Optimizing Additive Approximations of Non-Additive Distortion Functions

Solène Bernard
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL,
Lille, France
solene.bernard@centrale.centralelille.fr

Patrick Bas
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL,
Lille, France
Patrick.Bas@centralelille.fr

John Klein
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL,
Lille, France
john.klein@univ-lille.fr

Tomáš Pevný
Faculty of Electrical Engineering, Czech Technical
University,
Prague, Czech Republic
tomas.pevny@fel.cvut.cz

ABSTRACT

The progress in steganography is hampered by a gap between non-additive distortion functions, which capture well complex dependencies in natural images, and their additive counterparts, which are efficient for data embedding. This paper proposes a theoretically justified method to approximate the former by the latter. The proposed method, called *Backpack* (for BACKPropagable AttaCK), combines new results in the approximation of gradients of discrete distributions with a gradient of implicit functions in order to derive a gradient w.r.t. the distortion of each JPEG coefficient. Backpack combined with the min max iterative protocol leads to a very secure steganographic algorithm. For example, the error rate of XuNet on 512 × 512 JPEG images, compressed with quality factor 100 and a payload of 0.4 bits per non-zero AC coefficient is 37.3% with Backpack, compared to a 26.5% error rate using ADV-EMB with min max protocol (considered state of the art in this work) and a 16.9% error rate with J-UNIWARD.

ACM Reference Format:
Solène Bernard, Patrick Bas, John Klein, and Tomáš Pevný. 2018. Optimizing Additive Approximations of Non-Additive Distortion Functions . In *Proceedings of -*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Steganography by cover modification hides data into an innocuous looking content, such that the mere fact of a payload being hidden cannot be reliably detected. A counterpart of steganography is steganalysis, aiming to detect contents with hidden data. Antagonist goals of steganographers versus steganalysts force them to play a cat-and-mouse game (called *steganographic game* below) commonly found in security fields. This paper takes a step forward in removing humans from playing this game by proposing an automatic way to design attacks and counter-attacks.

State of the art steganographic algorithms rely on the principle of minimizing a distortion function, which needs to be additive for practical reasons. Before hiding the message (embedding), a steganographic algorithm assigns to each modifiable coefficient (e.g. a pixel or DCT coefficient if the cover object is an image) an embedding cost. The message is then hidden by syndrome trellis codes [9], which minimizes the additive distortion function under the constraint of communicating a message. For research purposes, the act of embedding a concrete message is skipped in favor of a mere simulation [10, 17]. Presently, the design of new steganographic algorithms therefore boils down to defining the embedding costs.

Note that the construction of costs falls into two (and half) categories. Heuristics are used in UNIWARD [14], HILL [20] or UERD [12], where costs promote preservation of properties in natural images that steganography might violate, as for example smoothness. An alternative, in our view more sophisticated, approach [11, 24, 26] calculates the cost to minimize detectability of a specific steganalyzer (also called *detection function* or *detector*). Note that this detection function (typically modeling the noise extracted from one cover image) is different from that used by real steganalytic detectors (built from numerous cover and stego images), since its direct minimization may be complicated in practice [25].

An evolution is to automate the design of detection functions. The very first attempt known to us was ASO [19], which convergence issues prevented a wider adoption and success. More recent approaches relied on generative adversarial networks (GAN) [30, 33] which simultaneously optimize a neural network predicting embedding costs and another neural network as steganalyzer. It is known that under some assumptions, GANs converge to the Nash equilibrium [23] of a steganographic game, with actions defined by spaces parametrized by chosen models. Yet, the convergence of GANs is difficult to achieve in practice. An alternative to GANs is [5] which does not use the fixed estimator of embedding changes. Instead, the distortions associated to each DCT sample are optimized by means of adversarial embedding (ADV-EMB) [29] for a given image and steganalyzer. The main contribution of [5] and [4] was to show, how to construct iteratively steganalyzers without suffering from the convergence issues of ASO.

The set of steganalyzers created by the min max protocol in [5] plays the role of non-additive distortion functions, as they measure the detectability of a given image. Indeed the adversaries (here the classifiers) capture mid-range dependencies between DCT coefficients which are by definition non-additive. In order to overcome

-, -, -
© 2018 .
978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/1122445.1122456

this issue, the embedding can induce correlations by either using lattices (see section 2.2) and/or by crafting specific non-symmetric additive costs as done here. For example ADV-EMB [29] adjusts costs of some symmetric additive distortion function (J-UNIWARD) w.r.t. the gradient of the loss of one steganalyzer by performing only one gradient descent step and by distinguishing costs related to +1 to costs related to -1 w.r.t. the gradient of the detectability function. In Section 2.4 it is shown however that ADV-EMB cannot reliably adjust embedding costs such that the resulting stego object is undetectable by the worst classifier from a set. Moreover, ADV-EMB does not natively take the length of embedded message into account.

This paper proposes a method called *Backpack* (for BACKPropagable AttaCK)[1] fixing flaws of ADV-EMB by finding embedding costs by gradient descent with constraint on message length included by means of implicit differentiation. To put the proposed method into a wider context, it optimizes parameters of additive distortion function (embedding costs) by challenging a non-additive distortion function (a set of steganalyzers) for a given image and message length. In this sense, the method is general. The experimental evaluation demonstrates its advantages with respect to ADV-EMB by improving the security of a steganographic algorithm found by min max protocol [5] by 11% when the distortion function is a deep neural network with a Xu-Net [32] architecture.

This paper is organized as follows. The next section recalls important prior arts for explaining and framing of the method and also shows the weakness of ADV-EMB algorithm. Section 3 formally defines the problem and shows how to calculate the gradient of a differentiable steganalyzer with respect to embedding costs. Experimental section 4 shows the effect of the proposed scheme on security of the obtained embedding costs.

Notations

In the following, letters in bold are used to represent vectors and matrices, a corresponding non bold letters are used for their elements. Calligraphic letters are used for sets. Cover and stego objects are denoted as vectors and they are respectively denoted as $\mathbf{c} = (c_i)$ and $\mathbf{s} = (s_i)$, $i \in \{1, \ldots, n\}$ (ranges in sums are dropped if it is clear from context). $\mathcal{B}$ denotes the set of allowed embedding changes made to the cover. Subscripts $i \in \{1, \ldots, n\}$ denote index of a coefficient, whereas superscripts $j \in \mathcal{B}$ are for embedding directions.

Spaces $\mathcal{I}$, $\mathcal{M}$, and $\mathcal{K}$ are respectively the space of all images, messages and keys with appropriate distributions, $\mathcal{P}_{\cdot}$, defined over them.

A steganalyzer (or steganographic detector or detector) is any function $f(\mathbf{x}) : \mathcal{I} \mapsto \mathbb{R}$ and $\mathbf{x}$ is assigned to stego class if the output is greater than some threshold t. A steganographic algorithm is any pair of functions $h_{\text{emb}}(\mathbf{x}, m, k) : \mathcal{I} \times \mathcal{M} \times \mathcal{K} \to \mathcal{I}$ and $g_{\text{ext}}(\mathbf{x}, k) : \mathcal{I} \times \mathcal{K} \to \mathcal{M}$ for which it holds that $g_{\text{ext}}(h_{\text{emb}}(\mathbf{x}, \mathbf{m}, k), k) = \mathbf{m}$ for all $\mathbf{m} \in \mathcal{M}$, $k \in \mathcal{K}$, and $\mathbf{x} \in \mathcal{I}$..

2 BACKGROUND

2.1 Distortion minimization principle

Steganography by cover modification maps a cover object $\mathbf{c}$ to a stego object $\mathbf{s}$ such that it communicates a message $\mathbf{m}$ while minimizing a distortion function $f(\mathbf{s}, \mathbf{c})$. The range of embedding changes $\mathbf{b} = \mathbf{c} - \mathbf{s}$ is usually restricted to the set $\mathcal{B}$, which is most of the time small $\mathcal{B} = \{-1, 0, +1\}$.

For general distortion functions f the above problem is NP-hard, therefore practical embedding schemes use an additive $f(\mathbf{s}, \mathbf{c}) = \sum_{i=1}^{n} \rho_i^{c_i - s_i}$, where ρ_i^j denotes embedding cost of changing the i^{th} pixel by adding $b_i = c_i - s_i$. This work both assumes asymmetric costs $\rho_i^{+1} \neq \rho_i^{-1}$ and that not making change can cause a distortion $\rho_i^0 \neq 0$. Additive distortion function (for a given cover) is fully determined by embedding costs $\boldsymbol{\rho}$ and vice versa.

Given cover object $\mathbf{c}$, message $\mathbf{m}$,, key k and embedding costs $\boldsymbol{\rho}$, the $\boldsymbol{\rho}$-parametrized embedding function $h_{\text{emb}}(\mathbf{c}, \mathbf{m}, k; \boldsymbol{\rho})$ solving the above optimization problem for additive distortion function is usually implemented by Syndrome Trellis Codes [9] (STC), which produce stego objects with distortion usually within 5% − 7% of that of the optimal embedding.

As shown in [10], the *optimal embedding* minimizing an additive distortion function can be simulated by adding random embedding changes $\mathbf{b}$ to a cover signal, i.e. $\mathbf{s} = \mathbf{c} + \mathbf{b}$, where embedding changes $\mathbf{b}$ are realizations of a random variable distributed according to $P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho}, \lambda) = \prod_{i=1}^{n} P_{b_i}(b_i = j|\rho_i, \lambda)$ where

$$P_{b_i}(b_i = j|\rho_i, \lambda) = \frac{e^{-\lambda \rho_i^j}}{\sum_{k \in \mathcal{B}} e^{-\lambda \rho_i^k}}, \quad i \in \{1, \ldots, n\}, j \in \mathcal{B} \tag{1}$$

with a condition that the entropy of the steganographic channel is equal to the length of the message, i.e :

$$H(P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho}, \lambda)) = |\mathbf{m}|, \tag{2}$$

where $|\mathbf{m}|$ denotes the length of the message in bits and $H(\mathbf{p}) = -\sum_{i,j} p_i^j \log p_i^j$ if $\mathbf{p} = (p_i^j)$ is a matrix. The stego object $\mathbf{s}$ created by the simulation is random, as it is a realization of a random variable and as the message is not fixed and assumed to be drawn randomly from a uniform distribution. In contrast, in STC, the embedding function is deterministic, since the message is fixed.

2.2 Non-additive distortion functions

The above theory is well developed for additive distortion functions, unfortunately many interesting distortion functions, mainly those corresponding to state of the art steganalyzers are non-additive and the above methodology does not apply.

Gibbs construction [8] proposes to solve the problem using Gibbs sampling, which is theoretically correct (albeit not optimal), but prohibitively expensive. A greedy approximation by dynamic programming is proposed in [25], but it is also computationally expensive. Other constructions rely on decomposing the image into disjoint lattices in order to synchronize embedding changes [21, 28]. Contrary to Gibbs sampling, the embedding stops when all the lattice have been visited once.

A completely different approach inspired by the creation of adversarial attacks is offered by adversarial embedding ADV-EMB [29].

[1] The name is derived from methods which inspired the work: backpropagation and adversarial attacks on neural networks.

ADV-EMB heuristically modifies embedding costs $\boldsymbol{\rho}$ of some existing embedding algorithm (J-UNIWARD) to avoid detection by a fixed steganalyzer f. Specifically, embedding costs are divided into a *common*, $\mathcal{L}_c$, and an *adjustable*, $\mathcal{L}_a$, group and they are modified according to

$$\begin{aligned} \hat{\rho}_i^j &= \rho_i^j \alpha^{j\mathrm{sign}(\frac{\partial f(\mathrm{s})}{\partial \mathrm{s}_i})} \text{iff } i \in \mathcal{L}_a, \\ \hat{\rho}_i^j &= \rho_i^j \text{iff } i \in \mathcal{L}_c, \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{\rho}}$ denotes new embeddings costs and $\alpha = 10$ is a parameter. Notice that embedding costs of coefficients in the common group are not changed. ADV-EMB uses heuristic to overcome a major hurdle in applying a gradient descend to optimization of embedding costs, which is that the embedding operation is not differentiable. Instead of using a proper gradient (or its estimate), it uses its sign and fixes the step. ADV-EMB has been reported to perform well against a single fixed detector. By an exhaustive search the algorithm minimizes the number of adjusted embedding costs to prevent modifying too many embedding coefficients, which might be easily detectable.

2.3 $\min\max$ protocol

ADV-EMB [29] allows efficiently to create a stego object secure with respect to a given differentiable steganalyzer. Although the same reference proposes an iterative algorithm to create a steganalyzer, such that the resulting stego images are secure, the proposed iterative algorithm does not have any theoretical guarantees. A theoretically justified approach proposed in [5] is revised below.

Security of a practical steganographic scheme h_{emb} with respect to a set of steganalyzers $\mathcal{F}$ is defined as the error of the best steganalyzer :

$$\arg\min_{f \in \mathcal{F}} \frac{1}{2}\left[\mathbb{E}_{x \sim P_c}\left[\mathrm{I}[f(\mathbf{x}) \geq t] + \mathrm{I}[f(h_{\mathrm{emb}}(\mathbf{x})) < t]\right]\right]. \tag{4}$$

where I is the indicator function, and t a decision threshold. Since $\mathcal{F}$ is in practice very large it is difficult to create a steganographic algorithm by explicitly optimizing this criteria. Ref. [5] decreases the computational complexity by (i) selecting a small but representative subset of $\mathcal{F}$and (ii) using ADV-EMB to create stego images maximally secure with respect to this small subset of $\mathcal{F}$. The subset of $\mathcal{F}$ is constructed iteratively starting with an empty set $\mathcal{F}^0 = \emptyset$. At the k^{th} iteration, the protocol consists of the following two steps :

1. Create a set of stego images $\mathcal{S}^k$ maximally secure with respect to the set of detectors $\mathcal{F}^{k-1} = \{f^0, f^1, \ldots, f^{k-1}\}$.
2. Build a new detector f^k detecting stego images produced in step (1) and add it to the set $\mathcal{F}^{k-1}$, i.e. $\mathcal{F}^k = \mathcal{F}^{k-1} \cup \{f^k\}$.

Notice that the construction of the small subset explicitly exhibits a cat and mouse game played by the community for decades, where step (1) corresponds to proposing a new steganographic algorithm maximally secure with respect to all known steganalyzers and step (2) corresponds to proposing a new steganalyzer breaking the newly proposed steganographic algorithm.

2.4 Flaws of ADV-EMB in $\min\max$ protocol

During the $k+1^{\mathrm{th}}$ iteration, the $\min\max$ protocol uses ADV-EMB to create stego images undetectable by all steganalyzers in $\mathcal{F}^k$. The gradient of the steganalyzer with respect to stego image used in Equation (3) therefore becomes :

$$\frac{\partial \max_{f \in \mathcal{F}^k} f(\mathbf{s})}{\partial \mathbf{s}} = \frac{\partial \tilde{f}(\mathbf{s})}{\partial \mathbf{s}},$$

where $\tilde{f} = \arg\max_{f \in \mathcal{F}^k} f(s)$. Since ADV-EMB calculates the gradient and adjusts embedding costs *just once*, the resulting stego image is optimized only with respect to the classifier $\tilde{f}$ and ignoring remaining adversaries $\mathcal{F}^k \setminus \tilde{f}$, which means that the resulting stego image created with adjusted embedding costs $\hat{\boldsymbol{\rho}}$ can be still detectable by them.

Moreover, as mentioned before, the use of the gradient in (3) is heuristic, and does not reflect the fact that the embedding costs should directly be modified according to the gradient of $f(.)$ w.r.t. the costs.

3 DIFFERENTIABLE STEGANOGRAPHY

3.1 Problem formulation

The problem we solve is for a given cover object $\mathbf{x}$ to find an embedding cost map $\boldsymbol{\rho}$ minimizing detectability of a stego object $\mathbf{s} = \mathbf{c} + \mathbf{b}$ by a non-additive distortion function $f(\mathbf{s})$ (read steganalyzer), where $\mathbf{b} \sim P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho}, \lambda)$ and f being almost everywhere differentiable with respect to $\mathbf{s}$. We therefore focus on *simulation* of embedding changes. The stego object $\mathbf{s} = \mathbf{c} + \mathbf{b}$ is a realization of a random variable and we minimize the expected detectability over all possible stego objects written as :

$$\arg\min_{\boldsymbol{\rho}, \lambda} \mathbb{E}_{\mathbf{b} \sim P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho}, \lambda)}\left[f(\mathbf{c} + \mathbf{b})\right], \tag{5}$$

subject to the entropy constraint:

$$H(P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho}, \lambda)) = |\mathbf{m}|. \tag{6}$$

We want to solve the above problem using gradient descend with respect to $\boldsymbol{\rho}$, because it is efficient but its use for this problem is difficult from two reasons: first, the optimization problem contains an implicit constraint on the entropy; second the gradient of the expectation of f with respect to $\boldsymbol{\rho}$ does not have an analytical expression and its exact computation (summation over all possible embedding changes) would be prohibitively expensive. The rest of this section first presents how to overcome these two hurdles and then describes an algorithm to efficiently minimize (5).

3.2 Estimating the gradient of expectation of a discrete distribution

Let us first focus on the problem of calculating the gradient of:

$$\frac{\partial}{\partial \rho} \mathbb{E}_{\mathbf{b} \sim P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho}, \lambda)}\left[f(\mathbf{c} + \mathbf{b})\right], \tag{7}$$

without the entropy constraint (we assume λ is fixed for now). The exact computation of the expectation (and therefore its gradient) is computationally very expensive. To compute it exactly one would need to sum over the support of all stego images (for a given cover), which has $|\mathcal{B}|^n$ (recall that $|\mathcal{B}|$ is the cardinality of embedding changes and n is the number of coefficients that can be modified during embedding). The rest of this section therefore focuses on an *approximation* of (7) that would be sufficiently accurate while being computationally cheap.

To simplify the notation, the calculation of the expected gradient is introduced for a single coefficient. This means that the modification b of a coefficient is a scalar with probability distribution described by a vector $\mathbf{p}$ of length $|\mathcal{B}|$. Since embedding changes are assumed to be independent (this is required by our constraint on the distortion function to be additive), the generalization to all coefficients of an object is straightforward.

3.2.1 Hardmax Gumbel. The problem of calculating the gradient of expectation of discrete probability distribution with respect to its parameters is very well studied problem. From the vast prior art, we have chosen the method [16] relying on the Gumbel distribution. This technique has an advantage of giving a general formula to draw samples according to any discrete distribution, so it can be used without a modification for n-ary coding, and its theoretical properties are well analyzed.

A discrete distribution defined by a probability vector $\mathbf{p} = (p^j)_{j\in\mathcal{B}}$, (recall that $\mathbf{p}$ specifies probability of changing a single coefficient to values allowed by embedding) can be sampled by dividing also interval $[0, 1]$ into $|\mathcal{B}|$ buckets of size $\mathbf{p}$, and then returning index of a bucket to which a random variable with uniform distribution on $[0, 1]$ falls. An alternative approach is to sample the same distribution according to following strategy :

$$b = \mathrm{MG}(\mathbf{p}, \mathbf{g}) = \arg\max_{j\in\mathcal{B}}(g^j + \log p^j), \tag{8}$$

where elements of $\mathbf{g}$ are independently sampled from the Gumbel distribution $G(0, 1)$. A softmax function

$$\mathrm{softmax}\,(x_1, \dots, x_n) = \frac{1}{\sum_{k=1}^{n} e^{x_k}}(e^{x_1}, \dots, e^{x_n}),$$

is a well known approximation of arg max, which can be seen from

$$\lim_{\tau\to 0} \mathrm{softmax}\left(\frac{x_1}{\tau}, \dots, \frac{x_n}{\tau}\right) = (0, 0 \dots, 0, 1, 0, \dots, 0),$$

where the 1 is on $\arg\max_i x_i$ place and τ controlling the smoothness of the approximation is called *temperature*. Figure 1 offers a visualization of the influence of τ on the output of the Softmax-Gumbel (SG) function, for a fixed realization of a random vector $\mathbf{g}$ and fixed probability vector $\mathbf{p}$.

Replacing arg max in Equation (8) by a softmax approximation with temperature leads to :

$$\tilde{b}_\tau \quad = \quad \mathrm{SG}_\tau(\mathbf{p}, \mathbf{g}) = \sum_{j\in\mathcal{B}} j\; z^j, \tag{9}$$

$$\text{with } \mathbf{z} \quad = \quad \mathrm{softmax}\left(\frac{\mathbf{g} + \log \mathbf{p}}{\tau}\right), \tag{10}$$

where $\mathbf{g}$ and $\mathbf{p}$ are as above. The last Equation (9) is easily differentiable with respect to $\mathbf{p}$ since the calculation can treat the random numbers $\mathbf{g}$ as constants (in machine learning this is called a *re-parametrization trick*).

The main advantage of re-parametrization this is that the gradient in Equation (7) can be now estimated using k samples $\mathbf{g}_1, \dots, \mathbf{g}_k$ as

$$\left(\frac{1}{K}\sum_{j=1}^{K} \frac{\partial f(\mathbf{c} + \mathrm{SG}_\tau(\mathbf{p}, \mathbf{g}_j))}{\partial \mathbf{p}}\right)\frac{\partial \mathbf{p}}{\partial \rho}. \tag{11}$$

The number K of drawn samples $\mathbf{g}_1, \dots, \mathbf{g}_K$, control the trade-off between variance of the estimate and computational complexity. It is common in to use K as small as $K = 1$, i.e. estimating the gradient from a single sample (compare this to complexity $|\mathcal{B}|^n$ of the exact computation in Equation (7)). The drawback of this estimator is that it is biased. In experiments presented in this paper, K was selected to the highest number which the GPU memory allowed.

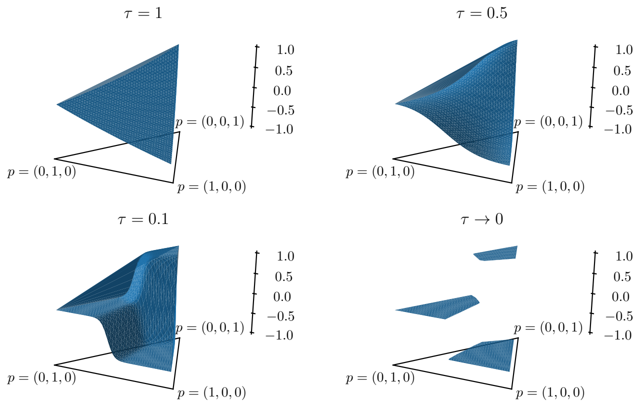

Figure 1: For a given value of triplet $g = (g^{-1}, g^0, g^{+1})$ (where $g^j \sim G(0, 1)$ are independently drawn from Gumbel standard distribution), value of the modification $\tilde{b}_\tau = \mathbf{SG}(p, g)$ is plotted the z-axis for all possible triplets of probabilities $p = (p^{-1}, p^0, p^{+1})$, and for 4 values of τ. The triplets are plotted in the trilinear coordinate system.

3.3 Incorporating constraint on entropy

Assuming gradient (7) can be efficiently approximated by Equation (11), compliance with the constraint on entropy (6) is still required. Among other possibilities (e.g. the method of Lagrange multipliers), one can remove constraints by incorporating them into the optimization term through an implicit function. This does not does not change the solution of the optimization but simplifies it. This subsection shows that this is indeed possible in problem (5).

Recall that for a given embedding cost $\boldsymbol{\rho}$, the probabilities of changing coefficients $P_{\mathbf{b}}$ are calculated by Equation (1) for a λ being solution of an entropy constraint given by Equation (6). Thus for a given $\boldsymbol{\rho}$, λ is a unique solution of some implicitly defined function, which is here denoted as $\Lambda(\boldsymbol{\rho}, |\mathbf{m}|) = \lambda$. Substituting this implicit function $\Lambda(\boldsymbol{\rho}, |\mathbf{m}|)$ into the Equation (1) for distribution of embedding changes, gradient of Equation (7) with respect to embedding costs ρ can be written as :

$$\frac{\partial}{\partial \rho}\mathbb{E}_{\mathbf{b}\sim P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho},\Lambda(\boldsymbol{\rho},|\mathbf{m}|))}\left[f(\mathbf{c} + \mathbf{b})\right],$$

which is free of λ but contains the implicit function Λ. According to the chain rule of derivatives, this gradient becomes :

$$\left(\frac{\partial}{\partial \mathbf{p}}\mathbb{E}_{\mathbf{b}\sim P_{\mathbf{b}}(\mathbf{b}|\mathbf{p})}\left[f(\mathbf{c} + \mathbf{b})\right]\right)\left(\frac{\partial \mathbf{p}}{\partial \rho}\right). \tag{12}$$

While good approximations of the first term can be obtained as explained in 3.2, computing $\frac{\partial \mathbf{p}}{\partial \rho}$ is more cumbersome since $\mathbf{p}$ depends on both ρ and λ.

Because $\mathbf{p} = P_{\mathbf{b}}(\mathbf{b}|\rho, \lambda)$, writing the total derivative gives:

$$\frac{\partial \mathbf{p}}{\partial \rho} = \frac{\partial P_{\mathbf{b}}}{\partial \rho}\frac{\partial \rho}{\partial \rho} + \frac{\partial P_{\mathbf{b}}}{\partial \lambda}\frac{\partial \Lambda(\rho, |\mathbf{m}|)}{\partial \rho}. \tag{13}$$

Although function Λ is implicit, its gradient $\frac{\partial \Lambda(\rho,|\mathbf{m}|)}{\partial \rho}$ can be computed. Recall that for a given ρ, λ is a solution of an entropy constraint (Equation (6)), therefore it holds that

$$H(P_{\mathbf{b}}(\lambda, \rho)) = H(P_{\mathbf{b}}(\Lambda(\rho, |\mathbf{m}|), \rho)) = |\mathbf{m}|, \tag{14}$$

and therefore $H(P_{\mathbf{b}}(\Lambda(\rho, m), \rho)) - |\mathbf{m}| = 0$. Applying the chain rule to this equation gives :

$$\frac{\partial}{\partial \rho} H(P_{\mathbf{b}}(\rho, \Lambda(\rho, |\mathbf{m}|))) = \frac{\partial H(P_{\mathbf{b}})}{\partial \rho}\frac{\partial \rho}{\partial \rho} + \frac{\partial H(P_{\mathbf{b}})}{\partial \lambda}\frac{\partial \Lambda}{\partial \rho} = 0, \tag{15}$$

from which the desired gradient of $\Lambda(\rho, |\mathbf{m}|), \rho)$ can be expressed as

$$\frac{\partial \Lambda(\rho, |\mathbf{m}|)}{\partial \rho} = -\left(\frac{\partial H(P_{\mathbf{b}})}{\partial \lambda}\right)^{-1} \frac{\partial H(P_{\mathbf{b}})}{\partial \rho}. \tag{16}$$

Combining Equation (16) and Equation (13) with Equation (12) yields to a closed form expression for the gradient :

$$\left(\frac{\partial}{\partial \mathbf{p}} \mathbb{E}_{\mathbf{b} \sim P_{\mathbf{b}}(\mathbf{b}|\mathbf{p})}[f(\mathbf{c}+\mathbf{b})]\right)\left(\frac{\partial P_{\mathbf{b}}}{\partial \rho} - \frac{\partial P_{\mathbf{b}}}{\partial \lambda}\left(\frac{\partial H(P_{\mathbf{b}})}{\partial \lambda}\right)^{-1}\frac{\partial H(P_{\mathbf{b}})}{\partial \rho}\right) \tag{17}$$

3.4 Optimizing embedding costs

Results presented in above sections allows us to efficiently approximate the gradient of Equation (7) by estimating a gradient of its smooth approximation with a constraint on the entropy. This allows use to adapt a gradient descend method to minimize detectability with respect to all detectors in a set $\mathcal{F}^k$ as needed in min max protocol.

The proposed algorithm with pseudocode shown in Algorithm 1 uses continuous approximation of discrete embedding changes to optimize the embedding costs ρ. In every iteration, it checks if detectability of stego images with discrete embedding costs is below threshold. If yes, the algorithm is terminated, otherwise it continues. If the detectability of stego images with continuous approximation is below a given threshold, the temperature is halved. In practice, there is also a limit on the maximum number of iterations. All expectations in the pseudocode are estimated from a single sample, as described in Section 3.2.1. The threshold on detectability is the detectability of the unmodified cover object.

The progress of the proposed algorithm on minimizing detectability of a single stego object against a single detector is shown in Figure 2. Although the optimization uses continuous approximation of stego objects (blue line), the main goal is to create stego objects with discrete embedding change (orange line). We can observe that in the very beginning, when temperature is high, there is a big difference between detectability of continuous approximations and that of real stego objects. But as the algorithm progresses and temperature decreases, this difference becomes negligible.

The proposed algorithm is iterative, therefore it does not suffer the weakness of ADV-EMB described in Section 2.4 and it is well suited to minimize detectability measured as a maximum over a set of steganalyzers.

```
Data: A JPEG cover image $\mathbf{c}$, initial embedding costs $\rho^0$,
      initial $\tau^0$
Result: An adversarial embedding costs $\boldsymbol{\rho}$
$\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}^0$;
$\tau \leftarrow \tau^0$;
$\tilde{o} \leftarrow \max_i \mathbb{E}_{\tilde{\mathbf{b}}_\tau \sim P(\lambda,\boldsymbol{\rho})}[f^i(\mathbf{c}+\tilde{\mathbf{b}}_\tau) - f^i(\mathbf{c})]$;
$o \leftarrow \max_i \mathbb{E}_{\mathbf{b} \sim P(\lambda,\boldsymbol{\rho})}[f^i(\mathbf{c}+\mathbf{b}) - f^i(\mathbf{c})]$;
while True do
    while $\tilde{o} > 0$ and $o > 0$ do
        Update $\boldsymbol{\rho}$ by one step of gradient descend with $\frac{\partial \tilde{o}}{\partial \boldsymbol{\rho}}$;
        $\tilde{o} \leftarrow \max_i \mathbb{E}_{\tilde{\mathbf{b}}_\tau \sim P(\lambda,\boldsymbol{\rho})}[f^i(\mathbf{c}+\tilde{\mathbf{b}}_\tau) - f^i(\mathbf{c})]$;
        $o \leftarrow \max_i \mathbb{E}_{\mathbf{b} \sim P(\lambda,\boldsymbol{\rho})}[f^i(\mathbf{c}+\mathbf{b}) - f^i(\mathbf{c})]$;
    end
    if $o \leq 0$ then
        Return $\boldsymbol{\rho}$
    else
        while $\tilde{o} \leq 0$ do
            $\tau \leftarrow \frac{\tau}{2}$;
            $\tilde{o} \leftarrow \max_i \mathbb{E}_{\tilde{\mathbf{b}}_\tau \sim P(\lambda,\boldsymbol{\rho})}[f^i(\mathbf{c}+\tilde{\mathbf{b}}_\tau) - f^i(\mathbf{c})]$;
        end
    end
end
```

Algorithm 1: The proposed algorithm optimizing embedding costs to minimize detectability of a stego object with respect to a set of steganalyzers $\mathcal{F}^k$.

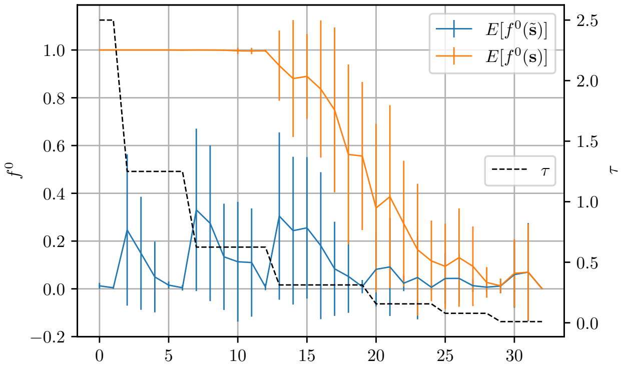

Figure 2: Effect of decreasing the temperature τ during optimization of a embedding costs for a given cover image. Is plotted on the left y-axis the average and variance of detectability (1 for stego class and 0 for cover class) given by classifier f^0 (for which $f^0(\mathbf{c}) = 0$) over 20 sampled simulated continuous stego object (blue plot where $\tilde{\mathbf{s}} = \mathbf{c}+\tilde{\mathbf{b}}_\tau$) and for 20 sampled simulated discrete stego object (orange plot, where $\mathbf{s} = \mathbf{c}+\mathbf{b}$), over the 33 steps of optimization on the x-axis.

4 EXPERIMENTAL EVALUATION

Backpack is below compared to ADV-EMB by each method implementing step 2 in the min max protocol (see Section 2.3). Note that due to the weakness described in Section 2.4 the ADV-EMB algorithm is computationally less expensive, since it is sufficient to optimize it against the last steganalyzer f^k, whereas Backpack needs to be optimized with respect to all classifiers $f \in \mathcal{F}^k$.[2]

4.1 Experimental settings

4.1.1 Images. The experiments use the JPEG version of the BossBase database [3] of size 512 × 512 in grayscale format and compressed with Quality Factor (QF) 100 and 75. All images are embedded using an embedding rate of 0.4 bits per non-zero AC DCT coefficient (bpnzAC) at each iteration of the algorithm.

4.1.2 Steganalysis. A proper evaluation of min max protocol (and its variants) requires two sets of steganalyzers and their overlap depends on who knows what. The first set of classifiers, $\mathcal{F}$ is available to Alice, who is running the min max protocol. In this work, this set contains all classifiers with XuNet architecture [32] (differing in weights). The second set, $\tilde{\mathcal{F}}$, of classifiers is available to Eve. In our experiments, $\tilde{\mathcal{F}}$ contains all classifiers with SrNet and XuNet architectures, classifiers trained with DCTR [13] or GFR [27] features. XuNet and SrNet were implemented in TensorFlow [1]. This experimental setup allows to investigate two different setups which practically express the assumptions that whether or not Alice knows which class of steganalyzers Eve uses.

At each iteration of the min max protocol, a new steganalyzer f^k is trained by classifying cover objects C and stego objects $\mathcal{S}^k$ created in previous iteration at the second step of the min max protocol. Steganalyzers are trained on full-size images of 512 × 512 coefficients, 2×4000 cover and stego objects for training, 2×1000 for validation set and using remaining 2 × 5000 to estimate error rates. The training database is shuffled after each epoch. In each batch, we apply data augmentation based on random mirroring and rotation of the batch images by 90 degrees. 280 epochs are used for training using Adam optimizer [18]. The configuration achieving the best validation accuracy is used as the result of training. XuNet, the classifier is trained starting with randomly initialized weights (zero mean Gaussian with standard deviation 0.01), initial learning rate is set to 0.001 and decreased after each 5000 steps to 0.9 times the current value. Remaining parameters of Adam are kept to default setting. The size of mini-batch is 32 (16 cover-stego pairs). The configuration of SrNet is the one proposed in the paper [6], except the training uses 280 epochs. The size of mini-batch is 16 (8 cover-stego pairs).

4.1.3 Optimization of embedding costs. Both compared methods requires initialization of embedding costs, for which those of J-UNIWARD [15] were used (this has been done in [29]). The ADV-EMB method for adjusting costs is implemented as described in Section 2.2. Backpack uses Adam [18] with a learning rate of 0.05 to optimize the embedding costs ρ in Algorithm 1. Gradients of expected error (Equation (11)) are computed with $k = 30$ samples until fourth iteration of min max protocol, with $k = 20$ samples until its eight iteration, and with $k = 10$ samples. Although a single sample is frequently sufficient, more samples improves the accuracy of predicted gradients and they can be calculated in parallel on the GPU in the same batch. However as min max protocol progresses, the gradient needs to be calculated with increasingly more models, which occupies the memory of GPU and therefore we had to progressively decrease the number of samples. The initial temperature was set to $\tau^0 = 10$. The error of steganalyzers was measured by Equation (4), which is the usual average error on cover and stego objects assuming equal prior probability of their occurrence (denoted as P_{err}). Since the goal of steganography is to be undetectable, higher value is better.

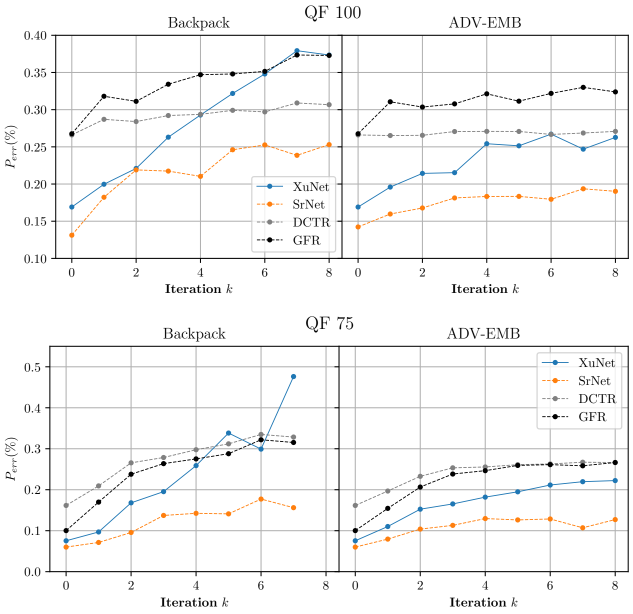

Figure 3: P_{err} of test sets w.r.t iterations of the protocol with QF 100 (top line) or QF 75 (bottom line), an embedding rate of 0.4 bpnzAC, cost initialized with J-UNIWARD and applied with our attack (left column) or ADV-EMB attack (right column). Assumed class of detectors is XuNet architecture, and real detectors are XuNet, SrNet, DCTR and GFR.

4.2 Discussion of results

QF	h_{emb}	P_{err} (%)			
		XuNet	SrNet	DCTR	GFR
100	J-UNIWARD ($k = 0$)	16.9	13.1	26.6	26.8
	ADV-EMB ($k = 8$)	26.5	18.9	26.5	32.4
	Backpack ($k = 8$)	**37.4**	**25.3**	**30.7**	**37.3**
75	J-UNIWARD ($k = 0$)	7.5	6.0	16.2	10.0
	ADV-EMB ($k = 7$)	22.0	10.7	26.7	25.8
	Backpack ($k = 7$)	**47.6**	**15.6**	**32.9**	**31.5**

Table 1: Values of P_{err} plotted in Figure 3 at $k = 0$ and $k = 7$ or $k = 8$ for QF 75 or QF 100, for both ADV-EMB and Backpack.

The main bulk of experimental results are presented in Figure 3 and in Table 1 showing error P_{err} of XuNet, SrNet, DCTR, and

[2] Note that due to the max function, it attacks in each iteration of Algorithm 1 a single steganalyzer from the set $\mathcal{F}^k$, but this single classifier is potentially different at every iteration.

GFR steganalyzers on testing data with respect to the iteration of min max protocol. The proposed Backpack method is clearly superior to the ADV-EMB. The P_{err} error of a XuNet steganalyzer trained after eight iterations is 37% on images created with the proposed method while it is 26% on those created by ADV-EMB (also after eight iterations) and 16% on those created by J-UNIWARD. This means that if [5] is considered by state of the art, the proposed method has improved by 11% (as measured by XuNet for which it has been optimized). These results are for JPEG 100 and in the optimistic case for Alice where she knows which type of steganalyzer Eve will use (but Eve optimizes her steganalyzer on Alice's stego images from her final iteration such that Kerckhoffs' principle is not violated). It is interesting to observe that even though Alice is not explicitly optimizing against SrNet, DCTR, and GFR steganalyzers, she is still improving the security of her steganograhic algorithm with respect to them, although the curve is not as steep as that for XuNet.[3]

The evolution of P_{err} also suggest that GFR steganalyzers relies on a similar type of information as XuNet (on JPEG images with QF 100), but DCTR and SrNet uses different type, as the improvement in the security is not as high.

Experimental results on JPEG images with QF 75 copy those on QF 100. Though there is an interesting difference in behavior of steganalyzer utilizing DCTR features. On QF 100 this steganalyzer is almost insensitive to the improvement in security with respect to XuNet, whereas on QF 75, it reflects the improvement. This suggests that at QF 100, it is detecting artifacts, which are not present at QF 75 and XuNet cannot see these artifacts. This might be caused by rounding artifacts described in [7].

Unlike ADV-EMB, Backpack does not contain any regularization minimizing the number of modified coefficients. It can be trivially added, for example by defining a prior on distribution of embedding changes, but it should be learned from data rather than added explicitly. Moreover, the min max protocol should theoretically correct too detectable embedding changes in subsequent iterations. Nevertheless the steady increase in steganographic security as measured by XuNet steganalyzers do not indicate that such regularization is needed.

5 CONCLUSION AND OVERVIEW

This paper framed *adversarial attacks* against steganalyzers into a perspective of the general steganographic problem, which is the minimization of non-additive distortion functions. It has shown that adversarial attacks can be seen as an optimization of an approximation of non-additive distortion function by its additive counterpart defined implicitly by costs of changing embedding coefficients.

The proposed method, called Backpack, relies on the fact that most state of the art steganalyzers, mainly those implemented by convolution neural networks, allow to calculate gradients of their output with respect to the input (by means of back-propagation). Backpack approximates discrete embedding changes by samples from Gumbel-Softmax distribution, which is nowadays a standard approach in machine learning field. It also uses differentiation of implicit function to effectively handle constraints on message length.

The experimental experimentally confirms the theoretical correctness of the approach. A security a steganographic scheme as measured by XuNet on 512 × 512 JPEG images compressed with quality factor 100 with payload of 0.4 bits per non-zero AC coefficient has increased to 37.3% whereas that of the previous state of the art known to authors was 26.5% under the same setting. Interestingly, although the steganographic algorithm was optimized with respect to XuNet steganalyzer, the security with respect to other steganalyzers realized by SrNet, GFR or DCTR features has increases as well.

Backpack's constraint on differentiable distortion functions seems to be limiting at first sight. Yet a line of works on black-box attacks and gradient obfuscation [2, 31] shows that adversarial attacks can be applied either to a differentiable surrogate of the true adversary, or without directly evaluating the gradient [22]. Such approaches are left for future works.

ACKNOWLEDGMENTS

This work has been funded in part by the French National Research Agency (ANR-18-ASTR-0009), ALASKA project: https://alaska.utt.fr, and by the French ANR DEFALS program (ANR-16-DEFA-0003). The authors acknowledge the support of the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" and a support by Czech Ministry of Education 19-29680L. This work was granted access to the HPC resources of IDRIS under the allocation 2019-AD011011259 made by GENCI (Grand Equipement National de Calcul Intensif).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. 265–283.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*. PMLR, 274–283.

[3] Patrick Bas, Tomáš Filler, and Tomáš Pevný. 2011. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In *International Workshop on Information Hiding*, Vol. 6958, LNCS. Springer Berlin Heidelberg, 59–70.

[4] Solène Bernard, Patrick Bas, John Klein, and Tomas Pevny. 2020. Explicit optimization of min max steganographic game. *IEEE Transactions on Information Forensics and Security* 16 (2020), 812–823.

[5] Solène Bernard, Tomás Pevnỳ, Patrick Bas, and John Klein. 2019. Exploiting adversarial embeddings for better steganography. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*. 216–221.

[6] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. 2018. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 14, 5 (2018), 1181–1193.

[7] Jan Butora and Jessica Fridrich. 2019. Reverse JPEG compatibility attack. *IEEE Transactions on Information Forensics and Security* 15 (2019), 1444–1454.

[8] Tomáš Filler and Jessica Fridrich. 2010. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security* 5, 4 (2010), 705–720.

[9] Tomáš Filler, Jan Judas, and Jessica Fridrich. 2010. Minimizing embedding impact in steganography using trellis-coded quantization. In *Media forensics and security II*, Vol. 7541. International Society for Optics and Photonics, 754105.

[10] Jessica Fridrich and Tomas Filler. 2007. Practical methods for minimizing embedding impact in steganography. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505. International Society for Optics and Photonics, 650502.

[11] Quentin Giboulot, Rémi Cogranne, and Patrick Bas. 2021. Detectability-based JPEG steganography modeling the processing pipeline: the noise-content trade-off. *IEEE Transactions on Information Forensics and Security* Early access, Early access (Jan. 2021), Early access. https://doi.org/10.1109/TIFS.2021.3050063

[12] Linjie Guo, Jiangqun Ni, Wenkang Su, Chengpei Tang, and Yun-Qing Shi. 2015. Using statistical image model for JPEG steganography: Uniform embedding

[3] Ref. [4] has demonstrated min max protocol to optimize with respect to steganalyzers of different architectures. This optimization is a bit more complicated, as their outputs needs to be carefully calibrated, and also computationally more complex. From these reasons, this experiment was avoided here and left to a future work. Theoretically, we do not see any reason why the proposed method should not work.

revisited. *IEEE Transactions on Information Forensics and Security* 10, 12 (2015), 2669–2680.

[13] Vojtěch Holub and Jessica Fridrich. 2015. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security* 10, 2 (2015), 219–228.

[14] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1–13.

[15] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1.

[16] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. https://arxiv.org/abs/1611.01144

[17] Christy Kin-Cleaves and Andrew D Ker. 2020. Simulating Suboptimal Steganographic Embedding. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*. 121–126.

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

[19] Sarra Kouider, Marc Chaumont, and William Puech. 2013. Adaptive steganography by oracle (ASO). In *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[20] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. 2014. A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 4206–4210.

[21] Bin Li, Ming Wang, Xiaolong Li, Shunquan Tan, and Jiwu Huang. 2015. A strategy of clustering modification directions in spatial image steganography. *Information Forensics and Security, IEEE Transactions on* 10, 9 (2015), 1905–1917.

[22] Thibault Maho, Teddy Furon, and Erwan Le Merrer. 2020. SurFree: a fast surrogate-free black-box attack. *arXiv preprint arXiv:2011.12807* (2020).

[23] Frans A Oliehoek, Rahul Savani, Jose Gallego-Posada, Elise Van der Pol, Edwin D De Jong, and Roderich Groß. 2017. GANGs: Generative adversarial network games. *arXiv preprint arXiv:1712.00679* (2017).

[24] T. Pevny, T. Filler, and P. Bas. 2010. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In *Information Hiding 2010* (Calgary, Canada).

[25] Tomas Pevny and Andrew D Ker. 2018. Exploring non-additive distortion in steganography. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. 109–114.

[26] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. 2015. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security* 11, 2 (2015), 221–234.

[27] Xiaofeng Song, Fenlin Liu, Chunfang Yang, Xiangyang Luo, and Yi Zhang. 2015. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. ACM, 15–23.

[28] Théo Taburet, Patrick Bas, Wadih Sawaya, and Rémi Cogranne. 2020. JPEG Steganography and Synchronization of DCT Coefficients for a Given Development Pipeline. In *ACM Workshop on Information Hiding and Multimedia Security*. Denver, United States. https://hal.archives-ouvertes.fr/hal-02553023

[29] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang. 2019. CNN-Based Adversarial Embedding for Image Steganography. *IEEE Transactions on Information Forensics and Security* 14, 8 (2019), 2074–2087. https://doi.org/10.1109/TIFS.2019.2891237

[30] W. Tang, S. Tan, B. Li, and J. Huang. 2017. Automatic Steganographic Distortion Learning Using a Generative Adversarial Network. *IEEE Signal Processing Letters* 24, 10 (2017), 1547–1551. https://doi.org/10.1109/LSP.2017.2745572

[31] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347* (2020).

[32] Guanshuo Xu. 2017. Deep convolutional neural network to detect J-UNIWARD. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. 67–73.

[33] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Shi. 2020. An Embedding Cost Learning Framework Using GAN. *IEEE Transactions on Information Forensics and Security* 15 (2020), 839–851. https://doi.org/10.1109/TIFS.2019.2922229