Backpack: a Backpropagable Adversarial Embedding Scheme

Solène Bernard, Patrick Bas, John Klein, Tomáš Pevný

Backpack: a Backpropagable Adversarial Embedding Scheme

Solène Bernard, Patrick Bas, *Senior Member, IEEE,* John Klein,and Tomáš Pevný, *Member, IEEE,*

***Abstract*—A $\min\max$ protocol offers a general method to automatically optimize steganographic algorithm against a wide class of steganalytic detectors. The quality of the resulting steganograhic algorithm depends on the ability to find an "adversarial" stego image undetectable by a set of detectors while communicating a given message. Despite $\min\max$ protocol instantiated with ADV-EMB scheme leading to unexpectedly good results, we show it suffers a significant flaw and we present a theoretically sound solution called Backpack. Extensive experimental verification of $\min\max$ protocol with Backpack shows superior performance to ADV-EMB, the generality of the tool by targeting a new JPEG QF100 compatibility attack and further improves the security of steganographic algorithms.**

***Index Terms*—Steganography, Steganalysis, Distortion function, Adversarial attacks**

I. Introduction

Steganography is the art of covertly communicating secret messages inside innocuous-looking objects. Steganalysis solves the opposite problem of detecting the presence of messages hidden. Steganographers and steganalysts have antagonistic goals, which makes game theory particularly well suited to model their interaction and it is deeply engraved into the history of the field. New steganographic methods have been created to become undetected by increasingly sophisticated steganalysis detectors and the other way around.

While steganalysis has observed a lot of automation driven by the adoption of increasingly more powerful machine learning techniques, most steganographic techniques are derived *heuristically*. A small list of exceptions are MiPod family [1], [2], [3], [4], Natural Steganography [5], Model-Based steganography [6], and ASO [7], but none of them is general enough to optimize steganography against arbitrary class of detectors.

If game theory matches so well the steganographer vs steganalyst competition, why steganography cannot be automatized by finding the equilibrium of the game modeling interaction between two parties? Until very recently, the main problem was the size of the game.[1] Practically, interesting games were simply too big to find a solution in a reasonably finite computational time. A $\min\max$ protocol [8], [9] has been proposed to find an approximate solution for the steganographic game exploiting a well-known double-oracle algorithm for approximately solving large games [10]. The protocol is instantiated in [8] with steganographic detectors based on CNNs (or their variants) and ADV-EMB algorithm [11] exploiting those detectors, demonstrated an increase of security of steganographic algorithms [8] that was not thought to be possible.

The $\min\max$ protocol (double-oracle algorithm), starts by solving a very small game, where the strategy set of each player contains exactly one strategy. In each iteration, the strategy set of each player is extended by their *best responses* to the current solution of the game (better *best responses* leads to a better solution). For steganalysts, finding the best response corresponds to creating a new detector by training a CNN. For steganographers, this amounts to embedding messages into images such that the resulting stego images are undetectable by a set of detectors created in previous iterations. In the rest of the paper (see section II), we show that since the ADV-EMB scheme can attack at most one detector at a time, it provides a very weak *best response* and hence decreases the quality of the solution found by the $\min\max$ protocol. We, therefore, propose a new algorithm called *Backpack*, which does not suffer the problems of ADV-EMB. Experimental results confirm that solutions found by $\min\max$ protocol are better with Backpack than with ADV-EMB, resulting in a more secure steganographic algorithm.

Backpack can be regarded as an adversarial attack against a fixed set of steganographic detectors, since it tries to create an image communicating the secret message while being undetectable against all of them. Its main advantage is that it is general as it can be used against any detector differentiable with respect to the input image while being relatively lightweight. The last property is very important, as for example a greedy variant of Syndrom Trellis Codes [12] is even more general than Backpack, but suffers from a high computational complexity preventing it to be effectively used inside $\min\max$ protocol. An efficient adversarial embedding scheme is consequently a key ingredient of $\min\max$ protocol providing the best response and vice-versa, an adversarial scheme without $\min\max$ protocol (or its variant solving the game) does not provide any convergence guarantees.[2]

A. Contributions of the paper

Compared to the work presented in [14], this paper provides necessary supplementary materials, including :

[1]By the size of the game, it is understood the number of steganographer's strategies (any function assigning costs to pixels or more generally embedding a message into a given cover object) and steganalyst's strategies (represented by any function from the space of images to $\{\text{cover}, \text{stego}\}$) .

[2]ASO [7] has proposed an attack targeted against an ensemble of linear classifiers utilizing SPAM [13] features, but it has lacked the $\min\max$ protocol, hence the algorithm did not converge with an increasing number of iterations.

1) A deeper analysis of the drawbacks of the ADV-EMB scheme, (see II.C)
2) A more detailed presentation of the principle of stochastic gradient optimization of additive approximation of non-additive distortion function, (see IV)
3) A comparison with an alternative formulation of ternary embedding changes based on the double-tangent approximation proposed for generative steganography (see V.B)
4) Comparisons between ADV-EMB and Backpack for an extended set of adversaries, (see V.C), this is an extremely important point which shows that, contrary to ADV-EMB, the iterative property of backpack enables to attack a set of adversaries. This again demonstrates that solving larger games where both players have larger and more powerful strategies leads to better solutions / more secure steganographic algorithms.
5) We further demonstrate the flexibility of Backpack by optimizing steganographic methods against recently proposed JPEG QF100 compatibility detector [15]. While JPEG QF100 might seem like a niche, it is very popular as it represents 16% of images uploaded to Flickr [16] by a range of very popular camera models.
6) A covariance analysis of the different DCT modes (see VII)
7) A sensitivity analysis to payload mismatch (see VII).
8) A sensitivity analysis to cover source mismatch (see VII).
9) A discussion about the stopping criterion.
10) An analysis of the behavior of Backpack for steganography in the spatial domain (see VII).
11) Companion codes and materials which can be used to either run the attack or use the trained adversaries available at https://gitlab.univ-lille.fr/solene.bernard/backpack .

This contribution is organized as follows. The next section II recalls two important prior arts (ADV-EMB and $\min\max$ protocol) whose combination achieves (previous) state-of-the-art steganography security performances. Section III presents the necessary background needed to design an adversarial attack in the context of steganography by cover modification: relationships between embedding costs and embedding probabilities, compliance with the entropy constraint, and definition of the objective function. It formalizes the problem we tackle, i.e. the computation of the gradient of expected detectability of stegos through function f w.r.t. the embedding costs. Section IV presents step-by-step how to calculate the gradient of a differentiable steganalyzer with respect to embedding costs thanks to the softmax Gumbel distribution and how to use such gradients to jointly deceive several classifiers. The experimental section V shows the effect of the proposed scheme on the security of the obtained embedding costs and analyzes the impact of the adversary on the embedding strategy.

B. Notations

In the sequel, letters in bold are used to represent vectors. The corresponding nonbold letters are used for vector elements. The calligraphic letters are used for sets. Cover and stego objects are respectively denoted as $\mathbf{x} = (x_i)^n$ and $\mathbf{y} = (y_i)^n$ where n is the number of pixels of the image. We use $\mathbf{z} = (z_i)^n$ to denote the stego objects that will be communicated by Alice. Note that $\mathbf{z}$ is a special type of $\mathbf{y}$. The corresponding sets are denoted as $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ respectively. $\omega \in \{0, 1\}$ denotes the class of an object which is either cover ($\omega = 0$) or stego ($\omega = 1$). N is the number of objects in the database.

More specifically, the next section uses the following additional notations. A steganographic algorithm is any pair of functions $h_{\text{emb}}(\mathbf{x}, \mathbf{m}, \mathbf{k}) : \mathcal{I} \times \mathcal{M} \times \mathcal{K} \to \mathcal{I}$ and $g_{\text{ext}}(\mathbf{x}, \mathbf{k}) : \mathcal{I} \times \mathcal{K} \to \mathcal{M}$ for which it holds that $g_{\text{ext}}(h_{\text{emb}}(\mathbf{x}, \mathbf{m}, \mathbf{k}), \mathbf{k}) = \mathbf{m}$ for all $\mathbf{m} \in \mathcal{M}$, $\mathbf{k} \in \mathcal{K}$, and $\mathbf{x} \in \mathcal{I}$. Spaces $\mathcal{I}$, $\mathcal{M}$, and $\mathcal{K}$ are respectively the space of all images, messages, and keys.

Furthermore, a steganographic detector is any function $f(\mathbf{x}) : \mathcal{I} \to \{\text{cover}, \text{stego}\}$, although it is more convenient to assume $f(\mathbf{x}) : \mathcal{I} \to \mathbb{R}$ and $\mathbf{x}$ is assigned to stego class if the output is greater than some threshold t.

II. Prior art on adversarial embedding

In this section, we present the two main ingredients necessary to implement a steganographic adversarial embedding scheme potentially offering high practical security. The first ingredient is the embedding adversarial scheme designed to fool the steganalyzer, the second one is a protocol that can be used to iterate through retraining and generating new stego images. The last paragraph highlights the inherent problem associated to the presented adversarial scheme.

A. The ADV-EMB scheme

The ADV-EMB scheme presented in [11] is the first adversarial embedding scheme proposed in steganography and it is inspired by the Fast Gradient Sign Method (FGSM) attack designed to generate adversarial contents. This heuristic method updates embedding costs coming from a reference method like J-Uniward [17] or UERD [18] according to the sign of the gradient of the classifier f. This cost modulation, applied on a subset of the image coefficients, is meant to move the stego image toward the detection region of the Cover class. The size of the subset is computed to be close to the border of the detection region and to avoid going too far in the Cover detection region, which would make the Stego image detectable after retraining. The updating rule for a cost ρ_i^+ associated with a $+1$ embedding change for coefficient i is given by:

$$\rho_i^{+,new} = \begin{cases} \rho_i^+ / \alpha & \text{if } \frac{\partial f}{\partial y_i}(\mathbf{y}) < 0, \\ \rho_i^+ & \text{if } \frac{\partial f}{\partial y_i}(\mathbf{y}) = 0, \\ \rho_i^+ \alpha & \text{if } \frac{\partial f}{\partial y_i}(\mathbf{y}) > 0, \end{cases} \tag{1}$$

where α is a parameter that the authors empirically set to 2.

B. The $\min\max$ protocol

Paper [8] proposes a $\min\max$ protocol based on the following observations: (i) for a steganographer aware of the steganalyst's model, an adversarial embedding scheme can be chosen to adapt the distortion function (attaching a cost to

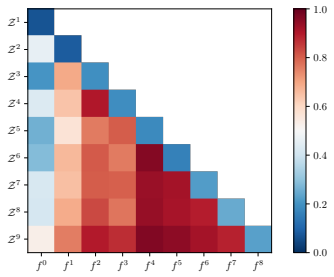

Fig. 1: Average detectability given by each classifier f^j (columns) evaluated on each adversarial stego database $\mathcal{Z}^i$ (rows) as part of $\min\max$ protocol with ADV-EMB for JPEG images at QF 75 and payload 0.4 bpnzAC, with J-Uniward as initial costs and XU-Net detectors. The blue color is for images detected as cover (the probability of stego class is below 0.5), whereas red is for images classified as stego.

each image coefficient) to avoid the detection and (ii) the steganalyzer can react by training a new classifier to detect the new technique. It leads to an endless game between two players with antagonistic goals. The outcome of the game relies on the pair of actions associated with both rivals. In other words, each player wonders "How can I anticipate what move will be played so that I behave optimally." This is the exact context of competitive games. The contribution of [8] consists in introducing game theory notions to solve the steganographic game. Defining the steganographic problem employing Game Theory allows defining an optimal steganographic algorithm (an embedding function) as a solution. Ref. [8] proposes to solve $\min\max$ equilibrium instead of the more common Nash equilibrium, which leads to more stable optimization. The same reference also proposes to replace optimization over infinite sets by a sequence of optimization over finite sets as used in the double-oracle algorithm [10].

Note that practically, this iterated game can be simulated on Alice's side, as she can alternate between building new stego images trying to defeat previously trained adversaries; and training an adversary, which is trained to distinguish the new stego images from the cover images, at the end of each iteration of the protocol. Consequently, each iteration consists of the creation of (i) new stego images minimizing detectability w.r.t. the strongest opponent over a finite number of detectors and (ii) a new detector to identify the new stegos. By doing so, it is shown that the embedding function converges and the steganographic security is improving. The experiments in [8] use previously proposed ADV-EMB to deceive a single detector, which is shown later has a pathological behavior, which limits the $\min\max$ protocol.

C. $\min\max$ *protocol combined with ADV-EMB*

The main issue of ADV-EMB is that it is meant to defeat only one classifier, and combined with the $\min\max$ protocol, ADV-EMB consequently only defeats the latest classifier. This drawback is illustrated in Figure 1, which shows after each iteration the detectability of the stego contents for each trained classifier.

We can notice that the low detectabilities are associated only with the last trained classifier, the previous ones being still able to correctly classify stego contents, even before retraining. It is non-surprising since ADV-EMB tries to defeat only one detector and ignores the other ones. Even if the $\min\max$ protocol leverages the best-generated stego contents to train a suited detector, ADV-EMB is not ideal since it does not take into account weaker but important classifiers before generating the stego content. One of the goals of this paper is to develop a new attack targeting a whole set of classifiers, this embedding will then unleash the ability of the $\min\max$ protocol to generate more efficient adversaries.

III. Problem formulation

Practically, steganography by cover modification can be simulated by drawing a stego simulating the impact of the embedding reached by an optimal coding function. It is done by drawing independently a modification b_i inside a pre-defined set of candidate integers $\mathcal{B}$[3] that will be added to each image coefficient indexed by i according to a discrete distribution defined by change rates $(\pi_i^j)_{j\in\mathcal{B}}$. The probability distributions of each image coefficient are directly obtained from the costs ρ_i^j and the size of the message $|\mathbf{m}|$ [19].

For a given size of message $|\mathbf{m}|$ (in bits) and an additive distortion function described by its cost map $\boldsymbol{\rho}$, a theoretical result gives the distribution of stego images provided by the optimal coding function. The probability distribution of stego modifications $P_{\mathbf{b}}\left(\begin{bmatrix} j_1 & .. & j_n \end{bmatrix}|\boldsymbol{\rho},\lambda\right) = \prod_{i=1}^{n} P_{b_i}(b_i = j_i|\boldsymbol{\rho},\lambda)$ gives independent probability distributions P_{b_i} over the image coefficients (thanks to the additivity property of the distortion function which is assumed to hold). The probability of modification of a cover coefficient indexed by i by value j is equal to

$$\pi_i^j = P_{b_i}(b_i = j|\boldsymbol{\rho},\lambda) = \frac{e^{-\lambda\rho_i^j}}{\sum_{k\in\mathcal{B}} e^{-\lambda\rho_i^k}} = p_i^j(\boldsymbol{\rho},\lambda), \qquad (2)$$

where λ is tuned such that the entropy of the probability distribution $P_{\mathbf{b}}$ is equal to the length of the message, i.e

$$H(\boldsymbol{\pi}) = -\sum_{i=1}^{n}\sum_{j\in\mathcal{B}} \pi_i^j \log \pi_i^j = |\mathbf{m}|. \qquad (3)$$

Finally, λ is itself a function of $\boldsymbol{\rho}$ and $|\mathbf{m}|$, because of the entropy constraint of Equation (3). But it exists no explicit expression of this function. Because we do want to emphasize the fact that λ depends on those two variables, we will sometimes write λ as $\lambda = \Lambda(\boldsymbol{\rho},|\mathbf{m}|)$. Solving equation $H(P_{\mathbf{b}}(.|\boldsymbol{\rho},\lambda)) = |\mathbf{m}|$ in order to find λ is usually achieved by binary search.

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 $\mathbf{y} = \mathbf{x} + \mathbf{b}$ by a non-additive distortion[4] function $f(\mathbf{y})$ (read steganalyst), where $\mathbf{b} \sim P_{\mathbf{b}}(.|\boldsymbol{\rho},\lambda)$ and f being almost

[3] for ternary embedding $b_i \in \mathcal{B} = \{-1, 0, 1\}$.

[4] note that on one side the adversary, i.e. the classifier, is non-linear hence non-additive. Even if Backpack is optimized against a set of non-additive adversaries, it leverages additive costs to embed in practice.

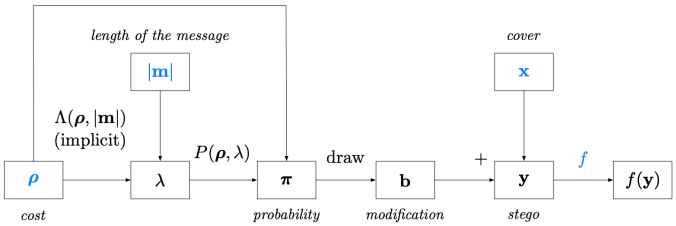

Fig. 2: The steganography by cover modification pipeline.

everywhere differentiable with respect to $\mathbf{y}$. We, therefore, focus on *simulation* of embedding changes. The stego object $\mathbf{y} = \mathbf{x} + \mathbf{b}$ is a realization of a random variable and the global corresponding pipeline is show in Fig. 2. Under this setting, our goal is to minimize the expected detectability over all possible stego objects written as :

$$\arg\min_{\boldsymbol{\rho},\lambda} \mathbb{E}_{\mathbf{b}\sim P_{\mathbf{b}}(.|\boldsymbol{\rho},\lambda)}[f(\mathbf{x}+\mathbf{b})], \tag{4}$$

subject to the entropy constraint:

$$H(P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho},\lambda)) = |\mathbf{m}|. \tag{5}$$

We want to tackle the above problem using a classical optimization method to remove all heuristics. Here we propose to use gradient descent with respect to $\boldsymbol{\rho}$. However, its use for this problem is not straightforward for two reasons: first, the optimization problem contains a constraint on the entropy; second, the exact computation of the gradient of the expectation of f with respect to $\boldsymbol{\rho}$ is prohibitively expensive. To compute it exactly, one would need to sum over the support of all stego images (for a given cover), which has a complexity of order $|\mathcal{B}|^n$ (recall that $|\mathcal{B}|$ is the cardinality of embedding changes and n is the number of coefficients that can be modified during embedding).

We are therefore interested in finding a computable value of:

$$\nabla_\rho \mathbb{E}_{\mathbf{b}\sim P_{\mathbf{b}}(.|\boldsymbol{\rho},\lambda)}[f(\mathbf{x}+\mathbf{b})], \tag{6}$$

that is, finding an *approximation* of (6) that would be sufficiently accurate while being computationally cheap.

For reasons to be unveiled shortly after (section IV-A), let us first temporarily focus on gradient computation for a fixed realization $\mathbf{b}$ of modifications. This amounts to back-propagating through the *forward pipeline* shown in Fig. 2, which contains all the successive operations made from the cost $\boldsymbol{\rho}$ to obtain a stego image $\mathbf{y}$.

The computation of the gradient can be computed by the following chain rule if all functions are differentiable:

$$\nabla_{\boldsymbol{\rho}} f(\mathbf{y}) = \nabla_{\mathbf{y}} f(\mathbf{y}) \cdot \frac{\partial \mathbf{y}}{\partial \mathbf{b}} \cdot \frac{\partial \mathbf{b}}{\partial \boldsymbol{\pi}} \cdot \frac{d\boldsymbol{\pi}}{d\boldsymbol{\rho}}. \tag{7}$$

For clarity, we use the conventional indexation of Jacobian matrices and vector $\nabla_{\boldsymbol{\rho}} f(\mathbf{y})$ is thus a line vector. The last Jacobian matrix of the chain rule is the total derivative of $\boldsymbol{\pi}$ w.r.t. $\boldsymbol{\rho}$. It is necessary because $\pi_i^j = p_i^j(\boldsymbol{\rho}, \lambda)$ is a function two dependant variables, because $\lambda = \Lambda(\boldsymbol{\rho}, |\mathbf{m}|)$.

Given the value of the total derivative, this gradient depends on $\frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\rho}}$, $\frac{\partial \boldsymbol{\pi}}{\partial \lambda}$ and $\frac{\partial \lambda}{\partial \boldsymbol{\rho}}$.

A considerable difficulty is the computation of the gradient of the modifications b_i with respect to the probabilities π_i^j. Because b_i is an integer drawn according to the probabilities $(\pi_i^j)_{j\in\mathcal{B}}$ and there is no direct expression of the gradient. A second issue comes from the computation of λ with respect to the cost $\boldsymbol{\rho}$. Because we saw that no explicit expression of Λ exists, the gradient is not straightforward.

The next section, which is the core of this contribution, proposes solutions to tackle this problematic differentiation and overcome the aforementioned difficulties.

IV. Differentiable steganography

The idea we propose is to optimize the cost $\boldsymbol{\rho}$ w.r.t. the detectability of a detector f by a standard optimization. We will use gradient descent on the costs in order to decrease $f(\mathbf{y})$. In the usual gradient descent setting, we need to compute the gradient $\nabla_{\boldsymbol{\rho}}\mathbb{E}[f(\mathbf{y})]$, and for a given $\boldsymbol{\rho}$, we would update it by the following formula

$$\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} - \alpha \nabla_\rho \mathbb{E}_{\mathbf{b}\sim P_{\mathbf{b}}(.|\boldsymbol{\rho},\lambda)}[f(\mathbf{x}+\mathbf{b})], \tag{8}$$

where $\alpha > 0$ is the step size of the gradient descent. We will see that several approximations and relaxations are necessary to achieve gradient descent to learn cost maps.

A. Re-parametrization trick

The first idea is to exploit the parameterization of the discrete (multinomial) distributions P_{b_i} so that we can compute the sample b_i as a deterministic function B of $\boldsymbol{\pi}_i = (\pi_i^j), j \in \mathcal{B}$ and a vector of independent random variables $\mathbf{r}_i$ drawn from another distribution R. We have $b_i = B(\boldsymbol{\pi}_i, \mathbf{r}_i)$. The path-wise gradients from b_i to $\boldsymbol{\pi}_i$ can then be computed without encountering any stochastic nodes. This is a very general scheme, and many pairs of $(B, \mathbf{r})$ could fit for drawing according to a discrete distribution.

Decoupling modification probabilities from the stochasticity of the randomly drawn modifications will allow computing the gradient of DCT coefficients y_i w.r.t. the distribution parameters π_i^j, as it is shown in section IV-B1. This reparametrization allows also to permute the gradient and the expectation:

$$\begin{aligned}\nabla_{\boldsymbol{\rho}} \mathop{\mathbb{E}}_{\mathbf{b}\sim P_{\mathbf{b}}(\mathbf{b}|\boldsymbol{\rho},\lambda)}[f(\mathbf{x}+\mathbf{b})] &= \nabla_{\boldsymbol{\rho}} \mathop{\mathbb{E}}_{\mathbf{r}\sim R}[f(\mathbf{x}+B(\boldsymbol{\pi},\mathbf{r}))] \\ &= \mathop{\mathbb{E}}_{\mathbf{r}\sim R}[\nabla_{\boldsymbol{\rho}} f(\mathbf{x}+B(\boldsymbol{\pi},\mathbf{r}))]\end{aligned} \tag{9}$$

where $\boldsymbol{\pi}$ depends on the variable $\boldsymbol{\rho}$. The main advantage of the re-parametrization is that the Monte-Carlo (MC) estimate of the gradient in Equation (6)

$$\mathop{\mathbb{E}}_{\mathbf{r}\sim R}[\nabla_{\boldsymbol{\rho}} f(\mathbf{x}+B(\boldsymbol{\pi},\mathbf{r}))] \approx \frac{1}{K}\sum_{\ell=1}^{K} \nabla_{\boldsymbol{\rho}} f(\mathbf{x}+B(\boldsymbol{\pi},\mathbf{r}_\ell)). \tag{10}$$

has lower variance than that of Equation (6). The number K of drawn samples $\mathbf{r}_1, \ldots, \mathbf{r}_K$, controls the trade-off between the variance of the estimate and the computational complexity. In the experiments presented in section V, K was set to the highest number that the GPU memory allowed.

Now the problem is focused on how to compute $\nabla_{\boldsymbol{\rho}} f(\mathbf{y}) = \nabla_{\boldsymbol{\rho}} f(\mathbf{x} + B(\boldsymbol{\pi}, \mathbf{r}))$, for fix random values $\mathbf{r}$. It is the subject of the next subsection.

B. Differentiable coefficient modifications

Because of the chain rule $\nabla_{\boldsymbol{\rho}} f(\mathbf{y}) = \sum_i \frac{\partial f(\mathbf{y})}{\partial b_i} \nabla_{\boldsymbol{\rho}} b_i$, we can focus on the value of $\nabla_{\boldsymbol{\rho}} b_i$. To simplify the notation, we can drop the index i. It means that, in the scope of this subsection, the modification b of a coefficient is a scalar with probability distribution described by a vector $\boldsymbol{\pi} = (\pi^j)_{j \in \mathcal{B}}$ of length $|\mathcal{B}|$. Likewise, $\boldsymbol{\rho}$ is $|\mathcal{B}|$-dimensional vector containing modification costs of a single coefficient.

The next paragraphs present two approaches to compute the gradient of Categorical distributions. Their respective qualities depend on the chosen pair (B, r). The canonical way to do so is to divide the interval $[0, 1]$ into $|\mathcal{B}|$ buckets of sizes $\boldsymbol{\pi}$ and then return the index of the bucket in which a random variable u with uniform distribution on $[0, 1]$ falls. This operation, called Stair, is a function of $\boldsymbol{\pi}$ and u. However, the derivative of Stair w.r.t. π_i^j is either equal to 0 or undefined, which means the gradient is not informative to gradient descent. We start by presenting the chosen differentiation for Backpack which relies on the Gumbel distribution and the softmax function. An alternative from [20], called Double-Tanh, is also presented.

1) Softmax Gumbel: Calculating the gradient of the expectation of a discrete probability distribution with respect to its parameters is a very well-studied problem. From the vast prior art, we have chosen the method [21] relying on the Gumbel distribution. This technique has the advantage of giving a general formula to draw samples according to any discrete distribution so that it can be used without a modification for n-ary coding, and its theoretical properties are well analyzed. It can be shown that discrete modifications can be drawn by sampling $\mathbf{g} = (g^j)_{j \in \mathcal{B}}$, which is a vector of independent entries sampled from the standard Gumbel distribution $G(0, 1)$, and applying the following deterministic function:

$$b = \mathrm{HG}(\boldsymbol{\pi}, \mathbf{g}) = \arg\max_{j \in \mathcal{B}} (g^j + \log \pi^j). \tag{11}$$

In the above function HG (called *Hardmax Gumbel*), the $\arg\max$ can be conveniently replaced by the softmax function:

$$\mathrm{softmax}\left(v^1, \ldots, v^n\right) = \frac{1}{\sum_{k=1}^n e^{v^k}} (e^{v^1}, \ldots, e^{v^n}),$$

which is a well-known approximation of $\arg\max$, as can be seen from

$$\lim_{\tau \to 0} \mathrm{softmax}\left(\frac{v^1}{\tau}, \ldots, \frac{v^n}{\tau}\right) = (0, 0 \ldots, 0, 1, 0, \ldots, 0),$$

where the 1 is on $\arg\max_i v^i$ position and τ is a *temperature* parameter controlling the smoothness of the approximation.

Replacing $\arg\max$ in Equation (11) by a softmax approximation with temperature leads to :

$$\tilde{b}_\tau = \mathrm{SG}_\tau(\boldsymbol{\pi}, \mathbf{g}) = \sum_{j \in \mathcal{B}} j \, \nu^j, \tag{12}$$

$$\text{with } \boldsymbol{\nu} = \mathrm{softmax}\left(\frac{\mathbf{g} + \log \boldsymbol{\pi}}{\tau}\right). \tag{13}$$

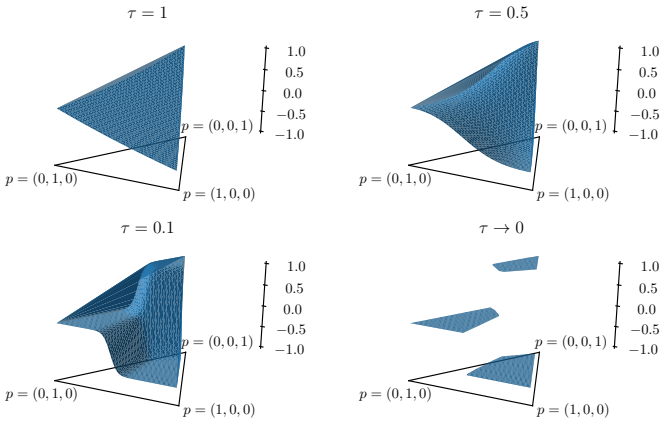

Fig. 3: For a given value of triplet $g = (g^{-1}, g^0, g^{+1})$ (where $g^j \sim G(0, 1)$ are independently drawn from Gumbel standard distribution), the value of the modification $\tilde{b}_\tau = \mathrm{SG}(p, g)$ is plotted the z-axis for all possible triplets of probabilities $p = (p^{-1}, p^0, p^{+1})$, and for 4 values of τ. The triplets are plotted in the trilinear coordinate system.

The gradient of the continuous modification $\tilde{b}_\tau$ w.r.t. $\boldsymbol{\pi}$ are easy to compute and have non-zero values. It can be conveniently plugged in the chain rule (7) although the resulting gradient is biased when $\tau > 0$.

Figure 3 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 $\boldsymbol{\pi}$.

2) Staircase and Double-$\tanh$ *functions:* A common approach to draw samples from a categorical distribution with probabilities $\boldsymbol{\pi} = (\pi^{-1}, \pi^0, \pi^{+1})$ is to draw a sample u from a uniform distribution $U(0, 1)$ and pass it through a *staircase* function defined as

$$b = \mathrm{Staircase}(\boldsymbol{\pi}, u) = \begin{cases} -1, & \text{if } u < \pi^{-1} \\ 0, & \text{if } \pi^{-1} \leq u < \pi^{-1} + \pi^0 \\ +1, & \text{otherwise} \end{cases} \tag{14}$$

For purposes of calculating gradients in (6), UT-GAN [20] replaces the non-differentiable discrete staircase function with this kind of continuous approximation, here extended[5] to the case of non-symmetric ternary costs:

$$\begin{aligned} \tilde{b}_\tau &= \mathrm{DT}_\tau(\boldsymbol{\pi}, u) \\ &= -\frac{1}{2} \tanh\left(\frac{\pi^{-1} - u}{\tau}\right) - \frac{1}{2} \tanh\left(\frac{\pi^{-1} + \pi^0 - u}{\tau}\right), \end{aligned} \tag{15}$$

where $\tau > 0$ is temperature parameter.

$\mathrm{DT}_\tau(\boldsymbol{\pi}, u)$ is differentiable with respect to π^j and the random number u is not in the gradient path. Similarly as for the Gumbel Softmax, the gradient of the continuous modification $\tilde{b}_\tau$ have non-zero values but the fact that they are not integers induces a bias in the computation of (10). The bias in question vanishes as $\tau \to 0$.

[5] In [20], the DT function is introduced in the $\pi^{-1} = \pi^{+1}$ case. By adding this constraint, and with an appropriate choice of parameter τ, it can be shown that the same DT function as in [20] can be retrieved from (15).

C. Using the gradient of an implicit function

Last step of chain rule (7) is about computing $\frac{d\boldsymbol{\pi}}{d\boldsymbol{\rho}}$. Because $\pi_i^j = p_i^j(\boldsymbol{\rho}, \Lambda(\boldsymbol{\rho}, |\mathbf{m}|))$, this Jacobian matrix is obtained from the total derivative formula as

$$\begin{aligned} \frac{d\boldsymbol{\pi}}{d\boldsymbol{\rho}} &= \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\rho}} \cdot \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\rho}} + \frac{\partial \boldsymbol{\pi}}{\partial \lambda} \cdot \nabla_{\boldsymbol{\rho}} \lambda \\ &= \frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\rho}} + \frac{\partial \boldsymbol{\pi}}{\partial \lambda} \cdot \nabla_{\boldsymbol{\rho}} \lambda \end{aligned} \tag{16}$$

Although function Λ is implicit, its gradient $\nabla_{\boldsymbol{\rho}} \Lambda(\rho, |\mathbf{m}|))$ can be computed. Recall that for a given $\boldsymbol{\rho}$, λ is a solution of an entropy constraint (Equation (3)), therefore it holds that

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

and therefore $H(P_{\mathbf{b}}(\boldsymbol{\rho}, \Lambda(\boldsymbol{\rho}, |\mathbf{m}|))) - |\mathbf{m}| = 0$. Applying total derivative to this equation as well gives :

$$\frac{d}{d\boldsymbol{\rho}} H(P_{\mathbf{b}}(\boldsymbol{\rho}, \Lambda(\rho, |\mathbf{m}|))) = \nabla_{\boldsymbol{\rho}} H(\boldsymbol{\pi}) \cdot \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\rho}} + \frac{\partial H(\boldsymbol{\pi})}{\partial \lambda} \nabla_{\boldsymbol{\rho}} \lambda = 0,$$

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

$$\nabla_{\boldsymbol{\rho}} \lambda = -\left(\frac{\partial H(\boldsymbol{\pi})}{\partial \lambda} \right)^{-1} \nabla_{\boldsymbol{\rho}} H(\boldsymbol{\pi}). \tag{17}$$

D. Final approximation of the gradient, with continuous modifications

In order to be able to compute $\frac{\partial \mathbf{b}}{\partial \boldsymbol{\pi}}$, Backpack uses the Softmax gumbel approximation[6] so that discrete modifications are no longer needed but instead smooth ones $\tilde{\mathbf{b}}_\tau$ controlled by a temperature τ can be used. The stego, obtained by the summation of the modifications to the cover are therefore denoted by $\tilde{\mathbf{y}} = \mathbf{x} + \tilde{\mathbf{b}}_\tau$. We can also show that $\frac{\partial \tilde{\mathbf{y}}^k}{\partial \tilde{\mathbf{b}}_\tau^k}$ equals to the identity matrix, because $\frac{\partial (x_i + b_i)}{\partial b_j} = [i = j]$. So we can remove it from the chain rule formula.

Combining Equation (9) with Equations (10), (7), (16) and (17) yields to a closed form expression for the gradient :

$$\begin{aligned} \nabla_{\rho} \mathop{\mathbb{E}}_{\tilde{\mathbf{b}}_\tau \sim P_{\tilde{\mathbf{b}}_\tau}(.|\boldsymbol{\rho}, \lambda)} [f(\tilde{\mathbf{y}})] \approx \left(\frac{1}{K} \sum_{k=1}^{K} \nabla_{\tilde{\mathbf{y}}^k} f(\tilde{\mathbf{y}}^k) \frac{\partial \tilde{\mathbf{b}}_\tau^k}{\partial \boldsymbol{\pi}} \right) \times \\ \left(\frac{\partial \boldsymbol{\pi}}{\partial \boldsymbol{\rho}} - \frac{\partial \boldsymbol{\pi}}{\partial \lambda} \left(\frac{\partial H(\boldsymbol{\pi})}{\partial \lambda} \right)^{-1} \nabla_{\boldsymbol{\rho}} H(\boldsymbol{\pi}) \right). \end{aligned} \tag{18}$$

For practical computation, this formula is not required, as it can be handled automatically by using the auto-differentiation capabilities implemented in most of the libraries dedicated to deep learning. Only Equation (17) is needed to specify the gradient of a non explicitly differentiable function Λ.

However, for the curiosity of the reader, we show below the explicit value, for a unique sample of stego, the value of $\frac{\partial f(\tilde{\mathbf{y}})}{\partial \rho_k^l}$, expressed from all computations in Table I.

[6]Using the Double-Tanh approximation is also possible but proved to lead to poorer performances, see V-B2.

E. Optimizing embedding costs

The theoretical analysis presented in the previous sections allows us to efficiently approximate the gradient of Equation (6) by estimating a gradient of its smooth approximation while complying to a constraint on the entropy. It allows to use the desired gradient descend method to minimize detectability with respect to all detectors in a set $\mathcal{F}^k = \left\{ f^i \right\}_{i=1}^k$ as needed in the k^{th} step of $\min\max$ protocol. However, due to the bias introduced by using continuous modifications, it would be unwise to solely plug our gradient estimate into the update rule (8) and perform descent. We need to find an algorithmic solution that mitigates the incurred bias and checks if smooth modifications based on gradients do optimize the detectability of integer-valued modifications.

The proposed algorithm with pseudocode shown in Algorithm 1 uses several continuous approximations of discrete embedding changes to optimize iteratively the embedding costs $\boldsymbol{\rho}$. It monitors the (signed) maximal detectability margin, i.e. the maximal observed difference between the stego-probabilities of a stego $\mathbf{x} + \mathbf{b}$ (integer-valued case) or $\mathbf{x} + \tilde{\mathbf{b}}$ (real-valued case[7]) and the corresponding cover $\mathbf{x}$. In each iteration, it checks if the margin of stego images with discrete modifications follows the same downward trend as the detectability of stego images with real-valued modifications. If the detectability margin of integer-valued stegos is negative, the algorithm terminates; otherwise, it continues. If the detectability margin of stego images with continuous modifications is negative, the temperature is halved in order to gradually reduce the aforementioned bias. A negative maximal margin means that a stego has defeated all classifiers. Algorithm 2 summarizes the steps needed to compute the margins. Note that the first step of this algorithm uses a binary search to determine λ which is a common practice in steganography.

The progress of the proposed algorithm on minimizing the detectability of a single stego object against a single detector is shown in Figure 4. 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 the detectability of continuous approximations and that of actual stego objects. But as the algorithm iterates and temperature decreases, this difference becomes negligible because distributions $P_{\tilde{\mathbf{b}}}(.|\lambda, \boldsymbol{\rho})$ and $P_{\mathbf{b}}(.|\lambda, \boldsymbol{\rho})$ are getting closer and closer.

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

V. Global evaluation

In this section, Backpack is compared to ADV-EMB in terms of their ability to provide stegos that fool detectors as part of the $\min\max$ protocol (see section II-B). Note that due to the weakness described in section II-C the ADV-EMB algorithm is computationally less expensive, since it

[7]with $b_i = \text{sign}(\text{round}(\tilde{b_i}))$ and $\text{sign}(0) = 0$.

Data: Cover image $\mathbf{x}$, initial embedding costs ρ^0, initial τ^0, sample size K, classifier set $\mathcal{F}$
Result: Adversarial optimized embedding costs $\boldsymbol{\rho}$
$\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}^0$ // initial cost map
$\tau \leftarrow \tau^0$ // initial temperature
$\{(\tilde{o}_k, o_k) \leftarrow \mathsf{Margins}(f, \mathbf{x}, \boldsymbol{\rho}, |\mathbf{m}|, \tau, K) \text{ for } f \in \mathcal{F}\}$
$\tilde{o} \leftarrow \max_k \tilde{o}_k$ and $o \leftarrow \max_k o_k$
// Maximum detectability margins for respectively smooth stego and actual stego images
while *True* **do**
  while $\tilde{o} > 0$ *and* $o > 0$ **do**
    Update $\boldsymbol{\rho}$ by one step of gradient descend with $\frac{\partial \tilde{o}}{\partial \boldsymbol{\rho}}$
    $\{(\tilde{o}_k, o_k) \leftarrow \mathsf{Margins}(f, \mathbf{x}, \boldsymbol{\rho}, |\mathbf{m}|, \tau, K) \text{ for } f \in \mathcal{F}\}$
    $\tilde{o} \leftarrow \max_k \tilde{o}_k$ and $o \leftarrow \max_k o_k$
  end
  if $o \leq 0$ **then**
    Return $\boldsymbol{\rho}$ // The average detectability of real stego $\mathbf{x} + \mathbf{b}$ is below the detectability of cover: the attack has succeeded
  else
    while $\tilde{o} \leq 0$ **do**
      // The detectability of smooth stego $\mathbf{x} + \tilde{\mathbf{b}}$ is below the detectability of cover
      $\tau \leftarrow \frac{\tau}{2}$ // Decrease the temperature
      $\{(\tilde{o}_k, \_) \leftarrow \mathsf{Margins}(f, \mathbf{x}, \boldsymbol{\rho}, |\mathbf{m}|, \tau, K) \text{ for } f \in \mathcal{F}\}$
      $\tilde{o} \leftarrow \max_k \tilde{o}_k$
    end
  end
end

Algorithm 1: Proposed algorithm optimizing embedding costs to minimize detectability of a stego object with respect to a set of steganalyzers $\mathcal{F}$. The algorithm uses calls to function Margins, defined in Algorithm 2.

Data: Classifier f which outputs the probability of stego class, cover image $\mathbf{x}$, embedding cost map $\boldsymbol{\rho}$, message length $|\mathbf{m}|$, temperature τ and sample size K
Result: Approximate detectability margins $(\tilde{o}, o)$ of smooth/actual stego from cover w.r.t. to classifier f.
Function Margins(f, $\mathbf{x}$, $\boldsymbol{\rho}$, $|\mathbf{m}|$, τ, K)**:**
  $\lambda \leftarrow \Lambda(\boldsymbol{\rho}, |m|)$ // to satisfy the constraint on entropy
  $o \leftarrow 0$ and $\tilde{o} \leftarrow 0$ // margin initialization
  for ℓ *from* 1 **to** K // iterate over samples
  do
    for *each pixel* i **do**
      $\pi_i^j \leftarrow p_i^j(\boldsymbol{\rho}, \lambda), \forall j \in \mathcal{B}$ // probability to add change j to pixel i
      $g_i^j \sim G(0,1), \forall j \in \mathcal{B}$ // draws from Gumbel standard distribution
      $\tilde{b}_i \leftarrow \mathrm{SG}_\tau(\{g_i^j\}_{j\in\mathcal{B}}, \{\pi_i^j\}_{j\in\mathcal{B}})$ // compute smooth changes
      $b_i \leftarrow \mathrm{HG}(\{g_i^j\}_{j\in\mathcal{B}}, \{\pi_i^j\}_{j\in\mathcal{B}})$ // compute actual changes
      $\tilde{y}_i \leftarrow x_i + \tilde{b}_i$
      $y_i \leftarrow x_i + b_i$
    end
    $\tilde{o} \leftarrow \tilde{o} + f(\tilde{\mathbf{y}})$
    $o \leftarrow o + f(\mathbf{y})$
  end
  $\tilde{o} \leftarrow \tilde{o}/K - f(\mathbf{x})$
  $o \leftarrow o/K - f(\mathbf{x})$ // normalization for averaging and gap from classifier response to cover
end

Algorithm 2: Algorithm to approximate theoretical margins $\mathbb{E}_{\tilde{\mathbf{b}}_\tau \sim P_{\tilde{\mathbf{b}}}(.|\lambda,\boldsymbol{\rho})}[f(\mathbf{x} + \tilde{\mathbf{b}}_\tau)] - f(\mathbf{x})$ and $\mathbb{E}_{\mathbf{b}\sim P_{\mathbf{b}}(.|\lambda,\boldsymbol{\rho})}[f(\mathbf{x} + \mathbf{b})] - f(\mathbf{x})$ from K MC samples of smooth changes (depending on τ) or integer-valued changes.

is sufficient to optimize it against the last steganalyzer f^k, whereas Backpack needs to be optimized with respect to all classifiers $f \in \mathcal{F}^k$.[8]

A. General experimental settings

1) Images: The experiments in this paper use the JPEG version of the BossBase database [22] of size 512 × 512 in greyscale format and compressed with Quality Factor (QF) 100 and 75.

2) Steganalysis: The detectability of the steganographic scheme is measured by the error rate P_{err}. It is defined, for a given classifier f discriminating between cover $\mathcal{X}$ and stegos $\mathcal{Y}$, by:

$$P_{\mathrm{err}} = \frac{1}{2}\left(\mathbb{E}_{x\in\mathcal{X}}\left[\mathrm{I}\left[f(\mathbf{x}) \leq t\right]\right] + \mathbb{E}_{y\in\mathcal{Y}}\left[\mathrm{I}\left[f(\mathbf{y}) > t\right]\right]\right), \quad (19)$$

where $\mathrm{I}(.)$ denotes the Iverson function. Unless stated otherwise, the threshold t is set to 0.5 for detectors whose output is a score in $[0; 1]$. Since the goal of steganography is to be undetectable, a higher P_{err} value is better.

A proper evaluation through the $\min\max$ protocol requires two sets of steganalyzers. The first set of classifiers $\mathcal{F}$ is available to Alice, who runs the $\min\max$ protocol. Depending on the experiments, in this work, this set might contain either all classifiers with XuNet architecture [23] (differing in weights) noted $\mathcal{F} = \{\text{XuNet}\}$, or all classifiers with XuNet, SrNet [24] and EfficientNet [25] architectures (denoted by

[8]Due to the $\max$ function, a single steganalyzer from the set $\mathcal{F}^k$ is attacked at each step of Algorithm 1, but this classifier is potentially different at every step.

$\frac{\partial \tilde{b}_{\tau,k}}{\partial \pi_i^j}$	$[k=i]\frac{1}{\tau}\frac{z_i^j}{\pi_i^j}(j-\tilde{b}_{\tau,i}) \qquad \text{where } z_i^j = \frac{e^{\frac{g_i^j+\log \pi_i^j}{\tau}}}{\sum_{l\in\mathcal{B}} e^{\frac{g_i^l+\log \pi_i^l}{\tau}}}$
$\frac{\partial \pi_i^j}{\partial \rho_k^l}$	$\lambda[k=i]\pi_i^j\left(\pi_i^l-[l=j]\right)$
$\frac{\partial \pi_i^j}{\partial \lambda}$	$\pi_i^j\left(\sum_{m\in\mathcal{B}}\rho_i^m\pi_i^m-\rho_i^j\right)$
$\frac{\partial H(\boldsymbol{\pi})}{\partial \lambda}$	$-\sum_{i=1}^{n}\sum_{j\in\mathcal{B}}\frac{\partial \pi_i^j}{\partial \lambda}\left(1+\log \pi_i^j\right)$
$\frac{\partial H(\boldsymbol{\pi})}{\partial \rho_k^l}$	$-\sum_{i=1}^{n}\sum_{j\in\mathcal{B}}\frac{\partial \pi_i^j}{\partial \rho_k^l}\left(1+\log \pi_i^j\right)=-\lambda\sum_{j\in\mathcal{B}}\pi_k^j\left(\pi_k^l-[l=j]\right)\left(1+\log \pi_k^j\right)$

TABLE I: Analytic formulas of the gradients needed to compute $\nabla_{\boldsymbol{\rho}} f(\tilde{\mathbf{y}})$.

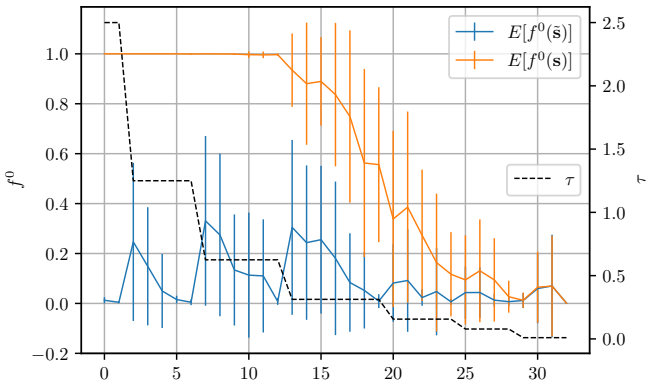

Fig. 4: Effect of decreasing the temperature τ during optimization of embedding costs for a given cover image. Is plotted on the left y-axis the average and variance of detectability (1 for stego class and 0 for cover class) given by classifier f^0 (for which $f^0(\mathbf{x})=0$) over 20 sampled simulated continuous stego objects (blue plot where $\tilde{\mathbf{s}} = \mathbf{x}+\tilde{\mathbf{b}}_\tau$) and for 20 sampled simulated discrete stego objects (orange plot, where $\mathbf{s}=\mathbf{x}+\mathbf{b}$), over 33 steps of optimization on the x-axis.

$\mathcal{F} = \{\text{XuNet}, \text{SrNet}, \text{EfficientNet}\}$). The second set, $\tilde{\mathcal{F}}$, of classifiers is available to Eve. In our experiments, $\tilde{\mathcal{F}}$ contains all deep-net type classifiers with XuNet, SrNet[9] and EfficientNet architectures as well as classifiers trained with DCTR [26], GFR [27] or JRM [28] features. This experimental setup allows investigating two different setups, which practically examine the case where Alice's assumption on Eve's classifiers is correct and when it is not.

At each iteration of the $\min\max$ protocol, a new steganalyzer f^k is trained by classifying cover objects and stego objects created in the previous iteration. Steganalyzers are trained on full-size images of 512×512 coefficients. We use 2×4000 cover and stego objects for training, 2×1000 for the validation set and the 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. 100 epochs are used for training using Adam optimizer [29]. The configuration achieving the best validation accuracy is used as the result of training.

For XuNet, the network is trained starting with randomly initialized weights (zero-mean Gaussian with standard deviation 0.01). Mini-batches have size 32 and contains 16 cover-stego pairs. The configuration of SrNet is the one proposed in [24]. It uses mini-batches containing 8 cover-stego pairs. The implementation of EfficientNet follows recommendations from [25]. We use version B0 and remove the stride in the first layer. It is initialized by a pre-trained network on ImageNet. No paired training is used, and the mini-batch size is 16.

The initial learning rate of respectively XuNet, SrNet and EfficientNet are 1e-3, 1e-3 and 5e-4. Its value during the training is changed by the scheduler `ReduceLROnPlateau`. The remaining parameters of Adam are kept to default setting.

3) Optimization of embedding costs: Both compared methods require initialization of embedding costs, for which those of J-Uniward [17] were used (this has been done in [30]). The ADV-EMB method for adjusting costs is implemented as described in section II-A. Backpack uses Adam [29] with different values of the step size of the gradient descent α to optimize the embedding costs ρ in Algorithm 1.

To make sure that the while loop in Algorithm 1 terminates, the total number of iterations cannot exceed a maximal number of steps. When $\mathcal{F}$ contains NNs of 3 architectures, this number is set to 500 until iteration 5 and to 2000 until the end of the protocol. When $\mathcal{F}$ contains NNs with XuNet architectures only, this number is set to 500 for whatever iteration.

Then, the number of samples needed to compute the gradient of expected error (Equation (10)) varies with the number of classifiers in $\mathcal{F}$. Although a single sample is frequently sufficient, more samples improve predicted gradients accuracy and can be calculated in parallel on the GPU in the same batch. However, as $\min\max$ protocol progresses, the gradients need to be calculated with increasingly more models, which occupies the memory of GPU and therefore, the number of samples has to be decreased progressively. It is why, when $\mathcal{F}$ is spanned by XuNet architectures only, we use $K=30$ samples until fourth iteration of $\min\max$ protocol, with $K=20$ samples until eighth iteration, and then with $K=10$ samples. Against three classifier architectures, at iteration 1,

[9]XuNet and SrNet were implemented in Pytorch.

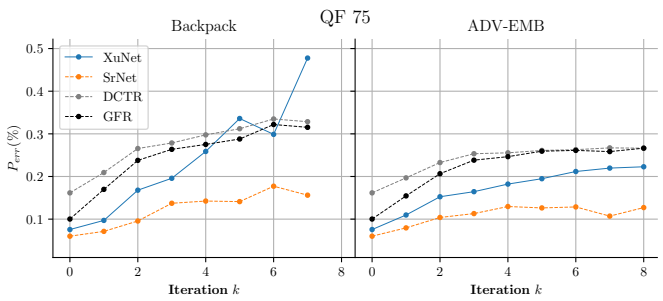

Fig. 5: P_{err} of test sets w.r.t iterations of the protocol with images with Quality Factor 75, an embedding rate of 0.4 bpnzAC, cost initialized with J-Uniward and applied with our attack Backpack with SGE (left) or ADV-EMB attack (right column). Assumed class of detectors is XuNet architecture, and real detectors are XuNet, SrNet, DCTR and GFR.

we estimated the gradient with $K = 5$ samples, at iteration 2, we used $K = 2$ samples, and starting from iteration 3, we used $K = 1$.

For all experiments, the initial temperature was set to $\tau^0 = 5$. The learning rate α is set to 0.5 for experiments with images at QF 75 and 0.1 for images at QF 100.

B. Results

1) Comparison between ADV-EMB and Backpack: For each attack (ADV-EMB and Backpack), we ran the protocol for 7 iterations with $\mathcal{F} = \{\text{XuNet}\}$ with images at QF 75 and payload 0.4 bpnzAC. At each iteration, the P_{err} of three other blind steganalysts (SrNet, classifiers based on DCTR and GFR features) are reported in Figure 5, and the final values are in Table II.

QF	h_{emb}	P_{err} (%) XuNet	SrNet	DCTR	GFR
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 II: Values of P_{err} plotted in Figure 5 at $k = 0$ and $k = 7$ or $k = 8$ for both ADV-EMB and Backpack.

We observe that the proposed Backpack method outperforms ADV-EMB. Starting from a P_{err} of 7.5% on costs obtained from J-Uniward, a XuNet steganalyzer trained after seven iterations with Backpack achieves a P_{err} of 47.6% while it achieves 22% using ADV-EMB.

2) Comparison between the two smoothing functions: Softmax-Gumbel or Double-Tanh.: We saw in Section IV-B that we can choose different differentiable functions to approximate zero-gradient functions: Softmax-Gubmel (SG) and Double-Tanh (DT). To compare the efficiency of those two smoothing functions to provide relevant gradient flow, we ran the protocols for images at QF 75 using either SG or DT. The results are shown for two iterations in Table III.

We can observe that both protocols produce stegos that are less detectable w.r.t. XuNet than J-Uniward. The protocol is therefore not only efficient for a specific smoothing function,

Iteration k	h_{emb}	P_{err} (%) XuNet
0	J-Uniward	7.5
1	Backpack with SG	9.7
	Backpack with DT	12.5
2	Backpack with SG	**16.8**
	Backpack with DT	14.4

TABLE III: Evolution of error rate of XU-Net for two protocols at QF 75 and embedding rate of 0.4 bpnzAC, the first one with Softmax-Gumbel (SG, see Equation (12)) and a second one with and Double-Tanh (DT, see Equation (15)), for two iterations.

but works for several ones, proving the generality of the protocol combined with Backpack. We can also observe that SG provides better security at iteration 2. In all the experiments we carried out, Backpack with SG turned out to always provide better performances after a couple of protocol iterations. Therefore, we recommend using this option.

C. Attacking several classifiers with Backpack

The main motivation behind Backpack is to design an attack that can jointly fool several detectors. The experiments described hereafter are meant to showcase this ability.

At first, let us focus on the ability to attack all classifiers in $\mathcal{F}^k = \{f^1, .., f^k\}$ from the past iteration of the $\min\max$ protocol when Alice uses only one network architecture. If we run the same experimental protocol as in Figure 1 but replacing ADV-EMB with Backpack, we obtain the detectability performances reported in Figure 6. We can observe that Backpack solves a major weakness of ADV-EMB highlighted in section III by defeating several detectors at the same time.

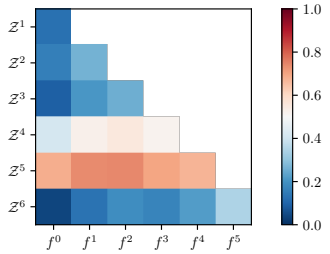

Fig. 6: For the protocol with Backpack with JPEG images at QF 75 and payload 0.4 bpnzAC, with J-Uniward and XU-Net, average detectability given by each classifier f^j (columns) evaluated on each adversarial stego database $\mathcal{Z}^i$ (rows). The blue color is for images detected as cover (the probability of stego class is below 0.5), whereas red is for images classified as stego.

Indeed, and contrary to ADV-EMB, until iteration 4, the images in the stego set issued by Backpack defeat on average all previous classifier $f^l, l \leq k$ (see the blue cells). With fixed experiment conditions, the task is more and more difficult and this is why iteration 5 does not provide as good results as lower iterations. At iteration 6, the number of optimization

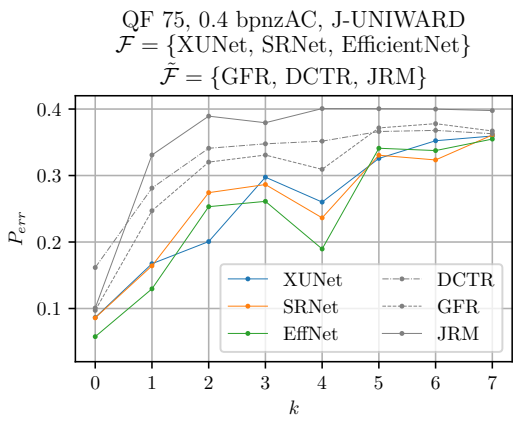

Fig. 7: Run of the protocol with Backpack for images at QF 75, payload of 0.4 bpnzAC, with assumed detectors XuNet, SrNet and EfficientNet, and additionally for real detectors GFR, DCTR and JRM.

steps of Backpack is increased, and a more powerful attack is obtained allowing to fool all six classifiers.

We now move to a more ambitious setting where Backpack has to optimize embedding costs by attacking classifiers with several architectures. On top of XuNet, we also add SrNet and Efficient-Net in the set $\mathcal{F}$ of the protocol. This means that the size of $\mathcal{F}^k$ is incremented by 3 at each protocol iteration. The evolution of the error rate can be observed in Figure 7.

1) Experimental setting details: Efficient-Net B0 (implemented in Pytorch with a first stride set to 1 to avoid the destruction of the stego noise) was initialized with pre-training on ImageNet. Starting from iteration 4, the GPU cannot load all models ($4 * 3 = 12$) to compute the next adversarial stegos with Backpack. Therefore, we only defeat the last models from 3 past iterations (9 models in total). The exit condition of Algorithm 1 is required to hold with precision 0.01.

2) Computational cost: At iteration 3, the optimization of the cost map takes, in average, 16.12 minutes per image, on GPU Nvidia V100 with 16Go of memory.

3) Results: For this experiment, because Efficient-Net might have difficulties converging, we fed the network with newly sampled stegos in each batch from the optimized cost maps obtained with Backpack. We can therefore apply Curriculum Learning easily because we simulate the embedding of a message of any length with the cost map. For networks that might have difficulties converging (such as B0 Efficient-Net), we can start the training with a high payload and decrease it gradually during the training until it reaches the wanted value. But we observe that it is important to fine-tune the learning by training finally on the stegos produced by Backpack.

VI. Attack and Evaluation at QF100

A. Extending Backback to detectors for images at QF 100

As shown in recent work [31], steganography in JPEG images of high-quality factors is very detectable. For quality factors 99 and 100, classifiers based on the rounding errors in the spatial domain after decompressing the JPEG images exhibit extremely high performance.

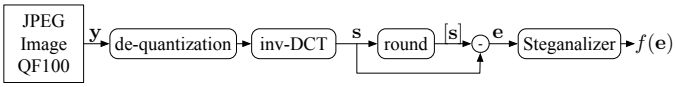

Fig. 8: Pipeline used to build the adversary at QF100.

JPEG decompression of an image composed of quantized coefficients is made of the following steps, each applied independently to each 8×8 block of the image $\mathbf{x}$:

1) de-quantization, i.e. term-by-term multiplication by the quantization matrix $\mathbf{q}$ (of size 8×8), such that $\mathbf{y} = \mathbf{x} \cdot \mathbf{q}$,
2) application of the inverse discrete cosine transform DCT^{-1}, such that $\mathbf{s} = \text{DCT}^{-1}(\mathbf{y})$. It can be seen as 64 convolutions (term-by-term multiplication then summation) applied to the block, *i.e.* $s_{ij} = \mathbf{w}^{ij} \star \mathbf{y}$ with $\mathbf{w}^{ij} = (w_{kl}^{ij})_{kl}$ and
$$w_{kl}^{ij} = \frac{\beta_k \beta_l}{4} \cos \frac{\pi k(2i+1)}{16} \cos \frac{\pi l(2j+1)}{16}, \qquad (20)$$
$\beta_0 = 1/\sqrt{2}, \beta_k = 1$ for $0 < k \leq 7$,
3) adding value 128 to each pixel,
4) clipping to the finite dynamic range $[0; 255]$,
5) rounding to integers. The rounding operation x to its closest integer (in the sense of L_1 norm) is denoted by the square brackets $[x]$.

The idea of the steganalyst proposed in [31] is to use the rounding error in the spatial domain when decompressing a JPEG image. The rounding error is obtained by keeping the spatial image $\mathbf{s}$ with float values obtained right after the inverse DCT (so without the rounding step), and subtracting to the value its rounded value $[\mathbf{s}]$:

$$\mathbf{e} = \mathbf{s} - [\mathbf{s}.] \qquad (21)$$

Figure 2 illustrates the processing pipeline used to extract the signal of interest $\mathbf{e}$, note that it is radically different than classical deep-learning steganalysis schemes using directly pixel values as inputs.

The empirical standard deviation of rounding errors appears to have a high discriminative power to detect stegos from covers (see top left plot in Fig. 9 for an illustration on BOSSBase with J-Uniward [17] for embedding rates of 0.01 and 0.05). For a given class of trainable detector f, it thus proves much beneficial to modify the data pipeline and feed f by a vector containing decompression rounding errors instead of the image itself. In this case, the newly obtained detector is referred to as the "e" version.

In the experiments presented in this section, we will run a protocol with Backpack for images at QF 100 and e-detectors as steganalysts. The first term in (7) is now given by

$$\nabla_{\mathbf{y}} f(\mathbf{y}) = \frac{df}{d\mathbf{e}} \cdot \frac{d\mathbf{e}}{d\mathbf{s}} \cdot \frac{d\mathbf{s}}{d\mathbf{y}}. \qquad (22)$$

Assuming that the derivative of the rounding function $[.]$ is equal to 0, we have $\frac{\partial s_j}{\partial e_i} = [i = j]$, meaning that $\frac{d\mathbf{e}}{d\mathbf{s}}$ is the identity matrix. Because DCT^{-1} is a linear operator, $\frac{d\mathbf{s}}{d\mathbf{y}}$ consists in applying DCT^{-1} as well. In practice, these additional operations can be coded as non-trainable layers

of the network and auto-diff computes the corresponding gradients.

B. Results for images at QF 100

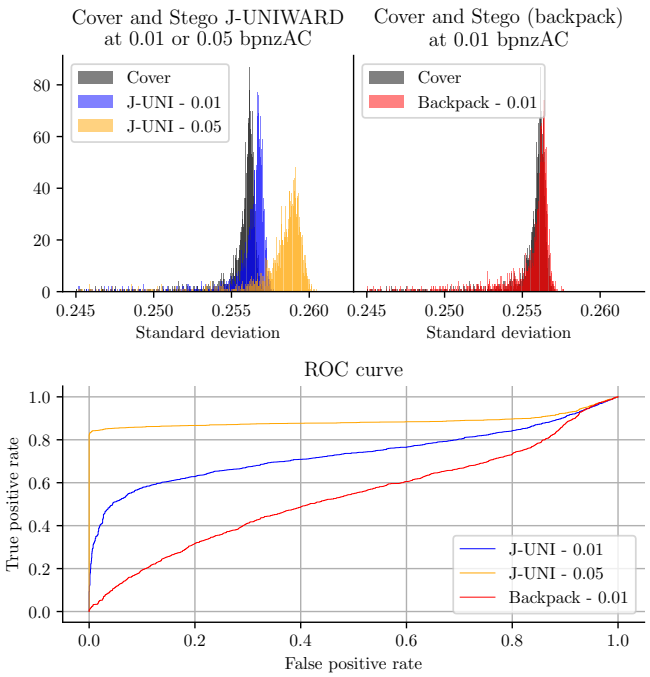

Fig. 9: (Top) Distribution of standard deviation of decompression rounding error of JPEG images at QF 100, for cover (gray), or stego embedded at 0.01 (blue) or 0.05 bpnzAC (orange) with J-Uniward (left), or with stego obtained with $\min\max$ protocol at payload 0.01 bpnzAC (right). (Down) ROC curve based on this statistic, to discriminate between cover and stego at payload 0.01 (blue) or 0.05 bpnzAC (orange) (left) or stego with $\min\max$ protocol (right). Database: 10000 BOSS Base greyscale 512×512 images.

1) Failure of the protocol at QF 100 for too high payload: For an embedding rate of 0.1 bpnzAC, e-XuNet gives an error rate of 0.03% on the test set (only 3 stegos over 5000 are misclassified, and all covers are correctly classified). This detector is almost perfect and is very challenging to avoid. This is why we ran the protocol for images at QF 100 with stegos generated with a very low payload of 0.01 bpnzAC, which gives an error rate of 5.97% for e-XuNet.

2) Protocol at QF 100 at payload 0.01: We show the results of a run of the protocol in Figure 10, for 4 iterations. We observe an increasing error rate, from 5.97% ($k = 0$) to 36.44% ($k = 4$) when Alice targets e-XuNet. Even for non-targeted classifiers e-SrNet, e-EfficientNet and e-GFR, the error rate increases significantly over iterations.

Furthermore, we repeated the statistical analysis as proposed in [31] which sheds light on the fact that the standard deviation of the decompression rounding error can easily discriminate covers from stegos at even a very low payload (even at 0.05 bpnzAC).

We show histograms of those standard deviations to compare stegos produced by J-Uniward and stegos produced at the last iteration of the protocol with Backpack. We can observe in Figure 9 that the empirical distributions of standard deviations of the decompression rounding error between covers and stegos are closer for Backpack (top right) than for J-Uniward (top left) at the same payload 0.01 bpnzAC. The same conclusion arises from the ROC curves, where for Backpack, the curve is close to the diagonal (bottom right), corresponding to near-perfect undetectability.

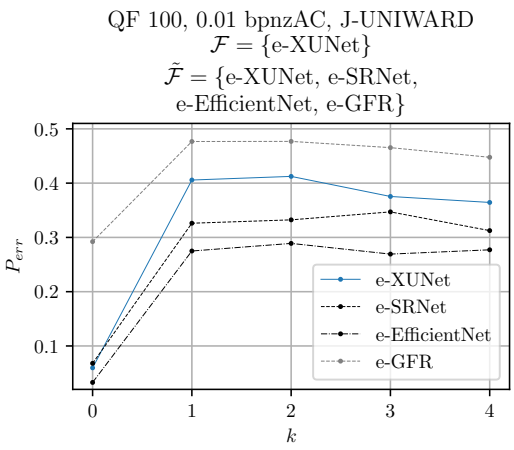

Fig. 10: Run of the protocol with Backpack for images at QF 100, with payload 0.01 bpnzAC, for assumed detector e-XuNet, and real detectors e-SrNet, e-EfficientNet, and e-GFR. Backpack uses SG and costs are initialized with J-Uniward.

VII. Further Analysis and Discussion

A. Analysis of the correlations between DCT coefficients

Among the several steganographic techniques, some state-of-the-art works show that synchronizing the modifications while embedding can increase the security of the scheme, for example by using lattices [32], [33]. We show in this subsection that it can also be achieved via the definition of an *asymmetric* additive distortion function, and it is why we observe correlations for the embedding with Backpack.

The correlation between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_X is defined as:

$$\operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathrm{E}\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y}. \quad (23)$$

We can show that symmetric costs cannot introduce correlations between coefficients because the covariance between two modes is equal to 0.

Let us consider ternary embedding, and define the categorical random variable $\{b_i^{(k)}\}_{(i,k)\in[\![1,256]\!]\times[\![1,N]\!]}$ as a random change made to coefficients belonging to DCT mode ($\lfloor i/8 \rfloor$, $i \mod [8]$) inside a 8×8 DCT block $\lfloor i/64 \rfloor$ inside the $(k) - th$ sample among all the N non-overlapping 16×16 blocks in the whole dataset of images, as illustrated in the left of Figure 12. Every change $b_i^{(k)}$ takes value $\{-1, 0, 1\}$ respectively with probability $\{\pi_i^{-1,(k)}, \pi_i^{0,(k)}, \pi_i^{+1,(k)}\}$. All

those random variables are independent. Their expectations $\mu_i^{(k)}$ are given by:

$$\mu_i^{(k)} = \mathbb{E}[b_i^{(k)}] = \pi_i^{+1,(k)} - \pi_i^{-1,(k)}. \tag{24}$$

In order to evaluate the correlations between DCT modes, let us define now the random variable $B_i : k \in [\![1, N]\!] \mapsto \mathbb{E}[b_i^{(k)}] = \mu_i^{(k)}$. Consequently, correlations between DCT modes $\mathrm{cov}(B_i, B_j)$ are given by:

$$\mathrm{cov}(B_i, B_j) = \mathbb{E}[B_i B_j] - \mathbb{E}[B_i]\mathbb{E}[B_j] \tag{25}$$

$$= \frac{1}{N}\sum_{k=1}^{N} \mu_i^{(k)}\mu_j^{(k)} - \frac{1}{N^2}\left(\sum_k \mu_i^{(k)}\right)\left(\sum_k \mu_j^{(k)}\right) \tag{26}$$

.

J-Uniward [17] gives a *symmetric* additive distortion function, i.e. $\rho_i^{-1} = \rho_i^{+1}$, so the embedding probability are also symmetric ie $\pi_i^{-1,(k)} = \pi_i^{+1,(k)}$. In this case, $\forall i, k, \mu_i^{(k)} = 0$ and so $\mathrm{corr}(B_i, B_j) = \mathrm{cov}(B_i, B_j) = 0$. So symmetric changes cannot induce correlations as opposed to asymmetric probabilities.

ADV-EMB (see section II-A) might favor correlations between DCT modes for two reasons. First, because the message is embedded sequentially in two steps like it is done in synchronization with lattices: (i) first message piece embedded in the common group, then (ii) remaining message piece embedded in the adjustable group, where the costs have been modified according to modifications made in the common group. A second reason is that it uses asymmetric costs. Indeed, ADV-EMB introduces asymmetric costs ($\rho^{-1} \neq \rho^{+1}$) because of its update rule shown in Equation (1), and it is interesting to notice that both Backpack and ADV-EMB both share this feature.

The asymmetry in cost leads to an asymmetry in probabilities, which are plotted in Figure 11. It shows the log-histograms of the differences between probabilities of $+1$ and -1 modifications of coefficients of several covers. A null difference for a coefficient is equivalent to symmetric costs. For both ADV-EMB and Backpack, there is a high quantity of differences close to 0, as a lot of costs are set to a high value in both directions. But the distributions of the differences are significantly different. In the case of ADV-EMB, we can observe that the absolute difference cannot be higher than 0.28. It might be due to the update rule of ADV-EMB, which makes the ratio between ρ^{-1} and ρ^{+1} equal to either 1, $1/\alpha^2$ or α^2. In the case of Backpack, the gradient descent may lead to considerable differences between probabilities, some of them reaching 0.5 in absolute value.

We can also observe that the quantity of asymmetric costs increases at each iteration of the $\min\max$ protocol. For ADV-EMB, this might be due to the increasing number of costs modified at each iteration k. For Backpack, this might be because more and more steps of gradient descent might be required to optimize cost maps. When initialized with a symmetric cost map such as J-Uniward, Backpack is likely to somewhat preserve symmetry if only a few gradient descent steps are used. Conversely, if Backpack performs many gradient steps, it is likely that Backpack will exploit the degrees of freedom provided by asymmetry and will thus significantly drift apart from symmetric costs.

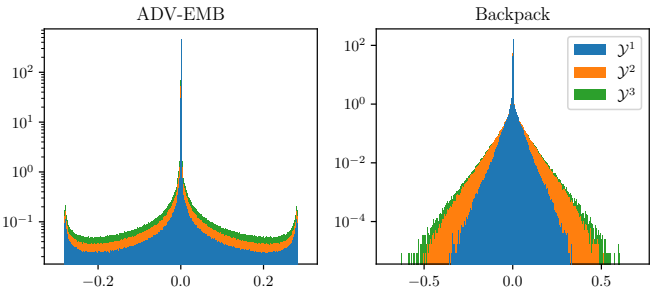

Fig. 11: Log histograms of $\{\pi_i^{-1} - \pi_i^1\}$ for DCT coefficients of 100 cover images, obtained at iteration 1, 2 and 3 for protocol with (left) ADV-EMB or (right) backpack, for images at QF 75 with payload 0.4 bpnzAC.

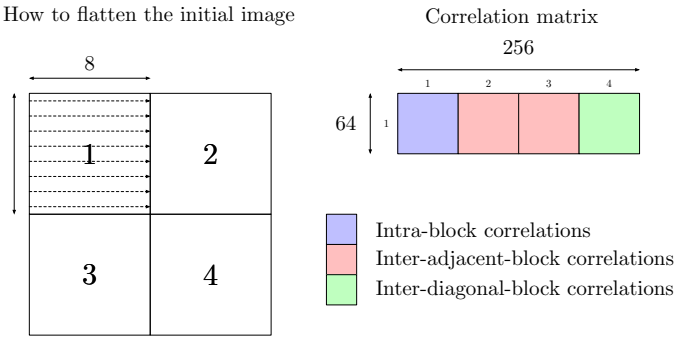

Fig. 12: How the image is flattened to build the correlation matrix (left). If i is the index of the row or the column of a coefficient in the correlation matrix (right), $\lfloor i/64 \rfloor + 1$, $\lfloor i/8 \rfloor + 1$ and $i \mod [8] + 1$ give respectively the index of the block (1,2,3 or 4), the index of the row and the index of the column of the coefficient within its block, in the original image.

By analyzing the covariance matrix of the stego signal of quantized JPEG coefficients (i.e. $\mathbf{b} = \mathbf{y} - \mathbf{x}$ the signal added to the JPEG Cover image to create the Stego image) via formula (26), we highlight the fact that the stego signal exhibits correlations between modes. These weak correlations are within the same block (intra-block correlations) or between adjacent blocks (inter-block correlations). We can compare it to the sensor noise shown in the last plot of Figure 14.

There are no correlations for J-Uniward, but there are some for Backpack. The correlation patterns are similar to the patterns of correlations analyzed on the sensor noise in the DCT domain (see [34]). However, if in [34] these correlations have been shown to favor *continuities* between blocks, the correlations induced by adversarial embedding are on the opposite sign, and we assume that they code *discontinuities* between blocks to remove block artifacts due to the embedding.

B. Payload mismatch

Because we can use a cost map to embed a message of any length, we can wonder if the cost maps optimized using Backpack during a run of $\min\max$ protocol with fixed payload can be re-used to embed messages of arbitrary length. In other words, does Backpack generalize across payloads?

To answer this question, we simulated embedding at payloads 0.1, 0.2, 0.3, and 0.5 in the cover from the optimized cost map obtained in the experiment in section V at payload 0.4 bpnzAC which relies on three neural architectures. Then we trained the three architectures to detect the obtained stegos. We plot on Figure 15 the error rate of every model at each payload, compared to J-Uniward.

It is interesting to notice that J-Uniward has an intuitive behavior, meaning that the smaller is the payload, the less detectable it is for each model. We do not observe the same behavior for the optimized costs with Backpack. Backpack costs give worse results than J-Uniward for embedding rates of 0.1 and 0.2. For embedding rates above 0.3 it gives better results. For Efficient-Net, the costs are tailored to the rate 0.4 as it achieves the highest undetectability for this rate.

C. Cover source mismatch

We can also wonder if the cost maps are transferable to another cover source. The answer is negative at the moment: when trying to embed a message into a cover using the cost map optimized on covers from another source, the stegos are highly detectable. This is because the steganalysts are not transferable as well: they all exhibit high error rates when it comes to discriminating between covers and stegos from another source. We believe that the lack of transferability between cover sources is due to the lack of transferability of the steganalysts or, in other words, the cover-source mismatch must be primarily solved within the steganalysis literature. We can conjecture that, if someday one can provide an efficient steganalyst with transferability across cover sources, the cost maps provided by a protocol run with this steganalyst would be transferable for other sources. This, of course, cannot be asserted or refuted at the time of writing.

D. Where to stop?

In the following paragraph, we discuss a remaining question: how far should we cross the decision boundary when we fool a classifier?

This question arises because we are solving an alternative game of the real game with an infinite set of actions. When Alice plays the $\min\max$ strategy, Eve can answer by creating a new action. Then, Alice hopes that the stego she creates will not be detected by the next detector. Eve's utility function is the output probability of stego class of a detector. When the probability is below 0.5, the stego fools the classifier.

In a previous work introducing the $\min\max$ protocol [8], we proposed to stop whenever the decision boundary is crossed, so whenever the stego probability class is below 0.5. We had the intuition that crossing the boundary too far would make the stego too detectable afterward. In the present paper,

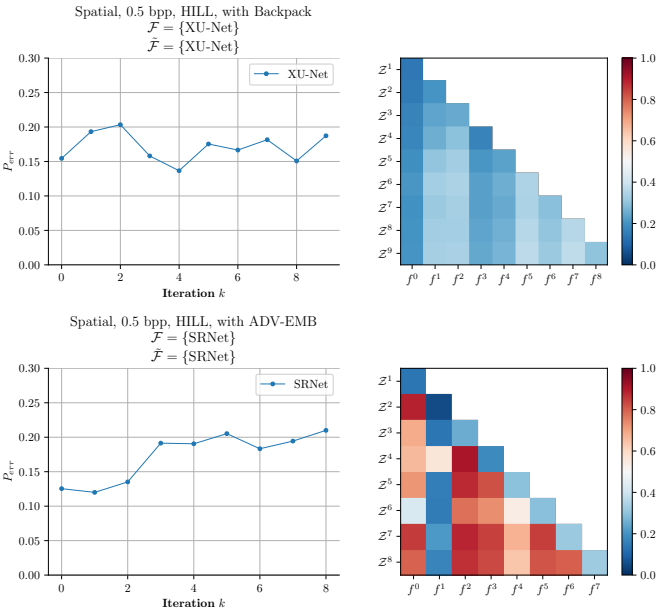

Fig. 13: Figures in the left column show the error of the detector trained to detect steganography on images Alice would send at i^{th} iteration of the protocol (on x-axis). Figures in the right column show the probability that stego images created at row$^{\text{th}}$ iteration will be detectable by column$^{\text{th}}$ detector.

we decided to stop when the detectability of the stego becomes smaller than the detectability of its corresponding cover. It has the advantage of not modifying a stego whose cover is already misclassified. However, all these ideas are not relying on an optimized process, and it would be interesting to find a principled stopping rule.

E. Study of spatial-domain steganography

To show, that backpack is not restricted to JPEG domain, it is used here in combination with the $\min\max$ protocol for steganography in the spatial domain. In this setting, Alice wishes to communicate the payload at 0.5bpp. She anticipates that Eve will use a XuNet detector, and she computes the initial costs in the Backpack attack (and for stego images at iteration zero) using HILL [35] steganography. A similar setting is used [8], where the $\min\max$ protocol is used with ADV-EMB attack.

The upper left figure in Figure 13 shows the probability of error P_{err} of Eve's detector trained to detect images produced by Alice at i^{th} iteration. In the first two iterations, the error increases from 15% to 20%, but then it drops and starts to fluctuate. This contrasts with the behavior of ADV-EMB attack (shown in the same figure) where, after an initial drop, the error continuously increases and after eight iterations reaches 0.21. This means that $\min\max$ protocol with both types of attacks reaches the same error, but four times faster with the proposed Backpack.

While the fluctuation of error when $\min\max$ protocol uses backpack might be disturbing, it is caused by the power of Backpack. Figure 13 (right columns) shows the probability of success of an attack against the set of detectors $\{1, \dots, l\}$,

where l is the step of the $\min\max$ protocol. Red colors mean that the attack is failing, while blue colors mean it is succeeding. Ideally, the lower left triangle should be blue, as is in the case of Backpack. In contrast, the same figure for ADV-EMB is mostly red, which shows that the attack is failing against a set of classifiers, as has been mentioned in the motivation above. The "nice" behavior of $\min\max$ protocol with ADV-EMB observed in Figure 13 (bottom left) is due to the fact that $\min\max$ protocol manages to remedy the flaws of ADV-EMB.

In the case of Backpack, the observed fluctuation is caused by "overshooting", which means that Backpack creates stego images that all trained detectors consider to be cover, but by doing so it introduces changes detectable by another (future) detector. Thus, this fluctuation corresponds to a process of the $\min\max$ protocol improving its model of covers, stored in sets of detectors[10]. Figure 1 in Ref. [9] anticipates such a behavior, but due to the low power of ADV-EMB it had not been observed earlier. According to Theorem 1 in [8], the fluctuation disappears as the number of iterations increases, but eight iterations seem not to be enough, and the computational complexity prevents us from doing more iterations.

Finally, it is interesting that these fluctuations have not been observed during embedding in the DCT domain. We believe this to be caused by plausible changes in DCT domain being more restricted.

VIII. Conclusions and perspectives

The proposed method, called Backpack, relies on the fact that state-of-the-art steganalyzers are (at the time of writing convolution neural network) differentiable w.r.t. the input image coefficients. Backpack uses a smooth approximation of the piece-wise constant function used to draw integer-valued coefficient modifications so that the coefficient modifications can be differentiable w.r.t. their probability distribution parameters. Two candidate approximation functions are investigated: Double-Tanh function and Softmax-Gumbel. Backpack also uses differentiation of an implicit function to back-propagate from the probabilities to the costs, while being compliant with the entropy constraint for message embedding.

The experiments confirm the theoretical correctness of the approach. Several runs of the protocol using Backpack and a $\min\max$ strategy validate the valuable performance of this adversarial strategy in various settings especially compared to ADV-EMB which quickly converges to a maximum error rate. The generality of the method makes it possible to optimize against several steganalysis schemes.

Future works can try to reduce the complexity of the Backpack algorithm by tuning the different hyper-parameters of the embedding scheme (the number of iterations, the learning rate) and the stopping criterion.

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 support by Czech Ministry of Education 19-29680L. This work was granted access to the HPC resources of IDRIS under the allocation 2019-AD011011259 and 2022-AD011012567R1 made by GENCI (Grand Equipement National de Calcul Intensif).

[10]This is the very experience most designers of new steganographic algorithms have.

References

[1] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *Information Forensics and Security, IEEE Transactions on*, vol. 11, no. 2, pp. 221–234, 2016.

[2] Q. Giboulot, R. Cogranne, and P. Bas, "Detectability-based jpeg steganography modeling the processing pipeline: the noise-content trade-off," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2202–2217, 2021.

[3] R. Cogranne, Q. Giboulot, and P. Bas, "Efficient Steganography in JPEG Images by Minimizing Performance of Optimal Detector," *IEEE Transactions on Information Forensics and Security*, pp. 1–16, Sep.

[4] A. D. Ker, T. Pevný, and P. Bas, "Rethinking optimal embedding," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2016, pp. 93–102.

[5] T. Taburet, P. Bas, W. Sawaya, and J. Fridrich, "A Natural Steganography Embedding Scheme Dedicated to Color Sensors in the JPEG Domain," in *Electronic Imaging 2019*, Burlingame, United States, Jan. 2019.

[6] P. Sallee, "Model-based steganography," in *International Workshop on Digital Watermarking (IWDW), LNCS*, vol. 2, 2003.

[7] S. Kouider, M. Chaumont, and W. Puech, "Adaptive steganography by oracle (aso)," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[8] S. Bernard, P. Bas, J. Klein, and T. Pevny, "Explicit optimization of min max steganographic game," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 812–823, 2020.

[9] S. Bernard, T. Pevny, P. Bas, and J. Klein, "Exploiting adversarial embeddings for better steganography," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019, pp. 216–221.

[10] B. Bosansky, C. Kiekintveld, V. Lisy, J. Cermak, and M. Pechoucek, "Double-oracle algorithm for computing an exact nash equilibrium in zero-sum extensive-form games," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 335–342.

[11] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "Cnn-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, 2019.

[12] T. Pevny and A. D. Ker, "Exploring non-additive distortion in steganography," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, 2018, pp. 109–114.

[13] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 2, pp. 215 –224, june 2010.

[14] S. Bernard, P. Bas, T. Pevny, and J. Klein, "Optimizing additive approximations of non-additive distortion functions," in *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021, pp. 105–112.

[15] J. Butora and J. Fridrich, "Reverse jpeg compatibility attack," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1444–1454, 2020.

[16] R. Cogranne, Q. Giboulot, and P. Bas, "The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis "Into The Wild"," in *ACM IH&MMSec (Information Hiding & Multimedia Security)*, ser. ACM IH&MMSec (Information Hiding & Multimedia Security), Paris, France, Jul. 2019.

[17] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[18] L. Guo, J. Ni, W. Su, C. Tang, and Y.-Q. Shi, "Using statistical image model for jpeg steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[19] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 705–720, 2010.

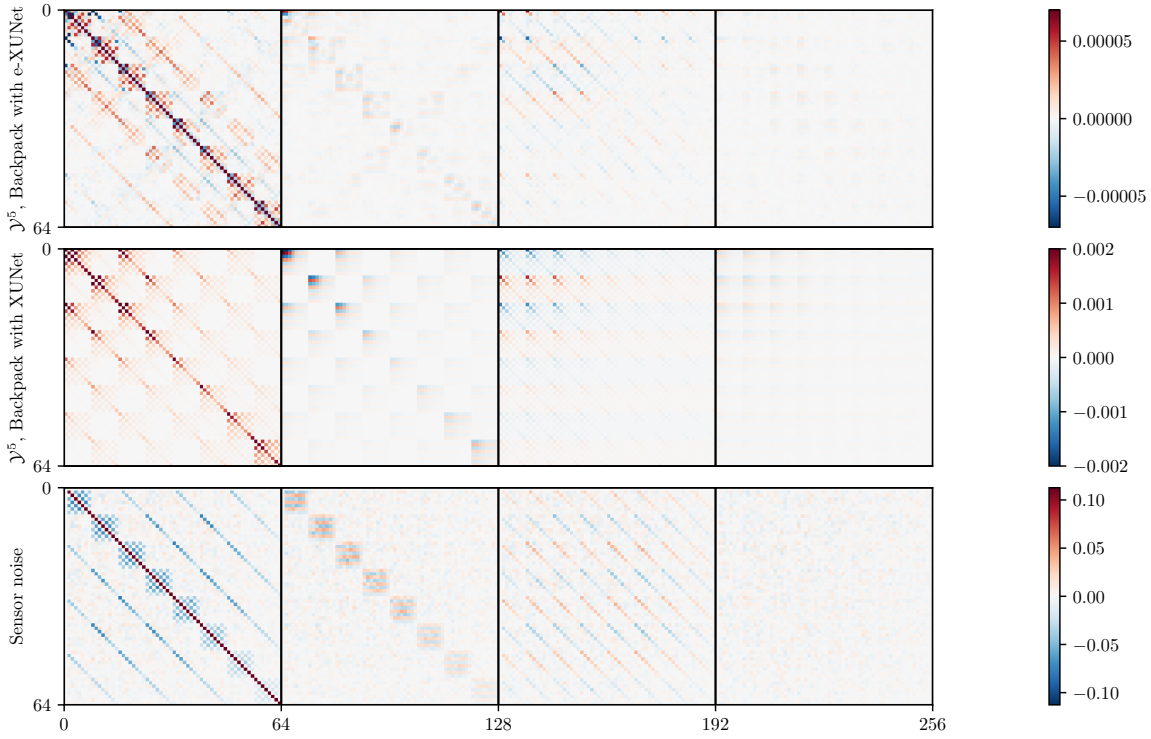

Fig. 14: Covariance matrix computed via formula presented in Equation 26 from the probability maps of the whole dataset of 10000 images, for (top) stegos obtained with Backpack at the 5-th iteration against e-XUNet, or (middle) against XU-Net, for QF 100 payload 1.0 bpnzAC, using 10000 images of BOSSBase decomposed into non-overlapping 16×16 blocks. (2 thresholding operations are applied to reduce the range. Red and blue are for respectively positive and negative correlations.) Last plot shows the correlations of the sensor noise computed from a single RAW image. The way the correlations are plotted is explained in Figure 12.

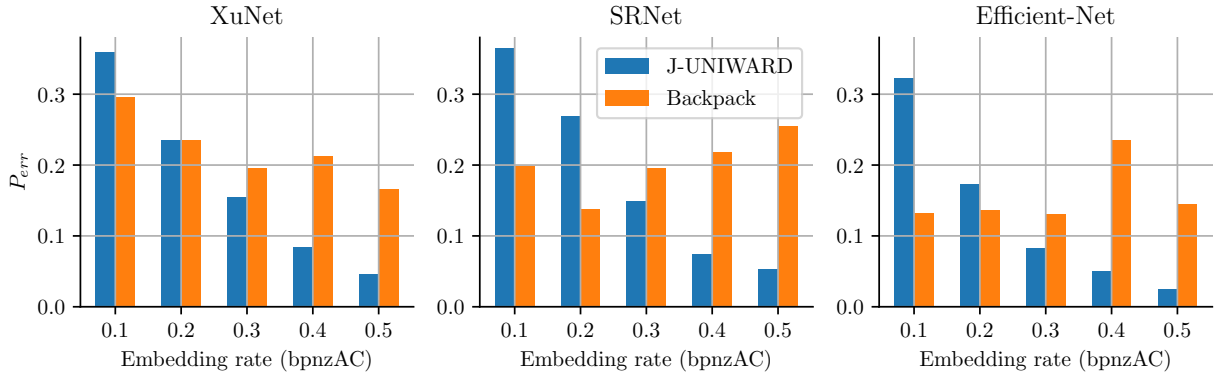

Fig. 15: Error rate of three models (XuNet, SrNet, and Efficient-Net) trained to detect stegos sampled for the cost maps at different embedding rates (from 0.1 to 0.5) where the cost are optimized with Backpack during a $\min\max$ protocol at payload 0.4bpnzAC.

[20] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Shi, "An embedding cost learning framework using gan," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 839–851, 2020.

[21] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[22] P. Bas, T. Filler, and T. Pevny, ""Break Our Steganographic System": The Ins and Outs of Organizing BOSS," in *International Workshop on Information Hiding*, vol. 6958, LNCS. Springer Berlin Heidelberg, 2011, pp. 59–70.

[23] G. Xu, "Deep convolutional neural network to detect j-uniward," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 67–73.

[24] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[25] Y. Yousfi, J. Butora, J. Fridrich, and C. Fuji Tsang, "Improving efficient-net for jpeg steganalysis," in *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021, pp. 149–157.

[26] V. Holub and J. Fridrich, "Low-complexity features for jpeg steganalysis using undecimated dct," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[27] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of

adaptive jpeg steganography using 2d gabor filters," in *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. ACM, 2015, pp. 15–23.
[28] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 3, pp. 868–882, 2012.
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[30] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "Cnn-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074–2087, 2019.
[31] J. Butora and J. Fridrich, "Reverse jpeg compatibility attack," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1444–1454, 2019.
[32] W. Tang, B. Li, W. Luo, and J. Huang, "Clustering steganographic modification directions for color components," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 197–201, 2015.
[33] T. Denemark and J. Fridrich, "Improving steganographic security by synchronizing the selection channel," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2015, pp. 5–14.
[34] T. Taburet, P. Bas, W. Sawaya, and J. Fridrich, "Natural steganography in jpeg domain with a linear development pipeline," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 173–186, 2021.
[35] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 4206–4210.

Solène Bernard received the Engineering diploma from the Ecole Centrale de Lille - France, in 2018, and then the Ph.D degree in automatic, computer engineering and signal and image analysis from Centrale Lille Institute in 2021. She has now a Post-Doctoral position in Pasteur Institute, working on drug discovery.

Patrick Bas received the Electrical Engineering degree from the Institut National Polytechnique de Grenoble, France, in 1997, and then the Ph.D. degree in signal and image processing from Institut National Polytechnique de Grenoble, France, in 2000. He has co-organized the 2nd Edition of the BOWS-2 contest on watermarking in 2007, and the BOSS and Alaska contests on steganalysis respectively in 2010 and 2019.

Tomáš Pevný received the master's degree in computer science from the School of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague, in 2003, and the Ph.D. degree in computer science from the State University of New York, Binghamton, in 2008. From 2008 to 2009, he held a Post-Doctoral position at Gipsa-lab, Grenoble, France. He is with Artificial Intelligence Center at Czech Technical University in Prague. His research interests are applications of non-parametric statistics (machine learning, density modeling) with a focus on computer security, steganography, and steganalysis.

John Klein obtained the habilitation à diriger les recherches in computer sciences from the University of Lille in 2017 and a Ph.D. in information sciences from the University of Rouen in 2008. Prior to that, he was an intern in Beijing University and obtained a Master degree in signal processing from the University of Bordeaux and an engineering degree from ENSEIRB in telecommunications. His research interests include several aspects of artificial intelligence on both symbolic (approximate reasoning and uncertainty models) and data driven (ensembling and deep learning) sides. His works are also frequently applied to image processing tasks such as image segmentation, object tracking, biomedical image analysis and multimedia security.