SCOOP : une méthode d'optimisation de réseaux de neurones adaptée aux attaques par canaux auxiliaires

Nathan ROUSSELOT[1,2] Karine HEYDEMANN[1] Loïc MASURE[2] Vincent MIGAIROU[1]

[1]Thales, France

[2]LIRMM, CNRS, Université de Montpellier, 161 rue Ada, 34095 Montpellier

Résumé – Cet article présente un nouvel algorithme d'optimisation pour l'entrainement des réseaux de neurones dans le cadre des attaques par canaux auxiliaires (DL-SCA). Cet algorithme, appelé SCOOP, est une descente de miroir stochastique de second ordre. Nous montrons que SCOOP est capable de repousser significativement les limites actuelles du DL-SCA en présence de masquage. Grâce à SCOOP, nous avons pu mener à bien la première attaque DL-SCA sur le jeu de données ASCADv2.

Abstract – This paper presents a new optimisation algorithm for training neural networks for side-channel attacks (DL-SCA). This algorithm, called SCOOP, is a second-order stochastic mirror descent. We show that SCOOP pushes further the current limits of DL-SCA in the presence of masking. Thanks to SCOOP, we mount the first successful DL-SCA attack on the ASCADv2 dataset.

1 Introduction

Les attaques par canaux auxiliaires (SCA) exploitent la corrélation entre les données manipulées ou les instructions exécutées par un système, et des grandeurs physiques observables comme la consommation électrique ou le temps d'exécution. Cette corrélation permet à un attaquant de déduire des informations sensibles ou secrètes, telles que des clés cryptographiques.

Pour s'en protéger, il existe plusieurs contre-mesures. L'une d'entre elles, le masquage d'ordre d, consiste à diviser les valeurs sensibles en $d+1$ valeurs aléatoires, appelées *shares*, tel que toute combinaison de d ou moins *shares* est statistiquement indépendante du secret. Une donnée sensible n'est jamais manipulée directement ce qui protège donc théoriquement contre les attaques d'ordre d, *i.e* contre les attaques recombinant au plus d échantillons temporels de la trace, telles que les attaques basées sur n'importe quel moment statistique d'ordre au plus d.

Les SCAs par profilage consistent à apprendre un modèle statistique de la fuite sur un clone de l'appareil ciblé, puis à utiliser ce modèle pour récupérer la clé secrète sur la cible. Les attaques par profilage peuvent être formulées comme un problème d'apprentissage automatique. Cela a provoqué un grand engouement pour les attaques SCA à base de réseaux de neurones [5]. Les réseaux de neurones profonds (DNN) sont théoriquement capables de casser tout schéma de masquage d'ordre supérieur. Cependant, malgré de nombreux travaux exploratoires sur les architectures, les DNNs ont toujours des difficultés en présence de ce type de contre-mesures.

Le problème auquel nous nous intéressons consiste à cibler les implémentations masquées sans aucune connaissance préalable autre que l'algorithme ciblé, ce que l'on appelle le cadre *non-worst-case* [6].[1]

Jusqu'à présent, les attaques par canaux auxiliaires basées sur des réseaux de neurones (DL-SCA) se sont inspirées des méthodes d'apprentissage profond communes. Cela implique alors des routines d'entraînement qui sont pensées pour la meilleure généralisation possible, pré-supposant que le problème d'optimisation est trivialement résolu par les méthodes de gradients stochastiques. Cependant, plusieurs travaux rapportent des comportements inattendus lors de la minimisation de la fonction de perte, notamment en présence de masquage [6]. En particulier, on observe une stagnation de cette fonction, le fameux *effet plateau* [6]. La taille de ce plateau, mesurée en epochs, est exponentiellement proportionelle à l'ordre de masquage. Bien que ce phénomène semble être un frein à l'efficacité des DL-SCA, il ne garantit aucunement la sécurité de l'implémentation. Dans le cadre d'une évaluation de sécurité avec un temps limité, cet effet plateau peut donc procurer un faux sentiment de sécurité.

Dans cet article, nous présentons un nouvel algorithme d'optimisation conçu spécifiquement pour les DL-SCA, afin de limiter l'effet plateau. Cet algorithme, appelé SCOOP, est une descente de miroir stochastique parcimonieuse (SMD) de second ordre. Nous montrons que SCOOP est capable de réduire significativement la taille de l'effet plateau, et donc d'améliorer l'efficacité des DL-SCA en présence de masquage.

La suite de cet article est organisée comme suit. Nous commençons par introduire l'effet plateau sur un jeu de données de référence, nous introduisons ensuite SCOOP et ses motivations théoriques. Nous terminons par une évaluation empirique de SCOOP sur des implémentations masquées, dont la première attaque DL-SCA sur le jeu de données ASCADv2 [8].

2 Effet Plateau

Le problème du DL-SCA se distingue du cadre classique de l'apprentissage profond sur plusieurs points. Tout d'abord, seule une fraction de la donnée secrète est disponible dans chaque trace ; par exemple, le poids de Hamming de celle-ci. Par conséquent, si on entraîne un réseau de neurones $\mathbf{F}$ avec un critère de négative log-vraisemblance (NLL), il n'est pas possible d'atteindre une loss de validation nulle (Fig. 1). En particulier, la NLL est une estimation de *l'information perçue* (PI) [7], telle que $\text{PI} = \mathbb{H}[Y] - \text{NLL}$, où $\mathbb{H}[Y]$ est l'entropie de la variable sensible Y. Ainsi, on dit d'un réseau de neurones (pour le SCA) qu'il est *fonctionnel* si $\text{NLL} < \mathbb{H}[Y]$. Ensuite,

[1]Il est également appelé : attaque *black-box* dans la littérature.

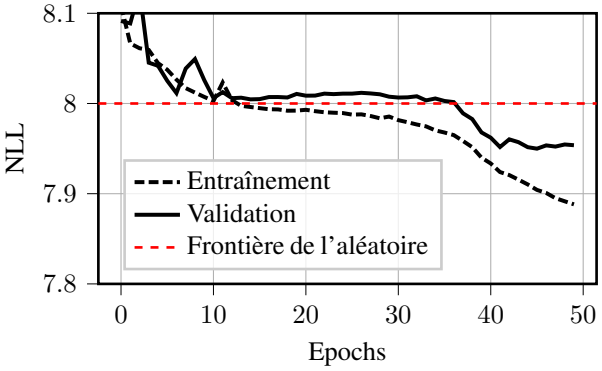


FIGURE 1 : Entraînement d'un MLP sur ASCADv1 avec ADAM.

le masquage d'ordre d introduit une indépendance entre les valeurs sensibles et les valeurs observées, qui se matérialise par l'apparition du fameux effet plateau, ce qui rend le problème d'optimisation plus difficile.

La Figure 1 montre l'évolution de l'entraînement d'un réseau de neurones MLP sur le jeu de données ASCADv1 [1]. Celui-ci contient les signaux d'une implémentation d'un AES masqué à l'ordre 1 sur 8 bits ; et a été extensivement exploré, faisant office de benchmark dans le domaine. On observe sur la Figure 1 que la fonction de perte stagne pendant quelques epochs (35 dans ce cas) avant d'entamer sa descente. Cela signifie que durant 35 epochs, le réseau de neurones n'apprend pas et est, au mieux, un prédicteur aléatoire.

Il a été observé que la longueur de ce plateau croît exponentiellement avec l'ordre de masquage [6]. Avec un masquage d'ordre supérieur, cet effet peut rendre l'évaluation de sécurité d'une implémentation masquée difficile, voire impossible, dans un temps limité.

3 SCOOP : un algorithme d'optimisation pour les DL-SCA

Pour repousser les limites du DL-SCA actuel, nous introduisons SCOOP, un algorithme d'optimisation basé sur la descente de miroir stochastique (SMD) de second ordre. SCOOP est motivé par des développements théoriques montrant que la magnitude et la variance des gradients diminuent exponentiellement avec l'ordre de masquage [9]. La seconde motivation de SCOOP est d'exploiter la parcimonie de l'information d'intérêt dans les données, en conséquence, nous faisons l'hypothèse que le modèle optimal $\mathbf{F}^\star$ est parcimonieux. SCOOP exploite la régularisation implicite de la descente de miroir stochastique [10]. En considérant une fonction potentielle $\psi = \ell_{1+\epsilon}$, où $\epsilon > 0$, on peut montrer que la solution de la descente de miroir stochastique est parcimonieuse. L'équation de mise à jour de Scoop est donnée par l'Equation 1.

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmin}} \eta_t \theta^T \hat{h}^{-1} \nabla_\theta L(\theta) + D_\psi\left(\theta, \theta^{(t)}\right) \qquad (1)$$

où η_t est le taux d'apprentissage à l'itération t, $\hat{h}$ est une estimation de la Hessienne, $D_\psi\left(\theta, \theta^{(t)}\right)$ est la divergence de Bregman entre θ et $\theta^{(t)}$, et $L(\theta)$ est la fonction de perte.

$$D_\psi\left(\theta, \theta^{(t)}\right) = \psi(\theta) - \psi\left(\theta^{(t)}\right) - \left\langle \nabla_\theta \psi\left(\theta^{(t)}\right), \theta - \theta^{(t)} \right\rangle$$

L'algorithme itératif de SCOOP est présenté dans l'Algorithme 1. La Hessienne est approchée par un estimateur de Hutchinson [3] (ligne 4) où v suit une loi de Rademacher mise à l'échelle (c'est à dire $v \in \{-\alpha, \alpha\}$ avec $|\alpha| < 1$). Ensuite, SCOOP calcule les moyennes mobiles exponentielles (EMA) du gradient et de la Hessienne (lignes 5 et 6). Le pas de descente est calculé dans l'espace dual (ligne 7) et projeté dans l'espace primal (ligne 8). La fonction c est une fonction de *clipping* qui stabilise le calcul de l'inversion de la Hessienne [4].

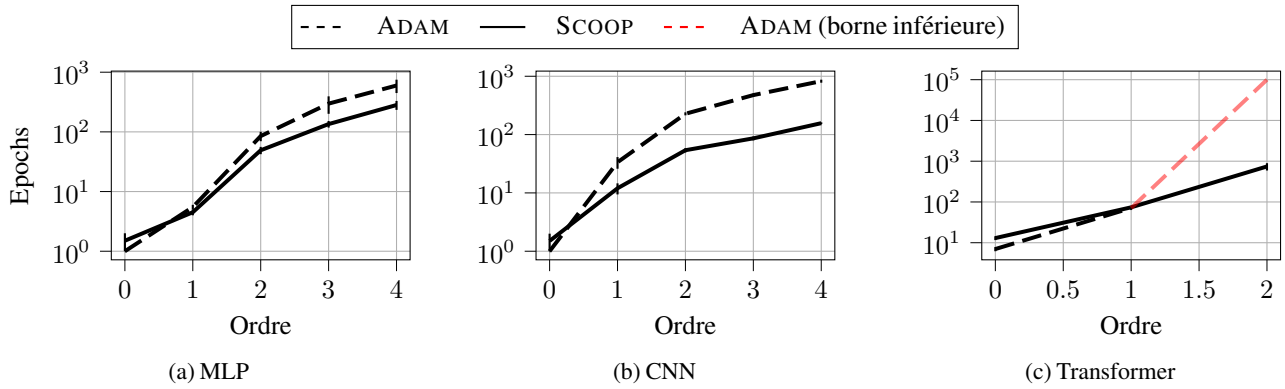
Algorithme 1 : SCOOP : SeCond-Order precOnditioned sParse stochastic mirror descent

1 **for** $t \leftarrow 1$ **to** T **do**
2 Calculer la fonction de perte du mini-batch $L(\theta_t)$;
3 Échantillonner v selon une distribution de Rademacher mise à l'échelle;
4 $\tilde{H}_t \leftarrow v \odot \nabla\left(\langle \nabla L(\theta_t), v \rangle\right)$;
5 $h_{t+1} \leftarrow \beta_2 h_t + (1-\beta_2)\tilde{H}_t$;
6 $g_{t+1} \leftarrow \beta_1 g_t + (1-\beta_1)\nabla L(\theta_t)$;
7 $\text{Step} \leftarrow (1+\epsilon)|\theta_t|^\epsilon \text{sign}(\theta_t) - \eta_t c\left(h_{t+1}^{-1} g_{t+1}\right)$;
8 $\theta_{t+1} \leftarrow \left|\frac{\text{Step}}{(1+\epsilon)}\right|^{1/\epsilon} \text{sign}(\text{Step})$;
9 **end**

4 Application à des données simulées

Pour vérifier la pertinence de SCOOP, nous avons entraîné trois réseaux de neurones (MLP, CNN et Transformer [2]) sur un jeu données simulé avec un masquage d'ordre 0 à 4 et un modèle de fuite poids de Hamming. Les données ne sont pas bruitées et sont codées sur 8 bits. Afin de rendre ce scénario tractable, nous utilisons la forte non-injectivité du poids de Hamming pour calculer efficacement la fonction de perte [6]. Chaque expérience est répétée 100 fois pour minimiser l'erreur expérimentale. Nous comparons l'entraînement de SCOOP avec ADAM. Toutes les expériences sont réalisées avec une NVIDIA RTX4500 Ada Generation. Due a une limitation en puissance et en mémoire GPU, nous avons limité le nombre d'epochs à 10^5 et l'ordre de masquage à 2 pour le Transformer. Les résultats sont présentés sur la Figure 2.

Nous constatons que SCOOP réduit efficacement la longueur du plateau dès qu'une contre-mesure de masquage est activée, ce qui est conforme à nos motivations. La Table 1 montre la réduction moyenne du plateau en utilisant SCOOP par rapport à ADAM. Nous pouvons constater que SCOOP est particulièrement efficace sur les schémas de masquage d'ordre élevé : la longueur du plateau est réduite d'un facteur allant jusqu'à 5 pour le CNN à l'ordre 3 et plus. Il semble que la réduction du plateau soit d'autant plus importante que l'ordre de masquage augmente, et qu'elle soit également sensible à l'architecture du modèle. Parallèlement, nous avons observé une augmentation du temps d'entraînement d'environ 5% seulement en utilisant SCOOP par rapport à ADAM. Enfin, pour le modèle Transformer, ADAM est incapable d'entrainer un modèle fonctionnel dans



(a) MLP (b) CNN (c) Transformer

FIGURE 2 : Comparaison de la longueur du plateau avec ADAM et SCOOP sur un ensemble exhaustif de données simulées pour trois architectures. Pour le Transformer (c), ADAM n'a pas réussi à récupérer le secret dans les 10^5 epochs allouées à l'ordre 2, c'est pourquoi nous traçons une borne inférieure à la place.

TABLE 1 : Réduction moyenne du plateau (en pourcentage) en utilisant SCOOP par rapport à ADAM sur un ensemble exhaustif de données simulées avec MLP et CNN.

Modèle	**n=0**	**n=1**	**n=2**	**n=3**	**n=4**
MLP	−50%	18.18%	42.15%	54.57%	52.89%
CNN	−50%	64.22%	76.27%	81.87%	80.97%

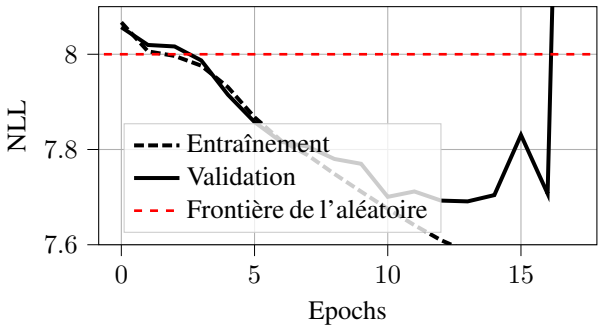


FIGURE 3 : Entraînement d'un MLP sur ASCADv1 avec SCOOP.

les 10^5 epochs allouées à l'ordre 2, tandis que SCOOP a pu le faire en moins de 10^3 epochs (Fig. 2c). Nous remarquons que SCOOP n'apporte pas de gain sur les traces simulées non protégées ($n=0$), ce qui est cohérent avec les motivations de conceptions de SCOOP qui émergent du masquage ($n \geq 1$).

5 Application à des données réelles

5.1 ASCADv1

ASCADv1 [1] est un jeu de données correspondant à une implémentation d'un AES protégé par un masquage booléen d'ordre 1. Dans un premier temps, nous reproduisons l'expérience que nous avons vue précédemment (Fig. 1). Nous remplaçons simplement ADAM par SCOOP et nous observons une réduction significative de la longueur du plateau. En particulier, le plateau est réduit de 35 à 3 epochs (Fig. 3)

Par ailleurs, si on regarde la distribution des poids du réseau de neurones entraîné avec SCOOP et ADAM, on observe que les poids sont plus concentrés autour de zéro avec SCOOP (Fig. 4).

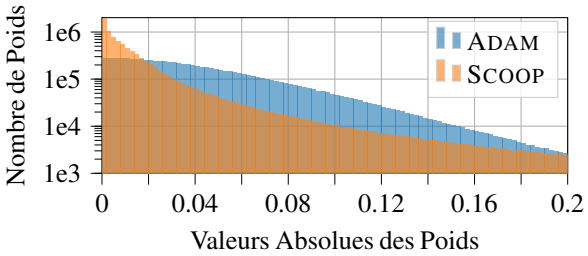


FIGURE 4 : Comparaison des distributions des valeurs absolues des poids du modèle entraîné contre ASCADv1 (orange) : ADAM, (bleu) : SCOOP.

Cela suggère que SCOOP favorise la parcimonie des poids, ce qui est conforme à nos motivations.

5.2 ASCADv2

Un autre jeu de données reconnu dans la communauté SCA est ASCADv2 [8], correspondant à une implémentation logicielle AES à masquage affine (pseudo-second ordre). Elle intègre un mécanisme de *shuffling* de boucle qui consiste à permuter l'ordre dans lequel les octets sont traités dans la boucle du sub-byte. À notre connaissance, la seule attaque concluante sur ASCADv2 exploite une faiblesse émergeant du schéma de masquage affine, à savoir que le masque multiplicatif est partagé entre tous les éléments de l'état AES [11]. Wu *et al.* ont désactivé le mécanisme de shuffling de boucles pour mener à bien leur attaque. Dans notre expérience, nous ne nous appuyons sur aucune hypothèse, le fameux scénario *non-worst-case*.

ASCADv2 est un jeu de données difficile non seulement en raison de ses contre-mesures, mais aussi de sa dimension. Chaque trace a une longueur de 15 000 échantillons, et les modèles entraînés pour s'adapter à des données d'une telle dimension nécessitent une puissance de calcul et une capacité de mémoire considérables.

Étant donnée notre configuration matérielle, nous écartons les architectures coûteuses telles que les CNN et les Transformers. Nous nous concentrons sur les MLP. Compte tenu de la difficulté du jeu de données, nous avons décidé de le diviser en trois parties : un jeu de profilage de 200 000 traces, un jeu de validation de 10 000 traces et un jeu d'attaque de 300 000 traces. Les jeux de profilage et d'attaque proviennent tous deux des « traces

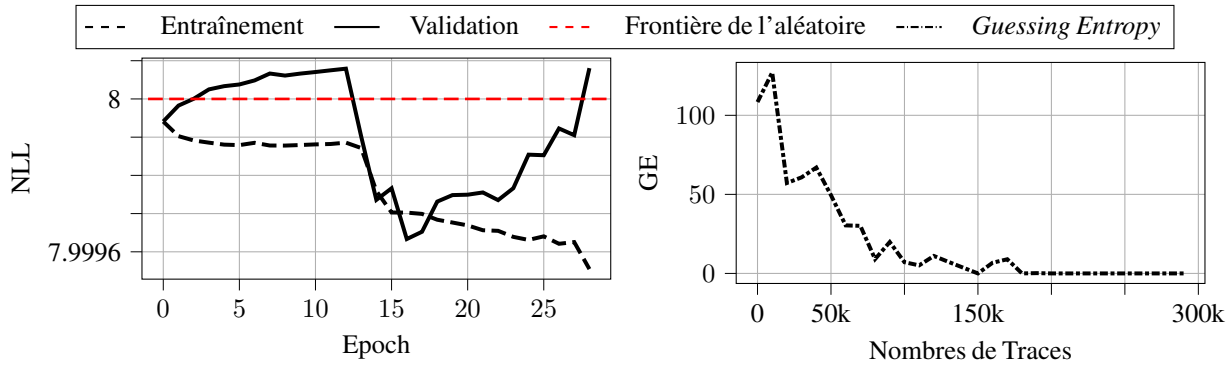


FIGURE 5 : Exemple d'entraînement d'un modèle fonctionnel sur le jeu de données ASCADv2 utilisant SCOOP et la *guessing entropy* correspondante (right).

de profilage » et sont respectivement les 200 000 premières et 300 000 dernières traces du jeu de données. Le jeu de validation est le même que celui fourni par les auteurs d'ASCADv2 [8]. Cela permet de monter une attaque avec un grand nombre de traces, ce qui est nécessaire pour casser le jeu de données.

Notre modèle est un MLP à une seule couche cachée de taille $n_{\text{hidden}} = 2/3 \times (\text{taille de l'entrée} + \text{taille de la sortie})$. Ce modèle a un total d'environ 231 millions de paramètres. Pour mesurer la performance d'une attaque DL-SCA, nous utilisons une métrique appelée *guessing entropy* (GE). La GE est l'espérance du rang du secret dans le vecteur des probabilités retourné par le modèle. Une GE de 0 signifie que le secret est retrouvé avec certitude. Cette mesure est faite avec plus ou moins de traces d'attaque N_a, l'objectif étant d'avoir le plus petit N_a tel que la GE est nulle.

Le MLP à une seule couche cachée permet de réussir une attaque en 10 heures d'entraînement. La Guessing Entropy est nulle après seulement 150 000 traces d'attaque. Parfois, moins de 50 000 traces d'attaque suffisent pour retrouver la clé. Lors de l'entraînement, on observe un plateau de 12 epochs. L'exemple d'entraînement, ainsi que sa GE, sont présentés dans Fig. 5.

6 Conclusion

Les attaques DL-SCA sortent des cadres classiques de l'apprentissage profond, et exhibent des défis nouveaux comme l'*effet plateau*. Devant ce constat, et en se basant sur une étude théorique des modèles DL-SCA [9], nous proposons d'utiliser un nouvel algorithme d'optimisation pour entraîner les modèles DL-SCA : SCOOP. SCOOP est une descente de miroir stochastique de second ordre, qui exploite la parcimonie de l'information d'intérêt dans les données DL-SCA. Nous avons montré que SCOOP est capable de réduire significativement la taille du plateau, et donc d'améliorer l'efficacité des DL-SCA en présence de masquage. Nous avons évalué SCOOP sur des données simulées et réelles, et avons monté la première attaque DL-SCA fructueuse sur ASCADv2.

Références

[1] Ryad BENADJILA, Emmanuel PROUFF, Rémi STRULLU, Eleonora CAGLI et Cécile DUMAS : Deep learning for side-channel analysis and introduction to ASCAD database. 10(2):163–188, juin 2020.

[2] Suvadeep HAJRA, Siddhartha CHOWDHURY et Debdeep MUKHOPADHYAY : EstraNet : An efficient shift-invariant transformer network for side-channel analysis. 2024(1):336–374, 2024.

[3] Michael F HUTCHINSON : A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

[4] Hong LIU, Zhiyuan LI, David HALL, Percy LIANG et Tengyu MA : Sophia : A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv :2305.14342*, 2023.

[5] Houssem MAGHREBI, Thibault PORTIGLIATTI et Emmanuel PROUFF : Breaking cryptographic implementations using deep learning techniques. *In Security, Privacy, and Applied Cryptography Engineering : 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings 6*, pages 3–26. Springer, 2016.

[6] Loïc MASURE, Valence CRISTIANI, Maxime LECOMTE et François-Xavier STANDAERT : Don't learn what you already know scheme-aware modeling for profiling side-channel analysis against masking. 2023(1):32–59, 2023.

[7] Loïc MASURE, Cécile DUMAS et Emmanuel PROUFF : A comprehensive study of deep learning for side-channel analysis. 2020(1):348–375, 2019.

[8] Loïc MASURE et Rémi STRULLU : Side-channel analysis against ANSSI's protected AES implementation on ARM : end-to-end attacks with multi-task learning. 13(2):129–147, juin 2023.

[9] Nathan ROUSSELOT, Karine HEYDEMANN, Loïc MASURE et Vincent MIGAIROU : Scoop : An optimizer for profiling attacks against higher-order masking. *Cryptology ePrint Archive*, 2025.

[10] Haoyuan SUN, Khashayar GATMIRY, Kwangjun AHN et Navid AZIZAN : A unified approach to controlling implicit regularization via mirror descent. *Journal of Machine Learning Research*, 24(393):1–58, 2023.

[11] Lichao WU, Guilherme PERIN et Stjepan PICEK : Not so difficult in the end : Breaking the lookup table-based affine masking scheme. pages 82–96, 2023.