A hitchhiker's guide to white-box neural network watermarking robustness

Carl de Sousa Trias, Mihai P Mitrea, Enzo Tartaglione, Attilio Fiandrotti, Marco Cagnazzo, Sumanta Chaudhuri

A HITCHHIKER'S GUIDE TO WHITE-BOX NEURAL NETWORK WATERMARKING ROBUSTNESS

Carl De Sousa Trias [1], *Mihai Mitrea* [1], *Enzo Tartaglione* [2], *Attilio Fiandrotti* [2,3], Marco Cagnazzo [4], Sumanta Chaudhuri [2]

[1] Télécom SudParis, Institut Polytechnique de Paris, France
[2] Télécom Paris, Institut Polytechnique de Paris, France
[3] Universitá di Torino, Italy
[4] Universitá di Padova, Italy
Email: carl.de-sousa-trias@telecom-sudparis.eu

ABSTRACT

The present study deals with white-box Neural Network (NN) watermarking and focuses on the robustness property. The first contribution consists of formalizing neuron permutation as a geometric attack, thus demonstrating the very existence of this class of attacks for NN watermarking. The second contribution consists in devising and demonstrating the effectiveness of the corresponding counter-attack. As a side result, the possibility of extending NN white-box watermarking scope beyond image classification is brought to light. The experimental study considers three state-of-the-art methods, four NN models, three tasks (image classification, segmentation, and video coding), and five types of attacks. We underline that none of the existing methods is robust against the geometric attack, and using the counter-attack advanced in this paper effectively ensures the robustness.

Index Terms— watermarking, neural network, white-box, robustness, geometric attacks, counter-attack.

I. INTRODUCTION

Neural Networks (NN) are currently serving as enablers for practically all multimedia-related tasks, such as image classification, segmentation [1] or compression [2]. Design, data collection, and training of NN require huge investment, and protecting the underlying intellectual property rights is not only an ethical issue but an economic one, as well. Moreover, such applications can also be deployed in critical contexts (*e.g.* autonomous driving), where it is key to verify that the NN functioning has not been corrupted.

Watermarking represents a promising solution to the above, and potentially other related problems [3], [4]. Watermarking [5] originally refers to *imperceptibly* and *persistently* embedding into multimedia contents some additional information (referred to as *watermark* or *mark*) according to a *secret key*. Inserted by an *authorized user*, the watermark detection is expected to track down an *unauthorized user* that would illicitly benefit from or modify that content.

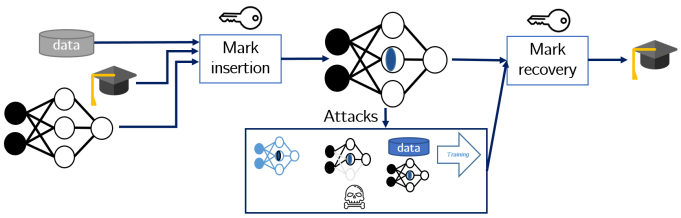

Fig. 1: Neural network watermarking synopsis.

This generic framework inherited from multimedia realm is to be reconsidered and extended to match the NN peculiarities, as detailed here after and illustrated in Fig. 1. First, the watermark is inserted into the NN model (defined as the set of parameters of a neural network, including the input-output functions). The watermark can be either retrieved from the parameters of the model (the so-called *white-box* methods [3], [6], [7]) or from the inference output by the watermarked model (the so-called *black box* ones [4], [8]). The *data payload* represents the size of the watermark, *i.e.* the quantity of information to be inserted and detected.

Second, the *imperceptibility* refers to the impact (if any) of the mark insertion in the task achieved by the NN. For instance, watermarking a NN for image classification is imperceptible when the class score distribution is not modified.

Third, *robustness* is the property of recovering the mark from the protected content even when it is subjected to malicious or mundane operations (commonly referred to as *attacks*).

Finally, the *secret key* refers to the information that should be kept secret and implicitly ensures the method's security (in the Kerckhoff's sense).

In practice, each watermarking method finds a trade-off among these four properties, according to the actual application constraints [5]. For instance, the authorized user can trade the data payload for reaching prescribed imperceptibility and robustness. On the other side, the unauthorized

user is expected to devise attacks that would abide to the imperceptibility constraint, while decreasing the robustness property.

The state-of-the-art analysis carried out in Section II highlights two methodological limitations of the NN watermarking landscape. First, the robustness is solely analyzed against mundane modifications related to NN life cycle (like pruning or fine-tuning, for instance) and implicitly assumes that the unauthorized user would not make any malicious attempt against the watermark. Secondly, although the NN application field is so broad, classification seems to be the only application benefiting from white-box watermarking.

The present study presents two contributions to the state-of-the-art in white-box watermarking [3], [6], [7]. First, the NN permutation attack [9] is formalized, thus demonstrating the very existence of *geometric* attacks for NN watermarking. Secondly, an effective counter-attack is devised and investigated on tasks beyond classification. The experiemntal study is based on three methods ([3], [6], [7]), four architectures (VGG16, ResNet34, DeepLabV3, and DVC), three tasks (image classification, segmentation, and video coding), and five types of attacks (Gaussian noise, fine-tuning, pruning, quantization, and permutation). Beyond analyzing the threats and opportunities related to NN geometric attacks and counter-attacks, this study serves as practical guidelines when designing effective NN watermarking methods.

II. BACKGROUND AND PROBLEM STATEMENT

This section first introduces the attack taxonomy as inherited from multimedia watermarking, then sketches the panorama of NN white-box watermarking solutions before identifying the issues raised by NN permutation attack.

II-A. Watermarking robustness and attack taxonomy

Robustness is the property of detecting the watermark, even when the watermarked model is subjected to modifications commonly referred to as *attacks*. The robustness is evaluated by assessing the ability to detect the watermark. For example, the BER (bit error rate) between the inserted and the recovered watermark can be computed [3], [6]. Alternatively, the correlation coefficient between the inserted and detected watermarks might be computed [7]. Conceptually, when evaluating the robustness, no distinction is made against mundane attacks (*i.e.* operations coming across with the usual NN life-cycle, like fine-tuning for better performances or pruning for lower footprint) and malicious attacks (*i.e.* operations specifically designed by unauthorized users to decrease the robustness).

In the multimedia realm, watermark attacks are classified as *removal attacks*, *geometric attacks*, *cryptographic attacks*, and *protocol attacks*. Removal attacks simply attempt to make the watermark unreadable. Geometric attacks do not try to remove the mark, but rather destroy the detector synchronization. Cryptography attacks aim at detecting and removing the watermark without any knowledge of the key, exploiting the fact that the embedded watermark is public and/or by assuming a detector (working with the proper key) is available. Finally, protocol attacks are meant to create ambiguity and confusion about watermark usage, even if properly detected. Removal and geometric attacks intimately relate to the insertion and detection methods. Cryptography attacks relate to the system security and secret key management and can be, for instance, based on *known text* attacks or on *oracle attacks*, as inherited from cryptography [10]. Protocol attacks deal with the practical watermark usage, as legal proof of copyright and/or integrity. **The present study will focus on removal and geometric attacks, while the last two classes can be conceptually considered complementary with respect to the paper scope.**

II-B. White-box neural network watermarking

The earliest NN watermarking methods [3] considers image classification, namely a wide residual network trained on CIFAR10 dataset or Caltech-101. A binary watermark of M bits is inserted in the so-called *flattened version of the layer* l, where M is lower than the number of input channels N_{l-1}. The key is represented by a random matrix $X \in \mathbb{R}^{N_{l-1} \times M}$. The mark is embedded during training via a regularization term minimizing the distance between the watermark and the projection of the flattened watermarked weights on the key. Watermark detection is achieved by projecting the watermarked (and possibly attacked) layer on the secret key, rounding the product results towards 0 or 1; the BER with respect to M is subsequently computed. The robustness is checked against fine-tuning (additional epoch of training without the embedding term up to 50% of the total training) and magnitude pruning (remove the fraction $\boldsymbol{T} \in [0.1; 0.99]$ of the smallest weights in terms of $L1$-norm).

While [6] inherits its key concept from [3], the mark is now embedded in the activation function of the selected layer. Four architectures are investigated: an MLP trained on MNIST, a test CNN and a WideResNet trained on CIFAR10, and ResNet50 trained on ImageNet. A binary watermark of M bits is inserted, according to a secret key represented by a random matrix $A \in \mathbb{R}^{N_l \times M}$. To embed the watermark, the output of the watermarked layer is estimated by a Gaussian mixture and two regularization terms are designed: the first one selects the Gaussian laws to be watermarked, while the second one, only activated for a subset of the training, minimizes the distance between the projection of those laws on the key and the watermark. Detection is performed by adapting the concepts in [3]. Robustness is checked against fine-tuning (up to 15% of the total training), magnitude pruning ($\boldsymbol{T} \in [0.1; 0.99]$, and watermark overwriting (embedding, with the same method, a new watermark).

The study in [7] randomly selects a set of parameters to be watermarked from multiple layers. Three classification models (ALL-CNN-C and ResNet32 trained on CIFAR10,

and LeNet5-caffe trained on MNIST) are considered. The watermark is represented by an image whose size depends on the model size. A subset of the initial weights is replaced by the pixels in the watermark, and their location is stored to serve as a secret key. The watermark is inserted via a regularization term making the inference highly sensitive to the selected parameters (hence, keeping those parameters unchanged during the training). Mark detection is achieved by recovering the selected parameters and by computing the Pearson's correlation between the original and retrieved watermarks. The robustness is checked against fine-tuning (up to 15% of the total training) and quantization (reduce the number of bits $\boldsymbol{B} \in [2; 16]$ representing the parameters).

II-C. Problem statement

The state-of-the-art analysis highlights two types of limitations in the white-box watermarking landscape. First, the robustness investigation preponderantly considers fine-tuning, pruning, and quantization, all belonging to the class of removal attacks. This originates the first question our study deals with: **"Do geometric attacks exist for NN watermarking? If so, how can they be handled?"**. Second, the application scope is generally restricted to image classification [3], [6], [7]. Moreover, [3] and [7] are *a priori* prone to be generalized to other application domains, while [6] is intimately connected to the classification task, and its conceptual generalization is not straightforward. So, the second question our study deals with is: **"Is NN watermarking restricted to classification tasks, or can it be effectively extended to other tasks? If so, is the robustness property modified?"**

III. GEOMETRIC ATTACKS TO NEURAL NETWORK

This work investigates i) whether geometric attacks can be defined for NN white-box watermarking, and ii) how can they be counter-attacked.

III-A. White-box permutation attacks

By definition, geometric attacks try to desynchronize the detector by altering the locations conveying the watermark. NNs are exposed to geometric attacks because they have many symmetrical, equi-loss representations that can be generated by a random *neuron permutation* within a layer, without affecting the neurons' functions. A corresponding permutation should also be applied to the input channel of the next layer (further referred to as *channel permutation*). Therefore, ensuring *a posteriori* resynchronization of neurons within a layer is a challenge in itself [11]. The process of permuting in-layer neurons can be accommodated by the following equations:

$$\boldsymbol{w}_{l,c,-}^{\pi_l} = \left\langle P_{\pi_l}, (\boldsymbol{w}_{l,c,-})^T \right\rangle \ \forall c, \tag{1}$$

$$\boldsymbol{w}_{l+1,-,n}^{\pi_l} = \left\langle P_{\pi_l}, (\boldsymbol{w}_{l+1,-,n})^T \right\rangle \ \forall n, \tag{2}$$

with $\boldsymbol{w}_l \in \mathbb{R}^{N_{l-1} \times N_l}$ being the weights for the l-th layer, $P(\pi_l)$ the applied permutation, $\langle \cdot \rangle$ inner product, and $(\cdot)^T$ the transpose operator. The equations above were derived for a single fully-connected layer without biases; yet, they can be extended to any other layer typology. This process can also be applied to any pair of consecutive layers.

In order to establish whether state-of-the-art white box methods are *a priori* robust against neuron permutation, they should be confronted to Eq. (1) and/or Eq. (2), as follows.

In [3], the detection is done by projecting the weights of the flattened watermarked layer on the secret key. Consequently, the neuron permutation on the l-th layer has no impact on detection. However, if the neuron permutation is applied to the $l-1$-th layer, the resulting channel permutation will completely desynchronize the watermark.

In [6], the detection is done by projecting the output of the watermarked layer on the secret key. Consequently, a complementary behavior with respect to [3] is encountered: the neuron permutation completely destroys the synchronization while the channel permutation preserves the synchronization.

In [7], the detection is done by using the secret key to locate the watermarked weights; hence, both neuron and channel permutations are likely to destroy the detection synchronization.

The above analysis demonstrates that Eq. (1) **and/or Eq.** (2) **stand for effective geometric NN watermarking attacks, as they jointly meet all the unauthorized user expectancies: (1) they succeed in destroying the mark detection, (2) they have no impact in the imperceptibility, as they preserve the watermarked NN output, and (3) they introduce no additional computational cost (in the sense that they just relate to the NN model representation and do not require any inference-related computation).** As a preliminary step towards ensuring robustness against this new type of attack, the possibility of defining counter-attack methods is investigated hereafter.

III-B. White-box permutation counter-attack

A posteriori resynchronization of neurons inside an NN layer subjected to neuron permutation is, in its general form, an exhaustive search problem in the space of factorial (over the number of neurons in the permuted layer) dimension. Regardless of the potential solution, the problem of recovering the original order for permuted neurons becomes even more complex for NN watermarking, when permuted neurons can also be modified by other types of attacks. The preliminary solution presented in [11] was not designed to be effective when supplementary operations (*e.g.* fine-tuning) are applied on the permuted neurons, while [9] targets the specification of a generic counterattack against the permutation. The advanced counterattack is based on creating a trigger set that differentiates one neuron from another and thus resynchronizes the model before retrieving the watermark. During the experiments, the permutation attack is applied to the

first or the second hidden layer of ResNet18 and ResNet50, with 160 elements in the trigger dataset. The performance of the counterattack is assessed by evaluating the BER (bit error rate) between the inserted and the retrieved watermark. The authors consider the counterattack successful for an experimental configuration when the BER is lower than 0.4 making the capacity of the original method reduced to 1 bit, indicating whether the watermark is inserted or not. The results show that the advanced counterattack has highly sensitive chances of success, depending on the experimental conditions. From a security point of view when using the same configuration as in [9], the information to be protected is the third layer of a ResNet-18, $\boldsymbol{w}_l \in \mathbb{R}^{64\times128\times3\times3}$ that has $73,728$ elements. According to [9], 160 Trigger inputs of size 32×32 (that is, $163,840$ elements) are created and should be kept secret. Hence, [9] requires at least twice more information to be kept secret than the information that is protected.

Our proposed counterattack consists of computing the cosine similarity between the un-attacked model, which is already public, and the attacked model. Indeed, despite the redundancy known to exist in NN models, we can expect the cosine similarity $S_C(\boldsymbol{w}_{l,i}, \boldsymbol{w}_{l,i}) = 1$, and hence, to have the following equation:

$$S_C(\boldsymbol{w}_{l,i}, \boldsymbol{w}_{l,i}) > S_C(\boldsymbol{w}_{l,i}, \boldsymbol{w}_{l,j}) \quad \forall j \neq i. \tag{3}$$

The original positions can be recovered by building the permutation matrix P_{π_l}:

$$(P_{\pi_l})_{i,j} = \begin{cases} 1 & j = \text{argmax}_k \left[S_C(\boldsymbol{w}_l, \boldsymbol{w}_l^{\pi_l})\right] \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

with $\boldsymbol{w}_l^{\pi_l}$ being a permuted version of the original weights. **To conclude with, Eq.** (3) **and Eq.** (4) **ensure effective reversion of the permutations described by Eq.** (1) **and Eq.** (2)**, and they can serve as a theoretical counter-attack in NN watermarking. Yet, there is no *a priori* ground about their behavior when several types of attacks are combined (*e.g.* permutation and fine-tuning), and an in-depth, complementary experimental study is required.**

IV. EXPERIMENTAL STUDY

This section presents a global yet detailed investigation of the robustness property. Section IV-A presents the experimental testbed, Section IV-B the results related to the robustness property in absence of any counter-attack, while Section IV-C illustrates the relevance of the geometric counter-attack.

IV-A. Experimental testbed

Watermarking methods and tasks. Three state-of-the-art methods are considered [3], [6], [7]. As explained in Section II, [3], [7] can be extended from classification towards image segmentation and video compression tasks, and they will be studied accordingly. In each case, the data payload and the imperceptibility are kept from their references. For each task, the imperceptibility criterion is provided by validation metrics considered during their training (*cf.* paragraph here-after). For [3] and [6], the watermark is inserted in one of the biggest convolutional layers and the penultimate layer, respectively; for [7] the watermarked weights are randomly selected through the whole model, respectively.

Watermarked architectures and training datasets. According to the three tasks, the watermarking methods are applied to four NN architectures trained on three datasets, namely: (1) VGG-16 and ResNet34 trained on CIFAR-10 for image classification, (2) DeepLabV3 [1] trained CityScapes [12] for image segmentation, and (3) DVC [2] trained on Vimeo-90k [13] and tested on UVG-dataset [14] for video compression. For the three tasks, the corresponding validation metrics are: (1) top-1 classification error, (2) the complementary mean Intersection over Unions (mIoU), and (3) the mean rate distortion *vs.* image quality, expressed in bit per pixel for a prescribed Multi-Scale Structural Similarity (bpp/MS-SSIM).

Attack parameters. First, four removal attacks are considered: Gaussian noise addition ($\mathcal{N}(0, \sigma_l \cdot \Omega)$, with $\Omega \in [0.01; 0.6]$, where σ_l is the standard deviation of the l-th layer), pruning (remove the $\boldsymbol{T} \in [0.1 : 0.99]$ fraction of the smallest weights in terms of $L1$-norm), fine-tuning (resume the training for up to 5% of the original number of iterations), and quantization (reduce the number of bits $\boldsymbol{B} \in [2; 16]$ used to represents the parameters). These attacks have been applied to the watermarked layers for [3] and [6]; this corresponds to the worst possible case for the authorized user, in the sense that, for a given imperceptibility value, they would provide the most harmful effects. In the case of [7], the attacks are applied over all the layers (as the mark is spread over an arbitrary, unknown, number of layers). In this case, in order to keep a fair comparison with [3] and [6], we target to keep constant the total amount of attacks induced in the watermarked NN, by adjusting the attack parameters accordingly, as detailed in Section IV-B. Second, the geometric attack and its counter-attack are applied to each and every layer in the NN.

IV-B. White-box robustness against attacks

The experimental results consider all the working configurations mentioned above and are illustrated in Table I.

In Table I, rows are first grouped according to the type of watermarked architecture (VGG16, ResNet34, DeepLabV3, DVC). Next, for each architecture, the rows are labeled according to the watermarking method. Columns are of three types, and provide information about the NN model in absence of any watermarking operation, on the watermarked NN in absence of any attack, and on the attacked watermarked NN. The first column is of the first type and presents the baseline performance of the NN model. The

Table I: Robustness evaluation for the different methods and architectures. For each combination, the parameter gives the value for an attack, imperceptibility is the performance on the validation set, and robustness corresponds to the watermarking metrics (C-BER and Pearson correlation coefficient) multiplied by 100. Blue box enlights a successful attack.

					Watermarked and attacked													
		Baseline	Watermarked		Gaussian			Pruning			Fine tuning			Quantization			Permutation	
		Perf.	Perf.	Rob.	Param.	Perf.	Rob.	Param.	Perf.	Rob.	Param.	Perf.	Rob.	Param.	Perf.	Rob.	Perf.	Rob.
VGG16 (classification)	[3]	11.01	11.86	100	0.6	31.4	100	0.99	25.15	100	5	11.79	100	2	90	100	11.86	**49**
	[6]		11.19	100	0.6	25.9	100	0.99	18.54	100	5	11.9	100	2	89.99	50	11.19	**62.5**
	[7]		12.49	99.99	0.06	24.6	73.97	0.9	77.22	99.99	5	8.81	99.99	2	80.77	99.99	12.49	**22.48**
ResNet34 (classification)	[3]	9.76	10.14	100	0.6	14.24	100	0.99	10.14	100	5	8.27	100	2	89.68	100	10.14	**55**
	[6]		8.72	100	0.6	19.35	100	0.99	23.38	100	5	8.21	100	2	90	37.5	8.72	**68.75**
	[7]		11.14	99.99	0.06	30.44	80.23	0.9	90	99.99	5	11.02	99.99	2	91.99	98.99	11.14	**11.04**
DeepLabV3 (segmentation)	[3]	33.1	36.23	100	0.6	42.51	100	0.99	94.72	100	5	36.46	100	2	97.99	62.5	36.23	**68.75**
	[7]		30.14	99.99	0.06	42.54	99.35	0.9	99.99	99.99	5	29.99	99.91	2	99.86	98.99	30.14	**46.39**
DVC (compression)	[3]	0.23/0.97	0.24/0.97	100	0.6	0.25/0.97	100	0.99	0.24/0.97	100	5	0.23/0.97	100	2	13.62/0.11	87.5	0.24/0.97	**37.5**
	[7]		0.23/0.97	99.99	0.06	5.81/0.21	62.56	0.9	0.50/0.63	99.99	5	0.23/0.97	93.62	2	13.62/0.31	99.99	0.23/0.97	**49.57**

next two columns are of the second type and provide the performance of the NN (according to the corresponding validation metric, IV-A) and the Robustness. The differences between the inserted and the recovered watermarks are expressed as complementary BER, denoted by C-BER and computed as ($C-BER = (1-BER)\times 100$) for [3], [6] and as Pearson coefficient (multiplied by 100) for [7]. The other columns are of the third type and are sub-grouped according to each investigated attack. In addition to performance and robustness, the "attacks parameter" is provided, except for the permutation attack where it is irrelevant. For each combination, Table I provides the parameter value for which the watermark can no longer be retrieved or, if the watermark fully withstands the set of values presented in Section IV-A, the value corresponding to the strongest attack. Note that information about the imperceptibility can be obtained by comparing the values of performance between the watermarked and baseline columns; similarly, information about the impact of the attacks in imperceptibility can be obtained by comparing the values of performance between the attack and watermarked columns.

Several conclusions can be drawn from Table I.

First, by comparing the Watermarked and Baseline columns, it is shown that, at least in absence of attacks, the application field of NN watermarking can be extended from classification to segmentation and compression. This conclusion is based on the fact that the three tasks result in quite an equal impact on performance. For classification, the relative differences in performance can be computed from the values presented in Table I; they range between -0.1 and 0.14, with an average of 0.05. Such values become -0.1, 0.1 and 0 for segmentation. In the case of video compression, while the MS-SSIM is constant, the relative variations in bpp become 0, 0.04, and 0.02. Note that actually the regularisation term included in [6], [7] for watermarking purposes also has a beneficial impact on the NN performance that can be increased with respect to the baseline. In each and every case, the watermark can be recovered (C-BER $= 100\%$ and Person's coefficient $= 0.99$). This opens the door to studies devoted to specific NN watermarking methods for segmentation and coding tasks.

Secondly, for each investigated NN and watermarking method, the robustness against the removal attacks is met, as either the watermark can be retrieved or the performance is lowered beyond the application purpose. In this respect, for any of the three tasks, the Gaussian, pruning, and fine-tuning attacks do not have any impact on the watermark detection, as demonstrated by values C-BER $= 100\%$ and Person's coefficient > 0.6. When considering the quantization attack, the watermark can be lost (C-BER $\leq 90\%$) but the performance decreased beyond the application requirements; just for illustration, in the case, [6] and VGG16 architecture, C-BER $= 50$ but the top-1 error becomes $= 89.99$. Similar behavior is encountered for [6] on ResNet34 and [3] on DVC.

In contrast to removal attacks, the geometric attack is always successful: for the same performance as the watermarked model, the mark cannot be anymore detected (C-BER $\geq 70\%$ and Person's coefficient ≤ 0.5). Hence, the effectiveness of the geometric attack defined by Eq. (1) and Eq. (2) is demonstrated, and the need for evaluating the counter-attack defined by Eq. (3) and Eq. (4) is proved.

IV-C. Geometric counter-attack performance

The counter-attack to geometric modifications is applied to each of the working configurations investigated in the previous sub-section. The results are synoptically displayed in Fig. 2 for [3], [6] and in Fig. 3 for [7].

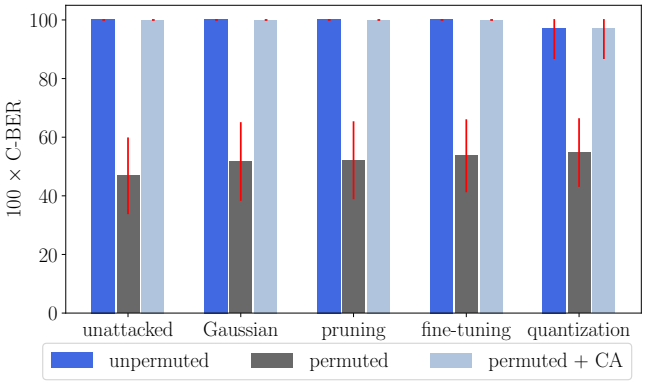

Fig. 2: Robustness evaluation of geometric counter-attack against the different removal attacks for methods [3], [6].

Each of these two figures is structured in five areas. The first area corresponds to the case when no removal attack is applied on the watermarked NN. The other 4 areas correspond to the cases when the four removal attacks are individually applied (from left to right: Gaussian noise addition, pruning, fine-tuning, and quantization, respectively). In its turn, each of these 5 areas shows three bars corresponding to the cases of: no additional geometric attack - labeled by (unpermuted), an additional geometric attack - labeled by (permuted), and an additional geometric attack followed by its counter-attack - labeled by (permuted+CA).

While the abscissas are identical for these two figures, their ordinates are different. Figure 2 provides average C-BER values (multiplied by 100) and their related $\pm$ standard deviation intervals (bounded at the maximum theoretical value of 100). The averages are computed over all the NN architectures, all the investigated attack parameters, and the methods in [3], [6]; the standard deviation is computed as an unbiased estimator over the same data. In Fig. 3, the coordinate corresponds to the Person's coefficient (multiplied by 100) and also presents average and $\pm$ standard deviation intervals (bounded at the maximum value of 100); this time, the average is computed only for [7], over all the NN architectures and all the investigated attack parameters; the standard deviation is also computed as an unbiased estimator Fig. 2 and Fig. 3 demonstrate that Eq. (3) and Eq. (4) are effective geometric counter-attacks: they can synchronize back the mark detection even when the geometric attack is applied in conjunction with any of the 4 investigated removal attacks.

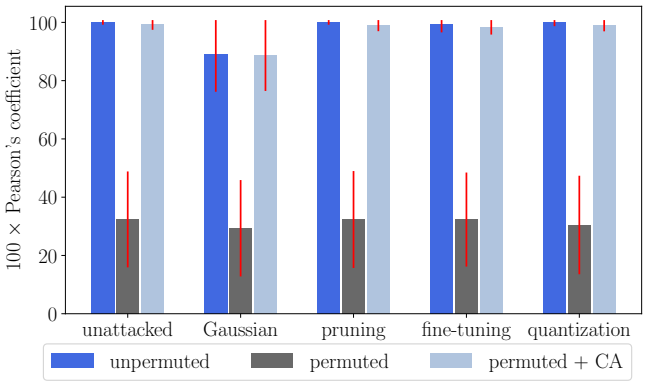

Fig. 3: Robustness evaluation of geometric counter-attack against the different removal attacks for method [7].

V. CONCLUSION

The present paper presents an in-depth investigation of NN watermarking robustness. First, it shows that the neuron and channel permutation operations can be transposed into an effective, new type of attack (the first in the geometric attacks family), and provides the matched counter-attack. Secondly, it demonstrates that the counter-attack is effective in ensuring robustness when the geometric attack is applied by itself or in conjunction with any of the four state-of-the-art removal attacks (Gaussian noise addition, pruning, fine-tuning, and quantization). As a side result, the study establishes that the NN watermarking scope can be extended from classification tasks to segmentation and compression, and identifies the performance gap to be bridged by future methods. Finally, the level of detail of the quantitative results presented in the study can provide guiding information for an experimenter who would like to get to a practical NN watermarking solution. Future work will be devoted to investigating the coupling of several types of removal attacks as well as to identifying the potential synergies and anatomies when coupling removal, geometric and cryptography attacks. Extending the principle from this study to devise a generic regularisation term that can be dynamically used as a counter-attack is also part of future work.

VI. REFERENCES

[1] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[2] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An end-to-end deep video compression framework," in *Proceedings of CVPR*, 2019.

[3] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of ICMR*. jun 2017, ACM.

[4] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *USENIX Security*, 2018.

[5] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*, Morgan kaufmann, 2007.

[6] B. Darvish Rouhani, H. Chen, and F. Koushanfar, "Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks," in *in proceedings of ASPLOS*, 2019.

[7] E. Tartaglione, M. Grangetto, D. Cavagnino, and M. Botta, "Delving in the loss landscape to embed robust watermarks into neural networks," in *2020 25th ICPR*. IEEE, 2021.

[8] E. Le Merrer, P. Perez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, no. 13, 2020.

[9] Fang-Qi Li, Shi-Lin Wang, and Yun Zhu, "Fostering the robustness of white-box deep neural network watermarks by neuron alignment," in *ICASSP 2022-2022 IEEE*. IEEE, 2022.

[10] A. G. Konheim, *Cryptography, a Primer*, John Wiley & Sons, Inc., USA, 1st edition, 1981.

[11] K. Ganju, Q. Wang, W. Yang, C. A Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proceedings of ACM SIGSAC*, 2018.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on CVPR*, 2016.

[13] T. Xue, B. Chen, J. Wu, D. Wei, and W. T Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, 2019.

[14] A. Mercat, M. Viitanen, and J. Vanne, "Uvg dataset: 50/120fps 4k sequences for video codec analysis and development," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020.