S. Mangard、E. Oswald 和 T. Popp。功耗分析攻击：揭示智能卡的秘密。施普林格，2006年12月。[13] B. L. Miller 和 D. E. Goldberg。遗传算法、选择方案与噪声带来的不同影响。*Evol. Comput.*，4(2)：113–131，1996年6月。[14] S. Picek、A. Heuser、A. Jovic、S. Bhasin 和 F. Regazzoni。类别不平衡的危害及机器学习用于侧信道评估时指标间的冲突。*IACR*密码硬件与嵌入式系统汇刊，2019(1)：209–237，2018年11月。[15] S. Picek、G. Perin、L. Mariot、L. Wu 和 L. Batina。综述：基于深度学习的物理侧信道分析。*ACM Comput. Surv.*，2022年10月。已正式接收。[16] J.-J. Quisquater 和 D. Samyde。电磁分析（EMA）：智能卡的测量与对策。载于 I. Attali 和 T. Jensen 编，智能卡编程与安全，第200–210页，柏林，海德堡，2001年。Springer Berlin Heidelberg。[17] E. Real、S. Moore、A. Selle、S. Saxena、Y. L. Suematsu、J. Tan、Q. Le 和 A. Kurakin。图像分类器的大规模进化。*arXiv*预印本 *arXiv:1703.01041*，2017年。[18] J. Rijsdijk、L. Wu、G. Perin 和 S. Picek。强化学习在基于深度学习的侧信道分析超参数调优中的应用。*IACR*密码硬件与嵌入式系统汇刊，2021(3)：677–707，2021年7月。[19] K. Simonyan 和 A. Zisserman。用于大规模图像识别的超深卷积网络。*arXiv*预印本 *arXiv:1409.1556*，2014年。

[20] F.-X. Standaert、T. G. Malkin 和 M. Yung。《侧信道密钥恢复攻击分析的统一框架》。载于 A. Joux（编），密码学进展 - *EUROCRYPT 2009*，第443–461页，柏林、海德堡，2009年。Springer Berlin Heidelberg。[21] K. O. Stanley 和 R. Miikkulainen。通过拓扑增广进化神经网络。进化计算，10(2):99–127，2002年。[22] Y. Sun、B. Xue、M. Zhang 和 G. G. Yen。进化深度卷积神经网络以进行图像分类。*IEEE*进化计算汇刊，24(2):394–407，2019年。[23] TELECOM ParisTech SEN研究组。DPA竞赛（2nd 届），2009–2010年。[24] L. Wouters、V. Arribas、B. Gierlichs 和 B. Preneel。重新审视剖析攻击中高效卷积神经网络架构的方法。*IACR*密码硬件与嵌入式系统汇刊，2020(3):147–168，2020年6月。[25] L. Wu、G. Perin 和 S. Picek。选我吧：针对基于深度学习的侧信道分析的自动化超参数调优。*IACR*密码学电子预印本档案，2020:1293，2020年。[26] L. Wu 和 S. Picek。去除一些噪声：关于使用自编码器对侧信道测量数据进行预处理。*IACR*密码硬件与嵌入式系统汇刊，第389–415页，2020年。[27] L. Wu、Y.-S. Won、D. Jap、G. Perin、S. Bhasin 和 S. Picek。解释一些噪声：基于深度学习的物理侧信道分析的消融分析。密码学电子预印本档案，论文2021/717，2021年。[28] G. Zaid、L. Bossuet、A. Habrard 和 A. Venelli。剖析攻击中高效卷积神经网络架构的方法。*IACR*密码硬件与嵌入式系统汇刊，第1–36页，2020年。