Efficient Steganography in JPEG Images by Minimizing Performance of Optimal Detector

Rémi Cogranne, *Member, IEEE*, Eva Giboulot, and Patrick Bas, *Senior Member, IEEE*

***Abstract*—Since the introduction of adaptive steganography, most of the recent research works seek at designing cost functions that are evaluated against steganalysis methods. While those approaches have been successful, they rely on intuitive principles and ad-hoc costs associated with each pixel or Discrete Cosine Transform (DCT) coefficient. Beyond the empirical assessments, the insights one can get from such approaches are very limited. On the opposite, this paper presents an original method for steganography in JPEG images that exploits a statistical model of the DCT coefficients. Within the framework of hypothesis testing theory, we use a statistical model of covers to derive the analytical expression of the most powerful detector. The objective of the steganographer is to minimize the statistical performance of this "omniscient detector" which represents a "worst-case" scenario for security. This paper shows how this method allows designing effective steganography, in terms of both security and computational complexity, in the two main use cases: when having only one single JPEG image and when the uncompressed image is available, case also known as Side-Informed (SI). A wide range of numerical comparisons shows that the proposed method outperforms the current state-of-the-art especially against the latest and most accurate steganalysis approaches based on Deep Learning.**

***Index Terms*—Steganography, steganalysis, JPEG images, hypothesis testing, statistical modeling.**

I. Introduction

STEGANOGRAPHY and steganalysis form a cat-and-mouse game: modern steganography aims at hiding information within an innocuous cover object using a secret key. The resulting stego-object should allow the intended receiver, with the shared secret key, to extract hidden secret data. Furthermore, the stego-object should remain indistinguishable from the cover object so that it can be sent over an unsecured channel while remaining undetected. On the other hand, an eavesdropper may wiretap communication and try to reveal if data are hidden into inspected objects using an arsenal of detection techniques referred to as steganalysis. From this setting, a cat-and-mouse game emerges naturally: steganography seeks to design more and more advanced data hiding systems to pass through the steganalyst undetected while, on the opposite side, the goal of steganalysis is to develop sharper and sharper analysis techniques to detect weak signals left by the steganographer with the highest accuracy possible [1].

While modern steganography may use a wide range of covers, it has been mostly developed for digital media and especially in images. Indeed, digital images are massively shared over the Internet making its use as a cover for steganography unsuspicious. Images are also very often compressed using the JPEG format which is simple enough to be easily manipulated for hiding data while also offering enough room for payload sizes of practical interest.

Steganography has also been largely improved thanks to the use of linear error-correcting codes from information theory. The first uses of Huffman codes [2] allowed reducing the number of changes, hence lowering detectability. A tremendous improvement was brought with Syndrome-Trellis Codes [3] that improved coding efficiency, closing up the gap with Shannon theoretical bounds. The STC also allows taking into account the cost associated with each element (either pixels or DCT coefficients from JPEG images). This pioneering work opened the doors for the so-called adaptive steganography that, using this cost function, embeds more data in locations where detection is expected to be harder.

Over the past two decades, steganalysis has largely benefited from machine learning. Indeed, while first techniques for hidden data detection were tailored to catch specific traces of steganography [4], machine learning quickly helped to design universal steganalysis method that is effective against a wide range of steganographic algorithms. Larger and larger handcrafted features sets [5], [6] associated with classifiers specifically designed to handle such high-dimensional features [7]–[9] have allowed to detect more and more subtle traces of data hiding. Recently, deep learning techniques have been used as an alternative that successfully managed to jointly optimize the features extraction and the classification tasks [10]–[12].

A. State-of-the-Art

A vast majority of recent works in steganography has attempted to design cost functions. For this purpose, the most

Manuscript received January 19, 2021; revised April 20, 2021, June 24, 2021, and August 9, 2021; accepted August 12, 2021. Date of publication September 10, 2021; date of current version April 5, 2022. This work was supported in part by the French National Research Agency through ALASKA Project: https://alaska.utt.fr under Grant ANR-18-ASTR-0009 and in part by the French ANR DEFALS Program under Grant ANR-16-DEFA-0003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jiwu Huang. *(Corresponding author: Rémi Cogranne.)*

Rémi Cogranne and Eva Giboulot are with the LIST3N Laboratory, Troyes University of Technology, 10004 Troyes, France (e-mail: remi.cogranne@utt.fr; eva.giboulot@inria.fr).

Patrick Bas is with the CRIStAL, Centrale Lille, UMR 9189, CNRS, Université de Lille, 59000 Lille, France (e-mail: patrick.bas@centralelille.fr).

The source codes implementing the steganographic methods are available on CodeOcean under DOI: 10.24433/CO.2423893.v3 (for JMiPOD) and 10.24433/CO.7938898.v2 (for SIMiPOD). The codes for all other steganographic methods, feature extractors, and classifiers used in this article is available from http://dde.binghamton.edu/download/.

Digital Object Identifier 10.1109/TIFS.2021.3111713

popular approach, by far, has been based on intuition with a practical goal to defeat current steganalysis techniques providing a *post-hoc* validation of the design. In the spatial domain, the main representatives of this approach includes S-UNIWARD [13] as well as HILL [14]. For JPEG images, J-UNIWARD [13], EBS [15] and UERD [16] have been designed following this approach.

The aforementioned algorithms are successful examples of this empirical approach and they have been widely adopted among the community as the current state-of-the-art. However, such design is very limited from a methodological point of view because it does not provide accurate understanding, explanation or insights which leaves researchers unable to understand why a given cost functions works. The problem is even more complex when considering that the results are not consistent over different datasets [17], [18].

Recently, two alternative approaches have been developed using an adversarial context. Those methods use feedback from steganalysis in order to understand how to perform steganography with the goal to defeat an "adversary". The first method along this approach is based on *deep learning* and often on Generative Adversarial Networks (GANs) [19]. In this scenario, the embedder knows the detection method used by the steganalyst and vice-versa such that each can iteratively improve its scheme: the steganalayst, as usual, trains a classifier for detecting hidden data while the steganographer adjust its embedding to evade detection. This approach has been shown to be efficient [20], [21]. In addition, jointly training the adversarial CNN should prevent designing an embedding scheme that is tailored for a particular detector, see results in [21]. However, it is very computationally demanding and it requires a whole dataset to learn how to perform the embedding operation which may not be convenient in some practical contexts.

The very last type of methods in steganography cast the problem of security within a "worst-case attack" scenario. It essentially assumes that one can face an omniscient steganalyst who knows everything required to apply an optimal statistical test. The very definition of such an optimal test is based upon hypothesis testing theory and requires a statistical models of cover and stego media. In this context, the steganographer analyzes this "optimal test adversary" in order to establish a closed-form expression of its statistical performance. This analysis allows the design of a steganographic method that aims at minimizing this statistical performance.

The first embedding scheme built from this approach [22] was not very efficient due to the technique used for image statistical parameters estimations. It was quickly followed by MiPOD [23], [24] that improves the estimation of pixels variances allowing achieving state-of-the-art performance which explains its broad adoption. While those prior works has shown the efficiency of the method, it was also studied as a promising approach toward the theoretical possibility to provide guarantees on security of data hiding under a given statistical model using the optimal test as an upper bound [24]–[27]. As shown in [28] this methodology aims at solving a fundamentally different problem. However the use of this approach into the most practical case of JPEG images is yet to be developed [29].

B. Contribution of Present Paper

The present paper proposes a steganographic method belonging to the last approach: it aims at filling the lack of efficient data hiding method based on minimizing performance of optimal detector for JPEG images. To this end, the present paper proposes several contributions. First of all, it models the impact of the Discrete Cosine Transform which leads to an original heteroscedastic statistical model of DCT coefficients for JPEG images. Based on this statistical model, we design a novel method for data hiding into JPEG compressed images. Both the statistical model and embedding method are sufficiently general and accurate to allow extension to the side-informed case, *i.e.* when uncompressed version of the image is available at the embedding.

The present paper is also provided with a reference source code for *Matlab* and *Python* which allows reproducibility and demonstrates that the proposed method can be implemented with very low computational complexity.

In addition, we present a wide range of numerical results, using different datasets, steganographic schemes and steganalysis methods among which the most recent and efficient ones based on Deep Learning. Those numerous comparisons show that the proposed method achieves better overall performance than current state of the art. From these results, we are also able to infer several facts of interest to the steganalyst. In particular, we confirm that several JPEG-image steganography schemes have been designed to resist feature-based steganalysis methods; therefore those shall not be used alone anymore to assess the practical steganography security thoroughly.

C. Novelty and Relation With Prior Works

This paper is closely related with several prior works, including some of our works. To show better the originality of the present work, we contrast below the contribution with those prior works.

First of all, let us acknowledge that the method developed in the present paper, considering the "worst-case" most powerful detector to minimize its statistical performance finds its roots in [22] and most clear in [23], [24]. Indeed, the former formalize the problem of steganography as the minimization of its Fisher Information while the later explicitly explained the approach focusing on the "omniscient detector". As explained and detailed in Section V the method used in the present paper for estimating pixels variance is based on the one developed in [23], [24] with light yet important modifications for achieving good performance.

Indeed, those modifications are required because it is important to note that the extension to JPEG images is far from being straightforward as already acknowledged in [29], [31]. While [29] constitutes the first extension of this method to JPEG images: it works rather poorly with respect to prior art and it is computationally very demanding. More fundamentally, the method presented in [29] fundamentally differs because it focuses on modeling and taking into account the quantization using an uncompressed pre-cover. In addition, it does not rely on the same model for the steganalyst: this prior work assumes that the warden knows the change probabilities in each direction. On the opposite, we explain why this is equivalent to knowledge of the side information and, hence, we propose a different approach. In the specific case of Side-informed steganography, using the uncompressed pre-cover, we slightly modified the statement of the problem assuming that the “almost omniscient” detector cannot distinguish the change direction (this constitutes the side-information unknown to the detector) but knows the quantization, such smaller modifications are less detectable.

The present paper also goes along several of our recent prior works: we developed a similar method based on minimizing the performance of the most powerful detector for data hiding in JPEG images based on the complete and exact knowledge of the acquisition and processing pipeline in [26], [30]. However, those works focus on developing the most accurate possible statistical model of images. This especially leads us to study the covariance of pixels [26] and DCT coefficients [25] and to take this aspect carefully into account when hiding data into JPEG images. Those works essentially show that such an accurate model is beneficial for steganography. However, because it is almost impossible to estimate correlations between each and every pixels from a single image, those works are can hardly be applied in practice.

On the opposite the present paper focuses on the broad context in which the steganographer has no side-information on the source : it is only given a digital image already compressed (typically out-of-camera picture). It is a follow-up of the conference paper [31] from which we have (1) improved the parameter model estimation for reaching higher performance in terms of security, (2) extended the numerical evaluation in order to assess the performance with respect to current-art steganalysis method based on *deep learning*, (3) extended the proposed method for Side-Informed steganography when unquantized values of DCT coefficients are available and (4) detailed the implementation (and provide source codes) and acknowledge the limitations of the statistical model we exploit.

D. Organization of the Paper

The present paper is organized as follows. Section II provides primers on JPEG compression and presents the proposed statistical model of DCT coefficients from JPEG images. Then Section III recalls the general methodology of the MiPOD family embedding schemes that minimizes the statistical performance of most powerful test for hidden data detection in the ideal case when all parameters are known to the “omniscient detector”. The natural extension of the method to the problem of Side-Informed steganography, with knowledge of unquantized DCT coefficients is presented in Section IV. Section V details the practical implementation and especially the estimation method from a given image. Section VI assesses the practical efficiency of the proposed method throughout comprehensive comparisons with prior art and using recent steganalysis methods based on deep learning. Eventually, Section VII discusses possible future works and conclude the present paper.

II. Statistical Models of JPEG Images

As briefly explained in Section I, the present paper proposes a novel method for data hiding by minimizing the performance of optimal detector in JPEG images. To do so, we will use hypothesis testing theory which requires a statistical model of both cover and stego images.

Usual model of DCT coefficients gather all coefficients which are represented altogether with the same unique statistical distribution model. The Laplacian distribution [32] certainly remains the most popular due to fair accuracy despite highest simplicity and because it was justified in [33]. Other statistical models have been proposed such as the Generalized Gaussian model [34]. Some attempts have been made to carry out steganalysis using Cauchy models [35] and a more accurate and advanced models in [36].

The fundamental limitation of these models for steganography is that *they only provide a model of the modes and not of individuals DCT coefficients*. Consequently, the steganographer cannot use such models without also assuming all DCT coefficients from a given mode are independent and identically distributed. This not only goes against the idea of adaptive steganography but has also been disproved by many recent models of DCT coefficient for steganography which model DCT coefficients individually [25], [30]. However taking into account correlations between DCT coefficients leads to computational difficulties both in the embedding – within the methodology presented in this paper, minimizing the statistical detectability while taking into account samples correlations leads to a non-additive scheme [37] – and in the estimation of the covariance matrix for practical implementation.

These more recent models show that, blocks of DCT coefficients can be modeled as multivariate Gaussians. A natural simplification to the multivariate model is to use a heteroscedastic model of the DCT coefficients – i.e to model DCT coefficients as independent *but not identically distributed*. This is justified by the fact that correlations between DCT coefficients have usually been found to be small [38]. Furthermore this simplification solves both the problem of non-additivity and of the difficulty of estimation – see Section V. Finally, we note that this model has been successfully used in MiPOD [24] and in various denoising algorithms [39]–[41] pointing to the fact that this simplification strikes a good balance between accuracy and simplicity.

A. Cover Image Model

The statistical model of DCT coefficient lies at the heart of the steganographic method proposed in the present paper. In addition, the practical implementation requires understanding of the foundation of such model. Therefore we will briefly recall that, in spatial domains, the pixels $x_{k,l}$ are generally modeled as independent Gaussian random variable with parameters:

$$x_{k,l} \sim \mathcal{N}(\theta_{k,l}, \varsigma_{k,l}^2) \tag{1}$$

where $\mu_{k,l}$ and $\sigma_{k,l}^2$ represent the expectation and the variance of the pixel $x_{k,l}$ with coordinates (k, l).

It is important to note that in this widely used *heteroscedastic* model, the variance is not the same over all pixels, mainly because of the shot noise generated by photo-counting [42], [43].

Let us recall that the JPEG compression is essentially based on Discrete Cosine Transform (DCT) applied on small blocks of 8 × 8 pixels. In order to simplify the notations, we will represent such a block of 8 × 8 pixels as a vector **x** of the 64 components arranged lexicographically. The corresponding DCT coefficient from JPEG image will be denoted as a vector **c**. As explained in Appendix C the DCT can be represented as a single linear operation:

$$\mathbf{c} = \mathbf{D}\mathbf{x} \tag{2}$$

where the matrix **D** of size 64 × 64 represents the linear transform of the DCT applied block-wise (see detail in Appendix C).

As already explained, within the method adopted in the present paper that uses the statistical performance of optimal detector, taking into account the correction between DCT coefficients makes the embedding much more complex as it would become non-additive [37]. However, it should be noted that our approach is restrictive on this aspect because the "optimal detector" is assumed to know the expectation and variance of DCT coefficients. We acknowledge that it has been shown, for instance in [44]–[46], that some existing techniques of changes synchronization help improving the empirical security significantly. This is mainly because those techniques exploit in an empirical manner the inherent correlation of DCT coefficients. On the opposite, in the present paper, we kept the model simple and assumed that DCT coefficients are independent and follow non-identical Gaussian distributions:

$$c_{m,n} \sim \mathcal{N}(\theta_{m,n}, \sigma_{m,n}^2) \tag{3}$$

where $\theta_{m,n}$ and $\sigma_{m,n}^2$ are respectively the mean value and variance of the m,n-th DCT coefficients (2).

These coefficients are quantized with a different quantization factor $\Delta_{m,n}$, depending on their position in each 8 × 8 block:

$$\bar{c}_{m,n} = \text{round}\left(c_{m,n}/\Delta_{m,n}\right). \tag{4}$$

The second assumption we make is that the quantization is negligible compared to the noise variance. We acknowledge that this assumption is not always accurate in practice. However, establishing the closed-form expression of the optimal test for steganalysis taking into account exactly the effect of quantization is extremely difficult, see for instance [47]. In addition, adaptive embedding naturally favors coefficients with the highest variance for which this assumption is acceptable.

This assumption allows us to simplify the probability mass function (pmf) of quantized DCT coefficients as follows – see the Appendix B for details:

$$p_0(k) = \mathbb{P}\left[\bar{c}_{m,n} = k\right] \doteq \frac{\Delta_{m,n}}{\sigma_{m,n}} \phi\left(\frac{k\Delta_{m,n} - \theta_{m,n}}{\sigma_{m,n}}\right), \tag{5}$$

where $\phi(\cdot)$ represents the standard Gaussian probability density function (pdf).

B. Stego Image Model

In order to derive the optimal test, in Section III that follows, one needs to model the statistical distribution of DCT coefficients after data hiding. In the present paper we propose to use a ternary embedding also referred to as ±1. In brief, whenever a bit to be hidden does not match the least significant bit (LSB) of the selected coefficient, this latter can be changed by ±1. In practice, the message is not directly hidden bit by bit; instead a coding method is used, usually based on linear error correcting codes such as STC [3]. The close to optimal properties of STC can be used to simulate embedding [48] when assigning each DCT coefficient $c_{m,n}$ a different probability $\beta_{m,n}$ of modification. Note that in the present paper we will denote $\beta_{m,n}$ the change-rate.

In terms of information theory the maximal payload R that can be embedded in this framework is given by:

$$R(\boldsymbol{\beta}) = \sum_{n,m} H_3(\beta_{m,n}), \tag{6}$$

where the ternary entropy $H_3(x)$ is given by:

$$H_3(x) = -2x \log_2 x - (1-2x)\log_2(1-2x). \tag{7}$$

It follows from the description of the embedding process above that the distribution of stego-coefficients $s_{m,n}$ is given by:

$$\begin{aligned} &\mathbb{P}\left[\bar{s}_{m,n} = \bar{c}_{m,n}\right] = \left(1 - 2\beta_{m,n}\right), \\ &\mathbb{P}\left[\bar{s}_{m,n} = \bar{c}_{m,n}+1\right] = \mathbb{P}\left[\bar{s}_{m,n} = \bar{c}_{m,n}-1\right] = \beta_{m,n}. \end{aligned} \tag{8}$$

which yields the following pmf of stego quentized DCT coefficients:

$$p_{\beta_{m,n}}(k) = (1-2\beta_{m,n})p_0(k) + \beta_{m,n}\left(p_0(k+1) + p_0(k-1)\right). \tag{9}$$

III. Steganography by Miminizing Performance of the Most Powerful Test

In the framework of the present paper, the embedder seeks to defend against the most powerful test. It is difficult in practice to model the information available to the detector. In a conservative approach, we propose to use the most stringent potential scenario that targets the *omniscient* detector, as defined in [24], which perfectly knows all distribution parameters of all DCT coefficients under both hypotheses; namely the expectations $\theta_{m,n}$, the variances $\sigma_{m,n}$ and the probabilities $\beta_{m,n}$. For the sake of clarity, let us denote $\bar{c}_{m,n}$ the quantized DCT coefficients from cover images, $\bar{s}_{m,n}$ the corresponding samples after data hiding and $\bar{z}_{m,n}$ is used when the nature (cover or stego) of inspected coefficients is unknown.

In this case, the problem of hidden data detection in JPEG images is reduced to a test between simple binary hypotheses:

$$\begin{cases} \mathcal{H}_0 : \bar{z}_{m,n} \sim \mathcal{P}(\theta_{m,n}, \sigma_{m,n}\,;\, 0), \\ \mathcal{H}_1 : \bar{z}_{m,n} \sim \mathcal{Q}(\theta_{m,n}, \sigma_{m,n}\,;\, \beta_{m,n}), \end{cases} \tag{10}$$

where $\mathcal{P}$ is the statistical distribution of DCT coefficients whose pmf is defined in (9) and parametrized by $\theta_{m,n}$, $\sigma_{m,n}$, $\Delta_{m,n}$ and $\beta_{m,n}$: obviously the distribution of cover with $\beta_{m,n} = 0$ yields to the pmf defined in (5).

In this case, the Neyman–Pearson Lemma [49, Theorem 3.2.1] states that for testing simple binary hypotheses the most powerful (MP) test is the Likelihood Ratio Test (LRT) defined by:

$$\log\Lambda(\mathbf{Z}) = \sum_{m,n} \log\Lambda\left(\bar{z}_{m,n}\right) = \sum_{m,n} \log\left(\frac{p_{\beta_{m,n}}(\bar{z}_{m,n})}{p_0(\bar{z}_{m,n})}\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau, \tag{11}$$

where the decision threshold τ is set in order to ensure that the false alarm is upper bounded by α_0:

$$\mathbb{P}\left[\Lambda(\mathbf{Z}) > \tau\right] = \alpha_0. \tag{12}$$

Note also that in (11) we used the log-LR which simplifies as a sum over all coefficients thanks to independence assumption. Note also that even is β's are unknown it has been shown that the above test remains optimal; more precisely is it asymptotically uniformly most powerful see [66].

Using the aforementioned statistical models for the distribution of cover and stego DCT coefficients, it is shown in the appendix A that the log-LR asymptotically converges in distribution to:

$$\log \Lambda^\star(\mathbf{Z}) \rightsquigarrow \begin{cases} \mathcal{N}(0, 1) & \text{under } \mathcal{H}_0 \\ \mathcal{N}(\sqrt{2\varrho}, 1) & \text{under } \mathcal{H}_1 \end{cases} \tag{13}$$

where $\Lambda^\star(\mathbf{Z})$ is the normalized log-LR:

$$\log \Lambda^\star(\mathbf{Z}) = \frac{\sum_{m,n} \log \Lambda(\bar{z}_{m,n}) - \mathbb{E}_{\mathcal{H}_0}\left[\log \Lambda(\bar{z}_{m,n})\right]}{\sqrt{\sum_{m,n} \mathbb{V}ar_{\mathcal{H}_0}\left[\log \Lambda(\bar{z}_{m,n})\right]}}, \tag{14}$$

with $\mathbb{E}_{\mathcal{H}_0}[x]$ and $\mathbb{V}ar_{\mathcal{H}_0}[x]$ the expectation and the variance of random variable x under hypothesis $\mathcal{H}_0$.

Note that the in the previous equations (11)-(12) the decision threshold τ is only related with the level of significance α_0 this is why none of those appear in the statistical analysis of the LR (13). Therefore, regardless of the desired steganalayst significance level α_0, the statistical performance of the most powerful LR test is entirely determined by the so-called "deflection coefficient" [24] defined by:

$$\varrho = \sum_{m,n} {\beta_{m,n}}^2 \left(\frac{\Delta_{m,n}}{\sigma_{m,n}}\right)^4. \tag{15}$$

The method proposed in the present paper consists in embedding in JPEG while Minimizing the Performance of Optimal Detector, hence the name J-MiPOD for the ensuing algorithm. It follows from the results (15) that the goal of the embedder is to minimize ϱ under payload constraint (6):

$$\beta_{m,n} = \arg\min \sum_{m,n} {\beta_{m,n}}^2 \left(\frac{\Delta_{m,n}}{\sigma_{m,n}}\right)^4, \qquad \text{subject to: } R = \sum_{n,m} H_3(\beta_{m,n}), \tag{16}$$

which is a constrained optimization problem whose resolution is explained in more details in Section V. Before detailing the implementation, we first describe how the proposed method can be extended for steganography in JPEG images with Side-Information (SI).

One can note that the proposed method allows finding the change probabilities directly $\beta_{m,n}$ while other traditional approaches in steganography, such as [13]–[16], usually defines ad-hoc costs functions.

As noted in [28] this difference in the very optimization goal has an important consequence: while the proposed method minimizes the sum of the squared embedding probabilities (16), additive distortion, on the opposite, seeks at minimizing the sum a weighted sum of embedding probabilities.

IV. Side-Informed J-MiPOD

Without loss of generality, Side-Informed (SI) steganography essentially consists in leveraging an additional information that is available only at embedding, hence unknown by the detector. One can note that the framework of the methodology proposed in the present paper assumes a "worst-case" in which the "omniscient detector" knows all information of interest for steganalysis. This seems fundamentally again the very concept of "side-informed" steganography. However we do not see a fundamental contradiction between designing a steganographic method assuming that the detector has access to information and a practical evaluation in which the empirical detector does not know this information. Besides, for the problem of side informed steganography we will state the problem assuming that the detector cannot know the change directions.

In the present paper we will focus on the most widely studied aspect of SI-steganography: when the embedder has access to an uncompressed version of the image [3], [13], [15]. This way, the embedder knows the unquantized values of DCT coefficients $c_{m,n}$ while, *in practice, an empirical* detector can only use their quantized values $\bar{c}_{m,n}$ (4).

In order to present clearly why the knowledge of the unquantized value of DCT coefficient is crucial, let us denote quantization error $e_{m,n} \in [-1/2; 1/2[$ as follow:

$$e_{m,n} = c_{m,n}/\Delta_{m,n} - \bar{c}_{m,n}. \tag{17}$$

Indeed, without this information, the embedder only knows $\bar{c}_{m,n}$ and, hence, minimizing the statistical performance of optimal detector leads to change coefficients value by ± 1 with the same probability. On the opposite, the embedder who knows the unquantized value $c_{m,n}$ can choose on purpose to modify this value either by $+\Delta_{m,n}(1/2 - e_{m,n})$ or by $-\Delta_{m,n}(1/2 + e_{m,n})$ to shift the quantized value by $+1$ or -1 respectively.

While it seems intuitively a good choice to favor the smallest modification it is not clear yet how to do so, especially in the context of an adaptive embedding with *ad hoc* costs. In [29] it has been proposed to assume that the "*omniscient detector*" knows the change probabilities in each direction $\beta^+_{m,n}, \beta^-_{m,n}$ as well as the quantization error $e_{m,n}$. While in the present paper, we adopted the "fine quantization" assumption (which supposes that noise variance is much larger than the quantization step), the work presented in [29] lifted the assumption of focuses on the impact of the rounding. While extremely interesting, this approach does not work well in practice.

On the opposite, in the present we assume that in the Side-Informed case the "imperfect knowledge" of the detector must be taken into account. Therefore we stated the very problem of SI steganography differently: we assumed that the detector cannot know the modification direction (hence each changes have the same probability $\beta^+_{m,n} = \beta^-_{m,n} = \beta_{m,n}$) but that the quantization error is known.

In order to maintain the advantage of Side-Information for all coefficients, we propose to force the modification direction as follows:

$$\begin{cases} \beta^+_{m,n} = 0 \;\text{ if }\; e_{m,n} < 0 \Leftrightarrow \operatorname{sign}(e_{m,n}) = -1, \\ \beta^-_{m,n} = 0 \;\text{ if }\; e_{m,n} > 0 \Leftrightarrow \operatorname{sign}(e_{m,n}) = 1 \end{cases} \tag{18}$$

where sign : $\mathbb{R} \rightarrow \{-1; 1\}$ represents the sign indicative function defined by $\text{sign}(x) = -1$ is $x < 0$ and $\text{sign}(x) = -1$ is $x > 0$.

However, this constraint prevent ternary embedding; therefore, when using the constraints (18) the problem of Side-Informed MiPOD becomes:

$$\beta_{m,n} = \arg\min \sum_{m,n} {\beta_{m,n}}^2 \left(\frac{\Delta_{m,n}(2e_{m,n} - \text{sign}(e_{m,n}))}{\sigma_{m,n}}\right)^4,$$

$$\text{subject to: } R = \sum_{n,m} H_2(\beta_{m,n}), \tag{19}$$

where the binary entropy $H_2(x)$ is defined as:

$$H_2(x) = -x\log(x) - (1-x)\log(1-x). \tag{20}$$

One can note that the expression of the detectability (19) corresponds exactly to the one established for the non-SI case (16) multiplied by the factor $(2e_{m,n} - \text{sign}(e_{m,n}))$ that represents the amount of change. This can be related to the well-known property in testing theory that the probability of detecting a change of magnitude $(2e_{m,n} - \text{sign}(e_{m,n}))$ depends on the change-to-noise ratio:

$$\frac{2e_{m,n} - \text{sign}(e_{m,n})}{\sigma_{m,n}},$$

which can be found in the expression (19) of the detectability.

Last but not least, we would like to provide explanation about the proposed SI steganography problem statement. Of course, we have tried to assume that the "omniscient detector" knows the probability of change directions. However, this approach works rather very poorly. We explain these results by the fact that, on a practical point of view, as discussed in the Section V, accurate estimation of pixels or DCT coefficient variance is hardly possible and we include in the estimated variance a contribution that represents modeling error by empirical detectors (or "local content complexity"). This leads to an overestimation of the variance where the change probabilities are the highest. However for those coefficients with high estimated variance, the ratio $e_{m,n}/\sigma_{m,n}$ becomes negligible and both modification directions become equally likely when minimizing the detectability.

V. Practical Aspects of Implementation

In practice, the implementation of the method proposed in Sections III-IV is far from being straightforward. This section discusses the most important aspects of the implementation.

A. Estimating Parameters From a Single Image

The proposed method has been built upon the assumption that the steganalyzer knows the variance and quantization steps of all DCT-coefficients. In practice, the steganographer is given a JPEG image from which those variances must be estimated (quantization steps are needed for decompression and, hence, available from file header).

Let us acknowledge that the exact variance estimation does matter significantly on the ensuing practical performance. We have noticed that the most accurate estimation of expectation and variance does not imply the highest efficiency for steganography. We explain this observation by the fact that a complex content of the image may prevent the steganalyze accuracy. Hence, as noted in [24], [27] the steganographer must find a tradeoff between accurate estimation of variance relying, using the most efficient denoising method, and preserving part of the textured content into the residuals to take it into account in the embedding. Therefore, what we will describe as an estimation of the variance of DCT coefficients also contains a significant amount of image modeling error.

As explained in Section II-A we propose in this paper to estimate the variance of DCT coefficient use pixels values in spatial, this is carried out into four steps: (1) decompression of the image back into the spatial domain (2) estimation of the expectation of pixels (3) local estimation of pixels variances, and (4) leverage the linearity of the DCT to obtain the variance of DCT-coefficients, see Appendix C.

We would like to clearly state that steps (2) and (3) are based on the method developed in [24] and reused in [31]. The method for variance estimation is mostly modified in the three main direction (1) we have adjusted the parameters value for improving empirical performance against current art steganalysis (2) due to quantization of DCT coefficient we dramatically limit the clipping, from below, of the estimated variance and (3) we replaced the moving average over a 7×7 window by a, average over DCT coefficients from the same mode over the neighboring blocks. In [31] only (2) has been implemented.

In order to ensure that the paper is self-contained we describe below the whole method for expectation and variance of pixels. First, the estimation of pixels expectation, denoted $\tilde{\boldsymbol{\mu}}$, is obtained, as in [24], using a simple Wiener Filter over blocks of pixels of size 2×2 which we denote as

$$\tilde{\boldsymbol{\mu}} = F(\mathbf{X}) \tag{21}$$

with $\mathbf{X}$ the image in spatial domain. One can subtract the estimated expectation to obtain the so-called residuals as follows:

$$\mathbf{R} = \mathbf{X} - \tilde{\boldsymbol{\mu}} = \mathbf{X} - F(\mathbf{X}). \tag{22}$$

Because the estimation of pixels expectation is far from being perfect, the residuals contain a non-negligible contribution from content. In order to estimate the variance of pixels, we modeled the non-zero expectation of residuals can be modeled using a with a trigonometric 2D polynomial of degree d.

To this end, we arrange the residuals centered at pixel (k, l) as a column vector $\mathbf{r}_{k,l}$ which is modeled as

$$\mathbf{r}_{k,l} \sim \mathcal{N}\left(\mathbf{H}\boldsymbol{\xi}_{k,l}, \sigma^2_{k,l}\mathbf{I}\right), \tag{23}$$

where matrix $\mathbf{H}$ represents the trigonometric polynomial and $\boldsymbol{\xi}_{k,l}$ the associated coefficients.

Note that the variance does not change abruptly between neighboring pixels; hence, we assumed in (23) that the small block the residuals $\mathbf{r}_{k,l}$ share the same variance $\sigma^2_{k,l}$.

From (23), it is straightforward that the maximum Likelihood estimation of the variance is given by

$$\tilde{\sigma}^2_{k,l} = \frac{\left\|\mathbf{H}^{\perp}\mathbf{r}_{k,l}\right\|_2^2}{q^2 - Nd} = \frac{\left\|\left(\mathbf{H}^{\top}\mathbf{H}\right)^{-1}\mathbf{H}^{\top}\mathbf{r}_{k,l}\right\|_2^2}{q^2 - Nd} \tag{24}$$

where $\mathbf{P}^{\perp}_{\mathbf{H}} = \left(\mathbf{H}^{\top}\mathbf{H}\right)^{-1}\mathbf{H}^{\top}$ geometrically corresponds to the orthogonal complement to the subspace spanned by the matrix $\mathbf{H}$ (the left null space of matrix $\mathbf{H}$). In (24) $Nd = d(d+1)/2$ is the total number of monomials, hence, $q^2 - Nd$ represents the "number of degree of freedom".

It is important to explain briefly two important remarks. First, within the proposed method it is proposed to estimate the variance in the spatial domain $\tilde{\sigma}^2_{k,l}$ and then, using the linearity of DCT (2) and statistical properties of Gaussian distribution, to calculated the estimated variance of the (unquantized) DCT coefficients $\tilde{\sigma}^2_{m,n}$ blockwise as follows:

$$\tilde{\boldsymbol{\Sigma}}^2_{m,n} = \mathbf{D}^{\top}\tilde{\boldsymbol{\Sigma}}^2_{k,l}\mathbf{D},$$

where $\tilde{\boldsymbol{\Sigma}}^2_{m,n}$ and $\tilde{\boldsymbol{\Sigma}}^2_{k,l}$ are the estimated variance of 8×8 blocks of DCT coefficients and pixels respectively and the matrix $\mathbf{D}$ is the matrix that represents the linear transform of the DCT. Second, for the purpose of numerical stability, one must prevent the estimated variance from being close to 0 since $\tilde{\sigma}^{-4}_{m,n}$ is used in equations (16) and (19). To this end, it was proposed in [24] to lower-bound the variance to 0.01. However it has been observed that this lower bound is much too high for DCT coefficients because, due to their quantization, their estimated is very often much smaller. Hence, the threshold on minimal variance value shall be set as small as possible: we set this the lower-bound to 10^{-10}:

$$\tilde{\sigma}^2_{m,n} = \max\left(\tilde{\sigma}^2_{m,n};\, 10^{-10}\right). \tag{25}$$

Note that this thresholding of variances, from below, is applied for estimated variance of DCT coefficient while the above procedure for variance estimation is applied in spatial domain hence the indices (k,l) for spatial domain and (m,n) for JPEG domain.

We have noted in some of our prior works [26], [30] that taking into account pixels correlation can largely improve the practical security against empirical detectors. To this end we have tried to impose a fixed "covariance structure" that we scale with the variance of pixels. While this improved the method, the computation complexity increases significantly, especially because one must take into account pixels from neighboring 8 × 8 blocks.

In practice, we noted that taking into account correlation between pixels have one notable effect: the embedding probabilities are smoothed, low-pass filtered in a conservative manner where a small embedding change propagates along the neighboring coefficients, see Fig. 1. This observation confirms those that have been presented in [14], [24], filtering the "costs" prevents the embedding algorithm from being "overly adaptive" and eventually increases its security. Driven by those observation, we implemented in the present paper a filtering of the Fisher Information $FI_{m,n} = \Delta^4_{m,n}/\sigma^4_{m,n}$ by a simple averaging over the 9 DCT coefficients corresponding to the same "modes" from the neighboring 8 × 8 blocks:

$$\tilde{FI}_{m,n} = \tilde{\sigma}^{-4}_{m+8i,n+8j} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} w_{i,j}\,\tilde{\sigma}^{-4}_{m+8i,n+8j}. \tag{26}$$

with $\sum_{i=-1}^{1}\sum_{j=-1}^{1} w_{i,j} = 1$.

TABLE I

EVOLUTION OF MINIMAL PROBABILITY OF ERROR $\overline{P}_{\mathrm{E}}$ MEASURED WITH EFFICIENTNET-b3 FOR PROPOSED J-MIPOD SCHEME FOR DIFFERENT VALUES OF THE NOISE ESTIMATION PARAMETERS, BLOCK SIZE q AND TRIGONOMETRIC POLYNOMIAL DEGREE d (TOP, $QF = 95$ AND $R = 0.2$, BOTTOM $QF = 75$ AND $R = 0.2$)

	q=3	q=5	q=7	q=9	q=11	q=13	q=15	q=17
d=3	0.412	–	–	–	–	–	–	–
d=5	0.409	0.420	–	–	–	–	–	–
d=7	0.396	0.405	0.398	–	–	–	–	–
d=9	0.392	0.397	0.393	0.387	–	–	–	–
d=11	0.398	0.395	0.390	0.396	0.384	–	–	–
d=13	0.387	0.398	0.392	0.387	0.394	0.391	–	–
d=15	0.383	0.380	0.392	0.396	0.392	0.387	0.383	
d=17	0.380	0.384	0.383	0.398	0.397	0.390	0.387	0.373

	q=3	q=5	q=7	q=9	q=11	q=13	q=15	q=17
d=3	0.199	–	–	–	–	–	–	–
d=5	0.161	0.195	–	–	–	–	–	–
d=7	0.143	0.157	0.166	–	–	–	–	–
d=9	0.134	0.143	0.144	0.163	–	–	–	–
d=11	0.136	0.127	0.135	0.140	0.142	–	–	–
d=13	0.138	0.131	0.137	0.123	0.130	0.132	–	–
d=15	0.111	0.129	0.141	0.131	0.129	0.132	0.127	–
d=17	0.108	0.135	0.121	0.119	0.126	0.129	0.133	0.135

This last step constitutes one of the main difference with prior work [31] in terms of variance estimation.

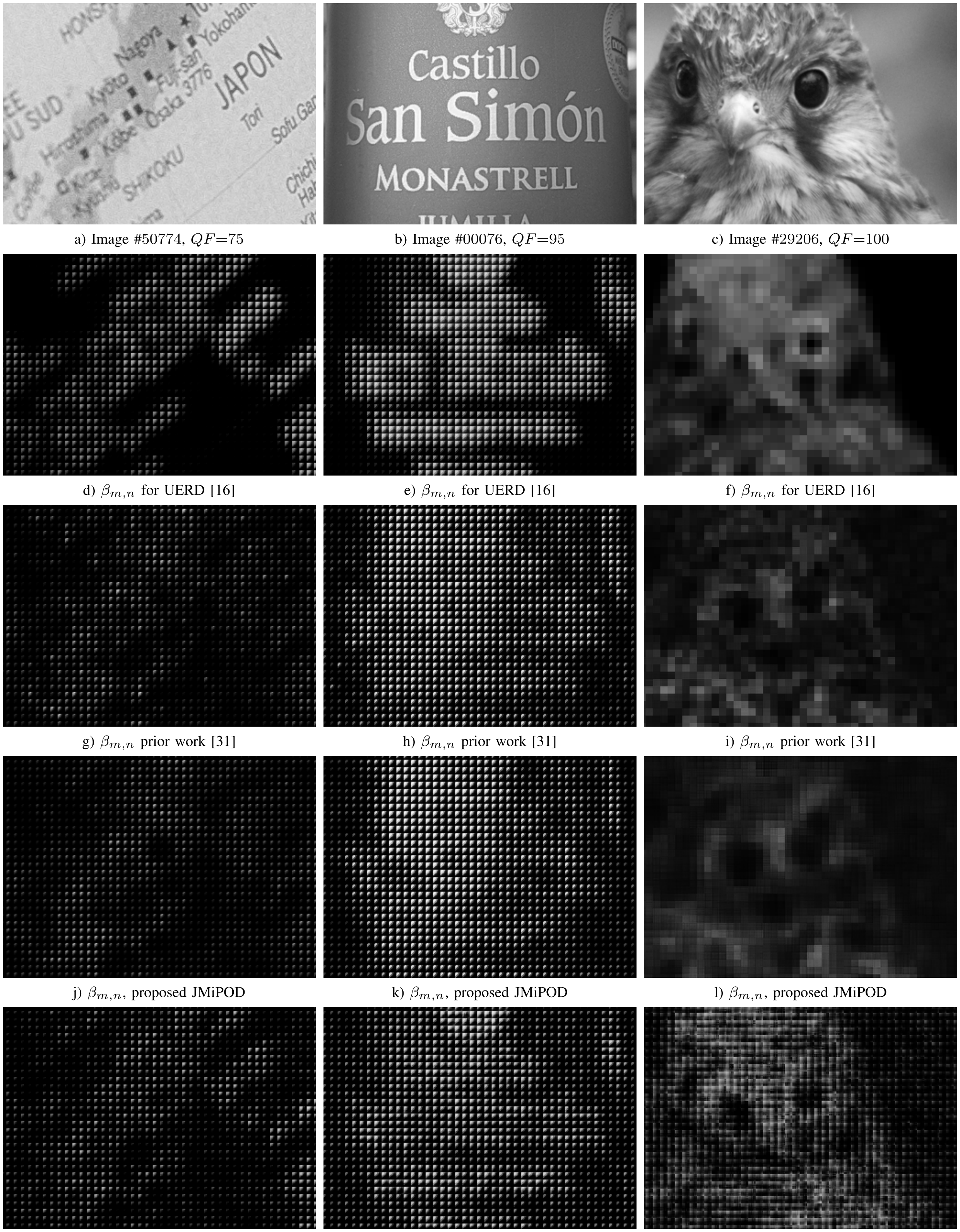
The second main difference lies in the parameters we have for variance estimation; indeed, in the present paper we have used a pragmatic approach of trials and evaluation against empirical detectors in order to instantiate the method. For instance, Table I shows the evolution of P_E, the minimal probability of error under equal prior, when changing the block size q and the 2D polynomial degree d with respect to state-of-the-art empirical detector based on EfficientNet-b3 over BOSS [50] + BOWS [51] databases. As opposed to what has been observed for spatial domain MiPOD [24], it seems that small block size and low degree results lead to a more secured adaptive steganography. Such results allow the setting of the variance estimator with block size $q = 3$, degree $d = 3$.

The same practical approach has been applied to determine the values of the averaging kernel filter (26); few experimentations with trials and evaluations lead us to consider the following weights:

$$\mathbf{w} = \frac{1}{20}\begin{pmatrix} 1 & 3 & 1 \\ 3 & 4 & 3 \\ 1 & 3 & 1 \end{pmatrix}. \tag{27}$$

B. Simulated and Actual Data Hiding

Once the variances of each and every DCT-coefficients have been estimated $\tilde{\sigma}_{m,n}$, the embedding requires solving the



a) Image #50774, QF=75 b) Image #00076, QF=95 c) Image #29206, QF=100

d) $\beta_{m,n}$ for UERD [16] e) $\beta_{m,n}$ for UERD [16] f) $\beta_{m,n}$ for UERD [16]

g) $\beta_{m,n}$ prior work [31] h) $\beta_{m,n}$ prior work [31] i) $\beta_{m,n}$ prior work [31]

j) $\beta_{m,n}$, proposed JMiPOD k) $\beta_{m,n}$, proposed JMiPOD l) $\beta_{m,n}$, proposed JMiPOD

m) $\beta_{m,n}$ for J-UNIWARD [13] n) $\beta_{m,n}$ for J-UNIWARD [13] o) $\beta_{m,n}$ for J-UNIWARD [13]

Fig. 1. Comparison JPEG images steganographic algorithms in terms of payload distribution among DCT coefficients.

constrained optimization problem (16) and (19) for determining the change probabilities $\beta_{m,n}$.

To this end, we used the method, originally proposed in [22], that is based Lagrange multipliers for constraint optimization. From (16), the Lagrange function is given by:

$$\mathcal{L} = \sum_{m,n} {\beta_{m,n}}^2 \left(\frac{\Delta_{m,n}}{\tilde{\sigma}_{m,n}} \right)^4 - \left(\lambda \sum_{m,n} H_3(\beta_{m,n}) - R \right). \quad (28)$$

Differentiating the previous relation (28) and finding the change probabilities $\beta_{m,n}$ for which the differentiate equals 0 leads to the following equation:

$$\beta_{m,n}\log_2(\beta_{m,n}-2) = f(\beta_{m,n}) = \lambda\frac{\tilde{\sigma}^4_{m,n}}{\Delta^4_{m,n}}, \tag{29}$$

$$\Leftrightarrow \beta_{m,n} = f^{-1}\left(\lambda\frac{\tilde{\sigma}^4_{m,n}}{\Delta^4_{m,n}}\right), \quad \forall(m,n) \tag{30}$$

The second part of the previous equation (30) allows straightforwardly computing the change probabilities $\beta_{m,n}$ using the estimated variance $\tilde{\sigma}_{m,n}$. However, the Lagrange multiplier must be determined in order to satisfy the payload constraint:

$$R = \sum_{n=1}^{N} H_3(\beta_{m,n}). \tag{31}$$

To this end, we replace in the payload constraint (31) the change probabilities by their expressions given in (30):

$$R = \sum_{m,n} H_3\left(f^{-1}\left(\lambda\frac{\tilde{\sigma}^4_{m,n}}{\Delta^4_{m,n}}\right)\right), \tag{32}$$

Thank to this last equation (32), we can now find –using binary search or bisection method– the value of the Lagrange multiplier λ that allows ensuring the payload constraint; plugging this value back into (30) allows computing the change probabilities $\beta_{m,n}$.

Note that to speed up the implementation we have tabulated the value of function $f(x) = x\log_2(x-2)$ such that computing the inverse $f^{-1}(\cdot)$ is reduced to a simple lookup table.

Eventually, as stated in Section III- IV, determining the change probability for each and every pixels allows simulating embedding at the information theoretical efficiency bounds. In order to carry out actual embedding one can use the STCs [3]. However the STCs requires a cost $\rho_{m,n}$ from which it determines the change probabilities $\beta_{m,n}$ in order to minimize a distortion function which corresponds to the average cost also referred to as the "distortion":

$$\text{Distortion} = \sum_{m,n} \rho_{m,n}\beta_{m,n}. \tag{33}$$

Note that formulations (33) and (15) may seem contradictory as the objective it is aimed at minimizing are different. Therefore, to allow using the STC, which minimizes the distortion (33), while applying the method proposed is the present paper that minimizes (15), one has to set the cost $\rho_{m,n}$ such that the change probabilities $\beta_{m,n}$ match in both case (up to STC coding loss). To this end, the embedding probabilities $\beta_{m,n}$ and the costs $\rho_{m,n}$ can be bound by the relation [3]:

$$\beta_{m,n} = \frac{e^{-\lambda\rho_{m,n}}}{1+2e^{-\lambda\rho_{m,n}}} \Leftrightarrow \rho_{m,n} = \frac{1}{\lambda}\ln(1/\beta_{m,n}-2), \tag{34}$$

where λ is a Lagrange multiplier used to satisfy the payload constraint. To perform actual embedding one can use the right side of Equation (34) in order to find costs $\rho_{m,n}$ from the embedding probabilities $\beta_{m,n}$ computed from (15) then use those costs with the STCs.

VI. Numerical Results

A. Common Core of All Experiments

In order to assess the efficiency and the security of the proposed method we carried out a large set of numerical experiments. For a meaningful comparison with prior arts, we compare the proposed method with UERD [16], UNIWARD [13] with and without side-information and SI-EBS [15]. For the sake of clarity we also included model-based method proposed in [29] as well as our prior work [31] that the present paper extends. In addition, we have included results obtained with the proposed method without the smoothing of variances of DCT coefficients (26) so that the reader can assess the importance of this step. Eventually we also included results from the spatial domain MiPOD [24] when this algorithm is used "as it" with transformation of variance to DCT-domain (2) in order to show the impact of lower-bounding the variance (25).

We have used two datasets; for the same reason of comparison with prior works, we have used BOSS [50] and BOWS [51] bases together both made of 10, 000 grayscale images of size 512 × 512. As explained in [18], [52] this dataset is very specific since all images have been processed in the same way and especially largely resized. To compare the security of embedding schemes under more realistic conditions we have also used the recent ALASKA base [52], [53]. The version we used is made of 80, 000 grayscale images of size 512 × 512. Each and every images from this dataset have been processed differently using a randomized process. All those datasets have been compressed using the `convert` command from `imagemagick`.

Eventually, we have used two main steganalysis method to assess the security of various embedding scheme. For backward compatibility we used the well-known features based method using DCTR [54] and GFR [55] features, as well as their Selection Channel Aware (SCA) versions [56], with the fast-linear classifier [9]. We have also included results from steganalysis based on deep learning because they constitute now the state-of-the-art. In this domain we have mostly used the recent and already well-established EfficientNet [57] because it has been shown to be extremely efficient for steganalysis in ALASKA Steganalysis Challenge [53], even slightly more than SRNet [11]. We would like to point out that we have also tried several other Deep Learning models such as the Simple JPEG stegnalaysis Net [12] the well-established SRnet [11], MixNet-S [58], NFNet [59] and the latest EfficientNet_v2s [60]; they all generally show the same conclusion especially in terms of embedding algorithm raking. Due to the space limitation those additional results are provided in "complementary material" along this paper.

Before presenting numerical comparisons with the state-of-the-art algorithms on efficiency, we would like to provide a visual comparison showing how different algorithms adaptively embed into JPEG images. To this end, Figure 1 shows an example of three different images from ALASKA along with the probability of change $\beta_{m,n}$ for the four main considered steganographic algorithms. Note that to enhance the visual difference we have used quite large payloads, measured in bpnzAC everywhere in this paper. Note also that the first column corresponds to $QF = 75$, the second column is an example for $QF = 95$ while the rightmost shows the

TABLE II

COMPARISON OF AVERAGE CHANGE RATES (RATIO OF EXPECTED NUMBER OF HIDDEN BITS DIVIDED BY THE TOTAL NUMBER OF AC COEFFICIENTS) OVER ALL IMAGES FROM BOSSBASE [50] FOR VARIOUS EMBEDDING ALGORITHMS AND PAYLOADS

	$R = 0.2$	$R = 0.4$	$R = 0.6$
		$QF = 100$	
Prior work [31]	0.0200	0.0487	0.0843
J-MiPOD (our proposal)	0.0205	0.0502	0.0868
J-UNIWARD	0.0256	0.0569	0.0919
UERD	0.0241	0.0553	0.0912
		$QF = 75$	
Prior work [31]	0.0047	0.0113	0.0192
J-MiPOD (our proposal)	0.0050	0.0121	0.0205
J-UNIWARD	0.0061	0.0135	0.0216
UERD	0.0056	0.0125	0.0201

interesting case of $QF = 100$ for which all quantization factors are 1.

Obviously, UERD adopts an "overly adaptive" strategy where all the payload is concentrated in a rather small number of blocks. On the opposite, our prior work [31] as well as J-UNIWARD both spread the payload across a very large number of blocks. J-UNIWARD, however, seems to use more DCT coefficients corresponding to mid-frequencies. The proposed method is similar to our own prior work, yet, as discussed in the Section V the changes in variance estimation leads to a more conservative approach in which some large areas are almost not used while the payload is spread more evenly in the rest of the image.

Such observations are confirmed in Table II that shows the average change rate overall BOSSbase [50] for a few payloads and quality factors. Our prior work [31] achieves the lowest average change rate because it spreads cost too much which makes may cause embedding into areas where it can be easily detected. J-UNIWARD, on the other hand, is always the most adaptive, which reduces the efficiency of STCs and hence increases the change rate. We can note that the proposed algorithm is slightly less adaptive than our prior work [31] because, as already discussed the changes in the estimation of variance makes it more conservative and prevents the embedding into large areas around coefficients which are considered as the most risky.

B. Comparison With Prior Art in Terms of Detectability

We now move to the most important part of the numerical evaluation, that is the comparison in terms of "detectability" of current art. As already stated, we have carried out a very large range of numerical results using BOSS + BOWS dataset and ALASKA images and using different JPEG quality factors: $\{100, 95, 85, 75\}$. Due to space limitations, we only present a few of them supporting the main conclusion that can be drawn. For all those results, we have used the widely adopted minimal probability of error under equal prior denoted P_E.

TABLE III

SECURITY COMPARISON OF STATE-OF-THE-ART EMBEDDING METHOD VIA $\overline{P}_{\mathrm{E}}$ OVER BOSS + BOWS DATASETS WITH FAST LINEAR CLASSIFIER [9] TRAINED USING SCA-GFR FEATURES [55], [56]

$\overline{P}_{\mathrm{E}}$	R=0.1	R=0.2	R=0.3	R=0.4	R=0.5	R=0.6
			$QF = 100$			
Model-based [29]	0.4286	0.3345	0.2575	0.1979	0.1504	0.1165
MiPOD-DCT [24]	0.4682	0.4259	0.3764	0.3305	0.2876	0.2509
Prior work [31]	0.4697	0.4279	0.3765	0.3315	0.2895	0.2518
w/o smoothing	0.4636	0.4155	0.3617	0.3135	0.2723	0.2350
J-MiPOD	0.4766	0.4449	0.4041	0.3637	0.3237	0.2863
J-UNIWARD	0.4395	0.3762	0.3173	0.2707	0.2303	0.1970
UERD	0.3895	0.3213	0.2714	0.2315	0.2011	0.1744
			$QF = 95$			
Model-based [29]	0.4062	0.2812	0.1698	0.0912	0.0414	0.0166
MiPOD-DCT [24]	0.4619	0.4057	0.3349	0.2684	0.2015	0.1469
Prior work [31]	0.4643	0.4088	0.3431	0.2737	0.2057	0.1499
w/o smoothing	0.4581	0.3969	0.3308	0.2611	0.1959	0.1426
J-MiPOD	0.4644	0.4143	0.3550	0.2856	0.2201	0.1617
J-UNIWARD	0.4677	0.4148	0.3482	0.2832	0.2141	0.1519
UERD	0.4167	0.3393	0.2660	0.2035	0.1520	0.1082
			$QF = 75$			
Model-based [29]	0.2359	0.0973	0.0322	0.0084	0.0023	0.0008
MiPOD-DCT [24]	0.2141	0.1478	0.0945	0.0576	0.0316	0.0166
Prior work [31]	0.3886	0.2659	0.1624	0.0951	0.0523	0.0262
w/o smoothing	0.3669	0.2335	0.1395	0.0771	0.0420	0.0211
J-MiPOD	0.3889	0.2644	0.1649	0.0959	0.0527	0.0271
J-UNIWARD	0.3657	0.2382	0.1415	0.0791	0.0418	0.0207
UERD	0.3091	0.1904	0.1128	0.0664	0.0372	0.0195

First, Table III shows the P_E obtained for all embedding schemes again SCA-GFR features set with fast linear classifier over BOWS + BOSS datasets. From this table it seems that, the prior work [29] is by far the least secure algorithm while UERD also performs generally significantly worse. We also note that J-UNIWARD is more competitive but it is largely subpar by J-MiPOD which seems just a little better than our prior work [31]. We also note that the proposed algorithm performs best for the highest quality factor $QF = 100$. On the opposite J-UNIWARD performs, roughly peaking, as well as the proposed J-MiPOD for $QF = 95$.

This last observation may be explained by the fact that J-UNIWARD, as well as many prior works, were designed using trials and evaluation with setting $QF = 95$ and payload $R = 0.4$ bpnzAC.

One can also note that MiPOD [24] can hardly be used for low QF while, interestingly, the smoothing of the variance does not seem to improve significantly the performance of the proposed method.

Let us now contrast Table IV that shows the very same results except that the steganalysis is carried with EfficientNet-b0. Interestingly, it seems that using Deep Learning detection method, UERD is much more detectable as compared to its competitors (up to almost −20% for QF95 and QF75). To a lessed extend, the comparison is also worse for our prior work [31], especially for QF75. The same observation for the proposed method without smoothing of variances. Eventually, the direct use of MiPOD [24] is extremely unsecure for low QF. Together those two last results highligth how the variance estimation is crutial for ensuring security of the proposed method and the importance of the improvement we have proposed.

Interestingly, J-UNIWARD seems extremely robust: it is the onle comptitor that still performs slightly less well than J-MiPOD with the notable exception of QF100 for which the proposed method is by far outperforming. The difference between those two embedding algorithms is significative for QF75 but smaller for QF95.

Last, Table V offers a very similar comparison but using images from the ALASKAv2 [53] dataset instead. This result

TABLE IV

SECURITY COMPARISON OF STATE-OF-THE-ART EMBEDDING METHOD VIA $\overline{P}_{\mathrm{E}}$ OVER BOWS + BOSS DATASETS WITH EFFICIENTNET-b0 [57]

$\overline{P}_{\mathrm{E}}$	R=0.1	R=0.2	R=0.3	R=0.4	R=0.5	R=0.6
$QF = 100$						
Model-based [29]	0.1989	0.0934	0.0493	0.0275	0.0142	0.0085
MiPOD-DCT [24]	0.3280	0.2176	0.1474	0.1044	0.0754	0.0632
Prior work [31]	0.3253	0.1966	0.1349	0.0919	0.0699	0.0562
w/o smoothing	0.3118	0.1894	0.1301	0.0894	0.0639	0.0527
J-MiPOD	$\underline{0.3635}$	$\underline{0.2458}$	$\underline{0.1786}$	$\underline{0.1334}$	$\underline{0.1039}$	$\underline{0.0819}$
J-UNIWARD	0.2748	0.2038	0.1174	0.0814	0.0617	0.0452
UERD	0.1329	0.0662	0.0424	0.0294	0.0224	0.0147
$QF = 95$						
Model-based [29]	0.3533	0.1836	0.0782	0.0272	0.0057	0.0020
MiPOD-DCT [24]	0.3945	0.3061	0.2211	0.1529	0.0962	0.0597
Prior work [31]	0.4285	0.3408	0.2386	0.1696	0.1044	0.0672
w/o smoothing	0.4340	0.3445	0.2543	0.1739	0.1089	0.0684
J-MiPOD	$\underline{0.4490}$	$\underline{0.3725}$	$\underline{0.2876}$	$\underline{0.1964}$	$\underline{0.1466}$	$\underline{0.0899}$
J-UNIWARD	0.4345	0.3610	0.2758	0.1889	0.1301	0.0774
UERD	0.2311	0.1396	0.0802	0.0487	0.0269	0.0224
$QF = 75$						
Model-based [29]	0.1291	0.0264	0.0050	0.0007	0.0005	0.0002
MiPOD-DCT [24]	0.0527	0.0207	0.0082	0.0040	0.0025	0.0012
Prior work [31]	0.2396	0.1049	0.0507	0.0207	0.0107	0.0055
w/o smoothing	0.2553	0.1246	0.0512	0.0289	0.0109	0.0040
J-MiPOD	$\underline{0.3193}$	$\underline{0.1806}$	$\underline{0.0959}$	$\underline{0.0524}$	$\underline{0.0249}$	$\underline{0.0187}$
J-UNIWARD	0.2918	0.1599	0.0807	0.0342	0.0137	0.0087
UERD	0.1584	0.0684	0.0302	0.0162	0.0087	0.0037

TABLE V

NUMERICAL COMPARISON OF SECURITY OF STATE-OF-THE-ART EMBEDDING METHOD VIA $\overline{P}_{\mathrm{E}}$ OVER ALASKA2 DATASET WITH EFFICIENTNET-b0 [57]

$\overline{P}_{\mathrm{E}}$	R=0.1	R=0.2	R=0.3	R=0.4	R=0.5	R=0.6
$QF = 100$						
Model-based [29]	0.3027	0.1913	0.1545	0.1339	0.1157	0.0911
MiPOD-DCT [24]	0.3855	0.2866	0.2315	0.2008	0.1740	0.1565
Prior work [31]	0.3878	0.2997	0.2369	0.2019	0.1727	0.1564
w/o smoothing	0.3539	0.2471	0.1984	0.1679	0.1485	0.1305
J-MiPOD	$\underline{0.4463}$	$\underline{0.3815}$	$\underline{0.3157}$	$\underline{0.2840}$	$\underline{0.2390}$	$\underline{0.2134}$
J-UNIWARD	0.3643	0.2995	0.2373	0.2109	0.1814	0.1613
UERD	0.2585	0.1851	0.1321	0.1367	0.1261	0.1153
$QF = 95$						
Model-based [29]	0.4117	0.2933	0.2417	0.1605	0.1005	0.0639
MiPOD-DCT [24]	0.4070	0.3488	0.2928	0.2442	0.1973	0.1723
Prior work [31]	0.4450	0.3763	0.3225	0.2614	0.2294	0.1870
w/o smoothing	0.4478	0.3908	0.3259	0.2686	0.2121	0.1802
J-MiPOD	$\underline{0.4601}$	$\underline{0.4126}$	$\underline{0.3562}$	$\underline{0.3002}$	$\underline{0.2563}$	$\underline{0.2277}$
J-UNIWARD	0.4594	0.4048	0.3431	0.2830	0.2306	0.1769
UERD	0.4089	0.3319	0.2687	0.2239	0.1867	0.1193
$QF = 75$						
Model-based [29]	0.2313	0.1107	0.0855	0.0398	0.0195	0.0030
MiPOD-DCT [24]	0.1273	0.0537	0.0400	0.0307	0.0228	0.0183
Prior work [31]	0.2427	0.1344	0.0773	0.0537	0.0401	0.0242
w/o smoothing	0.2765	0.1513	0.0891	0.0567	0.0358	0.0209
J-MiPOD	0.3116	$\underline{0.2068}$	$\underline{0.1404}$	$\underline{0.1043}$	$\underline{0.0676}$	$\underline{0.0424}$
J-UNIWARD	$\underline{0.3253}$	0.2040	0.1206	0.0727	0.0433	0.0268
UERD	0.2445	0.1388	0.0833	0.0511	0.0361	0.0279

is very interesting because those images are more realistic and much more diverse than those from BOSS and BOWS datasets, in particular due to the much more complex, realistic and randomized development processes as well as due to the larger set of cameras (more than 50 different models). This dataset is also made of way more images which allows training complex deep net models with more accuracy. Table V shows similar trends in the following aspects: (1) UERD [16] and model-based [29] remain, by far, much more detectable, (2) the proposed method generally outperforms its competitors, (3) J-UNIWARD is very efficient for QF95 (4) across all those results, the proposed J-MiPOD method seems comparatively more efficient than its competitors for higher payload. In addition the comparison between the proposed J-MiPOD with and without the smoothing of variances and MiPOD [24] transformed to JPEG domain confirms the importance in practice of the improvements we have proposed for variance estimation.

Last not least, we have noted will all deep learning models we have tested [11], [12], [57]–[60] that JPEG images compressed with QF100 become more detectable than those compressed with QF95 for the same payload in bpnzAC; this is in contradiction with what has always been observed with features-based steganalysis, see for instance results reported in [16], [54], [56]. This can be explained in part by the higher actual payload in QF100 due to the much smaller number of non-zeros AC coefficients (while the payload is measured in bpnzAC; bits per non-zero AC coefficients). In addition we would like to point out that this observation was already studied in [64]. This prior work provides a thorough analysis showing that, even for the same number of changes, the Fisher information tends to increase with the QF for the highest factors (typically from QF = 97). While this analysis allows explaining the results obtained with deep learning models, one should note that the results obtained with features-based approach car hardly be explained due to their handcraft designs and partial adaptation with QF.

Finally, let us provide a few more details about the method we have used to train the various deep learning models. First of all, regarding EfficientNet we have removed the very first pooling layer because this simple modification has been shown to be a quite efficient during ALASKA2 steganalysis Challenge [61], [62].

To simplify the training step we have used two important tricks. First, it has been shown in ALASKA Steganalysis Challenge [53] that, even though the classification task is very different from steganalysis, using weights pre-trained from imagenet dramatically speed up convergence. To this end, we have used the `timm` Python package for `pytorch`. Second, we have also adopted a curriculum learning very similar to the one proposed [63] starting with the lowest quality factor, QF75, and with non-adaptive LSBM steganography, reducing the payload step by step. For all the embedding algorithms we used the same process starting from scracted for 25 epochs using the "reduced on plateau" strategy to set the Learning Rate starting from 0.001. Then we used the weights obtained as a starting point for the training with higher quality factor iteratively using the same number of epochs. Once EfficientNet have been trained for each QF, we use iteratively the weights for lower payloads with only 8 epochs, starting with a much lower LR ($1e-4$ instead of $1e-3$).

Last but not least, Table VI compares the computational complexity of the state-of-the-art data hiding scheme for JPEG images. To this end we measured the average execution time over using the same programming language (and in fact same code for simulated embedding) all images from BOSS base with size 512 × 512. Table VI compares show the results obtained for two different payloads ($R = 0.2$ and $R = 0.4$) and

TABLE VI

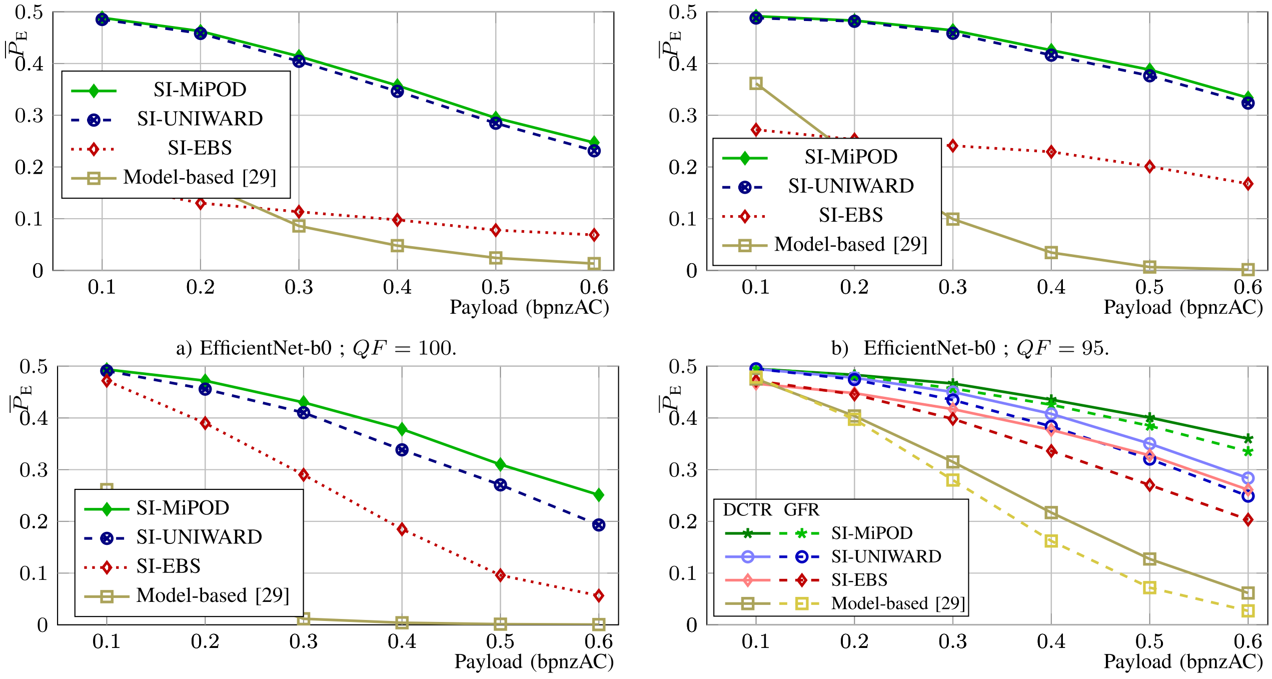
COMPARISON OF CURRENT ART EMBEDDING ALGORITHMS IN TERMS OF COMPUTATION TIME OVER; RESULTS OBTAINED USING MATLAB® NUMERICAL COMPUTING ENVIRONMENT AND REFERENCE IMPLEMENT WHEN AVAILABLE. LEFT, NON-SI SCHEMES; RIGHT, SI SCHEMES

Embedding simulation **without** multithreading nor multi-processing

	Model-based [29]	Prior work [31]	J-MiPOD	UERD	J-UNIWARD	Model-based-SI [29]	SI-MiPOD	EBS	SI-UNIWARD
$QF=100$	6.69 sec.	1.52 sec..	0.35 sec.	0.54 sec.	3.32 sec.	6.78 sec.	0.77 sec.	0.56 sec.	4.54 sec.
$QF=95$	4.07 sec	2.00 sec.	0.34 sec.	0.48 sec.	3.34 sec.	4.49 sec.	0.67 sec.	0.56 sec.	4.56 sec.
$QF=85$	3.76 sec.	2.42 sec.	0.37 sec.	0.48 sec.	3.36 sec.	3.88 sec.	0.65 sec.	0.52 sec.	4.50 sec.
$QF=75$	3.75 sec	2.50 sec.	0.39 sec.	0.49 sec.	3.34 sec.	3.78 sec.	0.69 sec.	0.53 sec.	4.64 sec.

Embedding simulation **with** multithreading and multi-processing

	Model-based [29]	Prior work [31]	J-MiPOD	UERD	J-UNIWARD	Model-based-SI [29]	SI-MiPOD	EBS	SI-UNIWARD
$QF=100$	27.52 sec.	4.87 sec..	1.37 sec.	1.17 sec.	8.49 sec.	24.52 sec.	1.83 sec.	1.31 sec.	11.02 sec.
$QF=95$	13.60 sec	6.60 sec.	1.23 sec.	1.05 sec.	8.38 sec.	13.64 sec.	1.71 sec.	1.27 sec.	10.89 sec.
$QF=85$	12.75 sec.	7.78 sec.	1.27 sec.	1.08 sec.	8.29 sec.	11.41 sec.	1.67 sec.	1.26 sec.	10.88 sec.
$QF=75$	12.71 sec	8.28 sec.	1.35 sec.	1.09 sec.	8.24 sec.	11.02 sec.	1.70 sec.	1.23 sec.	10.90 sec.



a) EfficientNet-b0 ; $QF = 100$.

b) EfficientNet-b0 ; $QF = 95$.

c) EfficientNet-b0 ; $QF = 75$.

d) Features-based detector (GFR+DCTR) ; $QF = 75$.

Fig. 2. Comparison of Side-Informed schemes, SI-MiPOD and current state-of-the-art SI-UNIWARD, through plots showing evolution of P_E as a function of the embedding payload for BOSS [50] and BOWS [51] bases and for various JPEG quality factors (100, 95 and 75). Contrast the difference between figures (a)-(c), whose results were obtained with EfficientNet-b0, with the figure (d) that compares detection accuracy obtained using GFR and DCTR.

four different JPEG QF (100, 95, 85 and 75). We have used Matlab® and we were warmed that this software uses native parallelization for some built-in functions; therefore, we have used two different settings, one without multithreading (using command `maxNumCompThreads(1)`) and one using multithreading and parallelization (using commands `parfor` and `parpool(·)`) with as many jobs as cores (12 in our case). The computation time has been measured on Matlab®version 9.6 (R2019a), using a DELL Precision 7730 laptop equipped with Intel®Xeon®E-2176M CPU @ 3.30GHz along with 32GB DDR4 memory @ 2667 MB/s.

One can note from Table VI that UERD, whose associated costs are extremely simple to compute is the fastest. On the opposite, J-UNIWARD that uses a wavelet decomposition and compute the ratio of wavelet coefficients after and before embedding for different direction is almost 8 times slower. The proposed method lies within those two extremes: it is slightly slower than UERD but more than four times faster than J-UNIWARD. One can note that the implementation of the proposed method uses built-in Matlab function multithreading (especially matrix multiplication and Wiener filtering) which makes it slightly less efficient when used in a parallelized manner.

Note that we have worked to improve the reference implementation with the goal to make it very efficient by simplifying many operations especially the three following (1) changing the convergence criterion to reduce the number of iterations to determine the Lagrange multiplier in (32) (2) using tabulated values for inverse function with reduced sampling. This latter simplification leads the proposed algorithm, in some case, to associate to large number of coefficients the exact same change probability. We have noted that this does not reduce the security, at least not in a measurable manner. This shows experimentally something that is generally acknowledged in the community, the security of embedding scheme mostly comes from the capability to select carefully pixels is which data should be hidden more than assigning very accurately change probabilities. As already mentioned we focus in the present paper on "simulated embedding" at the Shannon theoretical bound (6). However in practice one has to perform "actual" data hiding using a coding method to get closer to this bound such as the STC [3]. It should be emphasized that in such a case the proposed method is slower by about 0.8 1sec. Indeed, the proposed method requires to compute the desired change probabilities and then to turn them into costs usable by STC.

C. Comparison of Side-Informed Schemes

To conclude with the numerical experimentation we provide in Figure 2 a numerical comparison of state-of-the-art Side-Informed (SI) steganography. In this experimentation, we have included SI-UNIWARD [54] as well as SI-EBS [15]; the former is adopted as the most secured while the latter is considered as the sole effective competitor. In addition, we have added comparison with the model-based method proposed in [29] due its methodology that is similar to the one proposed in this paper.

Figure 2(a)-(c) show the average P_E obtained over BOSS and BOWS datasets using EfficientNet-b0 without the first pooling. One can see that the Side-Informed version of the proposed embedding scheme always outperforms SI-UNIWARD by up to $+1.5\%$ for QF100, $+1.2\%$ for QF95 and more than $+5\%$ for QF75. Contrast those results with those presented in Figure 2(d) against features-based detector using DCTR and GFR for QF75 : one can note that SI-UNIWARD seems more detectable in this case; similar results have been obtained for the other QF. This confirms the previous results presented for non-SI schemes, that UNIWARD is extremely robust against Deep Learning based steganalysis yet the method proposed in this paper is still slightly better.

On the opposite, SI-EBS shows a similar trends as UERD for non-SI case : both their security is substantially worse when switching from features-based detection to Deep Learning approaches. This may be explained by the fact that those two embedding schemes have been tailored to be robust against those features-based steganalysis. All those results show that it is not possible anymore to assess the security of steganography without Deep Learning based approaches.

VII. Conclusion

The present paper proposes a novel method for steganography of JPEG images that significantly differ from prior art that aims at designing, in a more or less ad-hoc manner, a cost function. On the opposite, we propose to exploit hypothesis testing theory in order to assess the statistical performance of optimal most powerful test in the worst case where all distribution parameters for each and every pixel are available at the detector. Within this "worst-case scenario", we leverage the closed-form expression of this optimal statistical performance to design a data hiding method that specifically aims at minimizing this performance. While explicit solution of this minimization is not straightforward, we simplified the statistical model in order to propose to find a tradeoff between accuracy and application through into a practical implementation that is efficient on a computational point of view. In addition, an extension of the method is presented for the Side-Informed case, when the embedder has the uncompressed image at it disposal. In both cases, the method is shown to be more efficient the current state-of-the-art as well as more secured against most efficient steganalysis approaches including those based on Deep Learning.

We are aware of the limitation of the statistical model upon which the proposed embedding method is based, especially assuming independence between DCT coefficients. We have detailed the method proposed for those estimation and explained why estimation of covariance between DCT coefficients within a single image in, to the best of our knowledge, currently out-of-reach. While we acknowledge that being able to use and estimate a more accurate statistical model of JPEG would enlarge the application, for instance to measure how many bits can be embedded "safely", the present method can be used "at it" to decide how to spread the payload into several images [31], [65]. Using a more accurate statistical model may also allow improving the efficiency of the present method; it constitutes a difficult research topic on its own [30].

Appendix A
Study of the Likelihood Ratio

Using the expressions for $p_0(\bar{z}_{m,n})$ (5) and $p_{\beta_{m,n}}(\bar{z}_{m,n})$ (9), let us compute the expression of the LR (11) for one observation $\Lambda\left(\bar{z}_{m,n}\right)$. For the sake of clarity, in this first part the index (m, n) for the location will be omitted:

$$\frac{(1-2\beta)\exp\!\left(\frac{-\Delta^2\bar{z}^2}{2\sigma^2}\right)+\beta\!\left[\exp\!\left(\frac{-\Delta^2(\bar{z}+1)^2}{2\sigma^2}\right)+\exp\!\left(\frac{-\Delta^2(\bar{z}-1)^2}{2\sigma^2}\right)\right]}{\exp\left(-\frac{\Delta^2\bar{z}^2}{2\sigma^2}\right)} \tag{35}$$

which can be simplified as follows:

$$\Lambda(\bar{z}) = 1 - 2\beta + \beta\left[\exp\!\left(\frac{\Delta^2}{\sigma^2}(\bar{z}-1/2)\right)+\exp\left(\frac{\Delta^2}{\sigma^2}(-\bar{z}-1/2)\right)\right]. \tag{36}$$

We can leverage the fine quantization assumption $\sigma \gg \Delta$, we to simplify the LR using its second-order Taylor expansion:

$$\Lambda(\bar{z}) \approx 1 + \beta\left(-\frac{\Delta^2}{\sigma^2} + \frac{\Delta^4}{\sigma^4}(\bar{z}^2 + 1/4)\right). \tag{37}$$

We can use the same fine quantization in order to simplify the log-LR $\log\Lambda(\bar{z})$ using its first-order Taylor approximation:

$$\log\Lambda(\bar{z}) = \beta\left(-\frac{\Delta^2}{\sigma^2} + \frac{\Delta^4}{\sigma^4}(\bar{z}^2 + 1/4)\right). \tag{38}$$

Keeping only the term that depends on the observation $\bar{z}$ the log-LR can be further simplified to:

$$\log\Lambda(\bar{z}) = \beta\left(-\frac{\Delta^2}{\sigma^2} + \frac{\Delta^4}{\sigma^4}\bar{z}^2\right). \tag{39}$$

which essentially depends on ${}^{\Delta^4}\!/_{\sigma^4}\bar{z}^2$. One can note that the above calculus are very similar to those presented in [24], [66] we the exception that σ is replaced by the ratio σ/Δ. This is perfectly understandable as the quantization is essentially a division which scales the standard deviation accordingly.

In order to establish the statistical performance of the most powerful LR test we will use Lindeberg's Central Limit

Theorem [49, Theorem 11.2.5]. To this end, one needs to establish the first two moment of the LR $\log\Lambda(\bar{z})$ under both hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ to characterize the asymptotic statistical performance of the LRT. Some straightforward algebra, see [24], [66], shows that:

$$\mathbb{E}_{\mathcal{H}_0}\left[\log\Lambda(\bar{z})\right] = 0, \quad \mathbb{V}ar_{\mathcal{H}_0}\left[\log\Lambda(\bar{z})\right] = \frac{2\beta^2\Delta^4}{\sigma^4}. \qquad (40)$$

$$\mathbb{E}_{\mathcal{H}_1}\left[\log\Lambda(\bar{z})\right] = \frac{2\beta^2\Delta^4}{\sigma^4}, \quad \mathbb{V}ar_{\mathcal{H}_1}\left[\log\Lambda(\bar{z})\right] \approx \frac{2\beta^2\Delta^4}{\sigma^4}. \qquad (41)$$

From the expression of those moments (40)–(41) and, again, invoking Lindeberg's CLT, it is straightforward to establish the asymptotic distributions (13):

$$\frac{\log\Lambda(\mathbf{Z})}{\sqrt{2\varrho}} = \log\Lambda^\star(\mathbf{Z}) \rightsquigarrow \begin{cases}\mathcal{N}(0,1) & \text{under } \mathcal{H}_0\\ \mathcal{N}(\sqrt{2\varrho},1) & \text{under } \mathcal{H}_1\end{cases} \qquad (42)$$

$$\text{with } \varrho = \sum_{m,n}\beta_{m,n}^2\frac{\Delta_{m,n}^4}{\sigma_{m,n}^4}. \qquad (43)$$

A. Extension to SI Embedding

Let us state again that the most difficult problems is to clear state the problem. More precisely, with the method proposed in the present work, it is important to clarify what the detectors. On the one hand, we assumed the "most powerful" test that knows all required parameters and, on the other hand, side informed steganography is based on partial ignorance of the warden.

We assumed that the embedder knows the unquantized cover $c_{m,n}$ and hence the quantization error $e_{m,n}$ (17). On the opposite, we assumed that the detector is not "fully omniscient" as it cannot distinguish the change probability in each direction, hence assumes that $\beta_{m,n} = \beta_{m,n}^+ = \beta_{m,n}^-$. However, to formalize the advantage to minimize the modification, we assumed that the detector knows the quantization error and hence can determine the minimal additive modification one should do to change the LSB value:

$$\bar{s}_{m,n}\Delta_{m,n} - c_{m,n} = \begin{cases}\Delta_{m,n}(1/2 - e_{m,n}) & \text{if } e_{m,n} > 0,\\ -\Delta_{m,n}(1/2 + e_{m,n}) & \text{if } e_{m,n} < 0.\end{cases}$$

So that for the detector each stego DCT coefficients follow the distribution defined by $\mathbb{P}\left[\bar{s}_{m,n} = \bar{c}_{m,n}\right] = 1 - 2\beta_{m,n}$ and $\mathbb{P}\left[\bar{s}_{m,n}\Delta_{m,n} = c_{m,n} + 1/2\Delta_{m,n}(2e_{m,n} - \text{sign}(e_{m,n}))\right] = \beta_{m,n} = \mathbb{P}\left[\bar{s}_{m,n}\Delta_{m,n} = c_{m,n} - 1/2\Delta_{m,n}(2e_{m,n} - \text{sign}(e_{m,n}))\right]$. It is obvious that this partial knowledge of the additive steganographic modification (while not being able to distinguish each direction) lead to the very same problem as the one of non-SI steganography detection except that the changes are weighted by $1/2\Delta_{m,n}(2e_{m,n} - \text{sign}(e_{m,n}))$; hence, putting this distribution of stego element into (35) does not modify significantly the rest of the calculus and, hence, is omitted due to the space constraints.

Appendix B
Approximation the DCT Coefficients Pmf

Let $\mathbf{c}$ be a vector of independent Gaussian random variables such that:

$$c_{m,n} \sim \mathcal{N}(\theta_{m,n}, \sigma_{m,n}^2) \qquad (44)$$

Let each DCT coefficient $c_{m,n}$ be quantized with a different quantization factor $\Delta_{m,n}$, depending on their position in each 8×8 block of the DCT grid:

$$\bar{c}_{m,n} = \text{round}\left(c_{m,n}/\Delta_{m,n}\right). \qquad (45)$$

The probability mass function (pmf) of quantized DCT coefficients can then be expressed as follows:

$$\mathbb{P}\left[\bar{c}_{m,n} = k\right] = \frac{1}{\sigma_{m,n}}\int_{\Delta_{m,n}(k-1/2)}^{\Delta_{m,n}(k+1/2)}\phi\left(\frac{x-\theta_{m,n}}{\sigma_{m,n}}\right)dx \qquad (46)$$

where $\phi(\cdot)$ represents the standard Gaussian probability density function (pdf).

Assuming the quantization step is not too large compared to $\sigma_{m,n}$, we can use the well-known Taylor expansion of the function ϕ [67, p.931] around the value $\Delta_{m,n}$, the midpoint of the quantization step. A short calculation shows that:

$$\mathbb{P}\left[\bar{c}_{m,n} = k\right] = \frac{\Delta_{m,n}}{\sigma_{m,n}}\phi\left(\frac{k\Delta_{m,n}-\theta_{m,n}}{\sigma_{m,n}}\right) + \epsilon\left(\frac{\Delta_{m,n}}{\sigma_{m,n}}\right), \qquad (47)$$

where the exact analytic expression of the corrective term is:

$$\epsilon\left(\frac{\Delta_{m,n}}{\sigma_{m,n}}\right) = \sum_{i=1}^{\infty}\frac{2(-1)^i}{2^{2i}(2i+1)!}\frac{\Delta_{m,n}^{2i}}{\sigma_{m,n}^{2i}}H_{2i}\left(\frac{\Delta_{m,n}-\theta_{m,n}}{\sigma_{m,n}}\right), \qquad (48)$$

with H_i the Hermite polynomial of order i [67, p.1350]. It is obvious that (47)-(48) yields the simplification:

$$\mathbb{P}\left[\bar{c}_{m,n} = k\right] = \frac{\Delta_{m,n}}{\sigma_{m,n}}\phi\left(\frac{k\Delta_{m,n}-\theta_{m,n}}{\sigma_{m,n}}\right) + o\left(\frac{\Delta_{m,n}}{\sigma_{m,n}}\right)^2 \qquad (49)$$

where $o(g(x))$ is the Landau notation for asymptotic comparison.

Appendix C
Details on Discrete Cosine Transform Formulation

Let us recall that, in brief, JPEG compression [68], [69] operates into four main steps:

1) Colors represented using Red, Green and Blue (RGB) channels in spatial domain are changed to YCbCr (luminance/chrominance) channels through a linear transformation.
2) The image is split into blocks of size 8 × 8 pixels over which the Discrete Cosine Transform is applied, which corresponds to a change of basis vectors.
3) Then the ensuing DCT coefficients are quantized, generally speaking, with a different step for each mode/frequency (position in the 8 × 8 matrix).
4) Eventually, non-destructive entropy compression is applied to encode the quantized DCT coefficients.

Steganography operates right after step 3, thus, the last step is omitted. Besides, the present paper focuses on grayscale images which are not subjected to color change. We will thus only focus on DCT transformation and quantization and, for the sake of simplicity we will focus on one single block $x_{k,l}$ whose corresponding DCT coefficients are denoted $c_{m,n}$ with indices between 0 and 7.

Formally, the DCT coefficients are given by:

$$\mathbf{c}_{m,n} = \sum_{m,n=0}^{7} w_k w_l \cos\left(\frac{k\pi}{8}\left(m+\frac{1}{2}\right)\right)\cos\left(\frac{l\pi}{8}\left(n+\frac{1}{2}\right)\right) x_{k,l}. \tag{50}$$

with $w_k = 1$ if $k > 0$ and $w_k = 2^{-1/2}$ if $k = 0$.

that when arranging matrices into vectors, one can write the DCT coefficients as follows:

$$\mathbf{c} = \mathbf{D}\mathbf{x}, \tag{51}$$

where the components of the orthonormal matrix $\mathbf{D} = \{d_{i,j}\}$, $(i, j) \in \{0, \dots, 63\}^2$ are defined:

$$d_{i,j} = w_i \cos\left[\frac{\pi}{8}\left(\lfloor j/8 \rfloor + \frac{1}{2}\right)\lfloor i/8 \rfloor\right] \tag{52}$$

$$\times \cos\left[\frac{\pi}{8}\left(j\%8 + \frac{1}{2}\right) i\%8\right]. \tag{53}$$

where $w_i = 2^{-1}$ if $i = 0$, $w_i = 2^{-1/2}$ if $i\%8 = 0$, $i > 0$ and $w_i = 1$ otherwise.

If follows from the Gaussian model (1) of non-uniformly distributed pixels and from its stability with respect to linear transformation that the DCT coefficients can be modeled as:

$$\mathbf{c} \sim \mathcal{N}\left(\mathbf{D}\boldsymbol{\mu}' , \, \mathbf{D}\boldsymbol{\Sigma}'\mathbf{D}^{\top}\right). \tag{54}$$

where $\boldsymbol{\mu}' = \mathbf{T}\boldsymbol{\mu}$ represents the expectation of rendered pixels and $\boldsymbol{\Sigma}' = \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}^{\top}$ their covariance matrix.

Acknowledgment

This work was granted access to the HPC Resources of IDRIS under the allocation 2021-AP011012582 made by GENCI.

This article is provided with two reference source-codes available under DOI: 10.24433/CO.2423893.v3 (for JMiPOD) and 10.24433/CO.7938898.v2 (for SIMiPOD).

References

[1] A. D. Ker *al.*, "Moving steganography and steganalysis from the laboratory into the real world," in *Proc. ACM IH&MMSec*, 2013, pp. 45–58.

[2] A. Westfeld, "F5—A steganographic algorithm," in *Proc. Int. Workshop Inf. Hiding* (Lecture Notes in Computer Science), vol. 2137. Berlin, Germany: Springer, 2001, pp. 289–302.

[3] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[4] A. Westfeld and P. Andreas, "Attacks on steganographic systems," in *Proc. 3rd Int. Workshop Inf. Hiding*, vol. 1999, pp. 61–76.

[5] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[6] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *Proc. IEEE Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 48–53.

[7] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

[8] R. Cogranne and J. Fridrich, "Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2627–2642, Dec. 2015.

[9] R. Cogranne, V. Sedighi, J. Fridrich, and T. Pevný, "Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?" in *Proc. IEEE Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[10] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016.

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

[12] J. Huang, J. Ni, L. Wan, and J. Yan, "A customized convolutional neural network with low model complexity for JPEG steganalysis," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2018, pp. 198–203.

[13] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, pp. 1–13, Dec. 2014.

[14] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2014, pp. 4206–4210.

[15] C. Wang and J. Ni, "An efficient JPEG steganographic scheme based on the block entropy of DCT coefficients," in *Proc. IEEE Int. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1785–1788.

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

[17] V. Sedighi, J. Fridrich, and R. Cogranne, "Toss that BOSSbase, Alice!" *Electron. Imag.*, vol. 2016, no. 8, pp. 1–9, Feb. 2016.

[18] Q. Giboulot, R. Cogranne, D. Borghys, and P. Bas, "Effects and solutions of cover-source mismatch in image steganalysis," *Signal Process., Image Commun.*, vol. 86, Aug. 2020, Art. no. 115888.

[19] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[20] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2074–2087, Aug. 2019.

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

[22] J. Fridrich and J. Kodovskỳ, "Multivariate Gaussian model for designing additive distortion for steganography," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 2949–2953.

[23] S. Vahid, J. Fridrich, and R. Cogranne, "Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model," *Proc. SPIE Electron. Imag.*, vol. 9409, Mar. 2015, Art. no. 94090H.

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

[25] E. Giboulot, P. Bas, and R. Cogranne, "Synchronization minimizing statistical detectability for side-informed JPEG steganography," in *Proc. IEEE Inf. Forensics Secur. (WIFS)*, 2020, pp. 1–6.

[26] E. Giboulot, R. Cogranne, and P. Bas, "JPEG steganography with side information from the processing pipeline," *Proc. IEEE Int. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2767–2771.

[27] E. Giboulot, R. Cogranne, and P. Bas, "Detectability-based JPEG steganography modeling the processing pipeline: The noise-content trade-off," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2202–2217, 2021.

[28] A. D. Ker, T. Pevny, and P. Bas, "Rethinking optimal embedding," in *Proc. ACM IH&MMSec*, 2016, pp. 93–102.

[29] T. Denemark and J. Fridrich, "Model based steganography with precover," *Electron. Imag.*, vol. 2017, no. 7, pp. 56–66, 2017.

[30] T. Taburet, P. Bas, W. Sawaya, and R. Cogranne, "JPEG steganography and synchronization of DCT coefficients for a given development pipeline," in *Proc. ACM IH&MMSec*, 2020, pp. 139–149.

[31] R. Cogranne, Q. Giboulot, and P. Bas, "Steganography by minimizing statistical detectability: The cases of JPEG and color images," in *Proc. ACM IH&MMSec*, 2020, pp. 161–167.

[32] R. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, no. 6, pp. 835–839, Jun. 1983.

[33] E. Y. Lam, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[34] F. Müller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electron. Lett.*, vol. 29, no. 22, pp. 1935–1936, Oct. 1993.

[35] R. Böhme and A. Westfeld, "Breaking Cauchy model-based JPEG steganography with first order statistics," in *Proc. ESORICS*, vol. 3193. Berlin, Germany: Springer, 2004, pp. 125–140.

[36] T. H. Thai, R. Cogranne, and F. Retraint, "Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1980–1993, May 2014.

[37] T. Pevny and A. D. Ker, "Exploring non-additive distortion in steganography," in *Proc. ACM IH&MMSec*, New York, NY, USA, 2018, pp. 109–114.

[38] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: A blind image denoising algorithm," *Image Process. Line*, vol. 5, pp. 1–54, Jan. 2015.

[39] F. Feschet, "Implementation of a denoising algorithm based on high-order singular value decomposition of tensors," *Image Process. Line*, vol. 9, pp. 158–182, Jun. 2019.

[40] S. Hurault, T. Ehret, and P. Arias, "EPLL: An image denoising method using a Gaussian mixture model learned on a large set of patches," *Image Process. Line*, vol. 8, pp. 465–489, Dec. 2018.

[41] Y. Le Montagner, E. D. Angelini, and J.-C. Olivo-Marin, "An unbiased risk estimator for image denoising in the presence of mixed Poisson–Gaussian noise," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1255–1268, Mar. 2014.

[42] Y. Reibel, M. Jung, M. Bouhifd, B. Cunin, and C. Draman, "CCD or CMOS camera noise characterisation," *Eur. Phys. J. Appl. Phys.*, vol. 21, no. 1, pp. 75–80, Jan. 2003.

[43] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[44] T. Denemark and J. Fridrich, "Improving steganographic security by synchronizing the selection channel," in *Proc. ACM IH&MMSec*, Jun. 2015, pp. 5–14.

[45] W. Li, W. Zhang, K. Chen, W. Zhou, and N. Yu, "Defining joint distortion for JPEG steganography," in *Proc. ACM IH&MMSec*, 2018, pp. 5–16.

[46] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1905–1917, Sep. 2015.

[47] C. Zitzmann, "Statistical decision methods in hidden information detection," in *Information Hiding* (Lecture Notes in Computer Science). New York, NY, USA: Springer-Verlag, 2011, pp. 163–177.

[48] C. Kin-Cleaves and A. D. Ker, "Simulating suboptimal steganographic embedding," in *Proc. ACM IH&MMSec*, 2020, pp. 121–126.

[49] E. Lehmann and J. Romano, *Testing Statistical Hypotheses*. New York, NY, USA: Springer, 2005.

[50] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Information Hiding* (Lecture Notes in Computer Science), vol. 6958. Berlin, Germany: Springer, 2011, pp. 59–70. [Online]. Available: agents.fel.cvut.cz/boss/

[51] P. Bas and T. Furon. (Jul. 2007). *Bows-2 Contest Break Our Watermarking System*. [Online]. Available: http://bows2.ec-lille.fr/

[52] R. Cogranne, Q. Giboulot, and P. Bas, "The ALASKA steganalysis challenge: A first step towards steganalysis," in *Proc. ACM IH&MMSec*, 2019, pp. 125–137.

[53] R. Cogranne, E. Giboulot, and P. Bas, "ALASKA#2: Challenging academic research on steganalysis with realistic images," in *Proc. IEEE Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.

[54] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 219–228, Feb. 2015.

[55] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganaly-sis of adaptive JPEG steganography using 2D Gabor filters," in *Proc. 3rd ACM workshop Inf. hiding multimedia Secur.*, 2015, pp. 15–23.

[56] T. D. Denemark, M. Boroumand, and J. Fridrich, "Steganalysis features for content-adaptive JPEG steganography," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1736–1746, Aug. 2016.

[57] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, 2019, pp. 6105–6114.

[58] M. Tan and Q. V. Le. (2019). *Mixconv: Mixed Depthwise Convolutional Kernels.* [Online]. Available: https://alaska.utt.fr

[59] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," 2021, *arXiv:2102.06171*. [Online]. Available: http://arxiv.org/abs/2102.06171

[60] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," 2021, *arXiv:2104.00298*. [Online]. Available: http://arxiv.org/abs/2104.00298

[61] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich, "ImageNet pre-trained CNNs for JPEG steganalysis," in *Proc. IEEE Inf. Forensics Secur.(WIFS)*, Dec. 2020, pp. 1–6.

[62] K. Chubachi, "An ensemble model using CNNs on different domains for ALASKA2 image steganalysis," in *Proc. IEEE Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.

[63] Y. Yousfi, J. Butora, J. Fridrich, and Q. Giboulot, "Breaking ALASKA: Color separation for steganalysis in JPEG domain," in *Proc. ACM IH&MMSec*, 2019, pp, pp. 138–149.

[64] J. Butora and J. Fridrich, "Effect of JPEG quality on steganographic security," in *Proc. ACM IH&MMSec*, 2019, pp. 47–56.

[65] R. Cogranne, V. Sedigi, and J. Fridrich, "Practical strategies for content-adaptive batch steganography and pooled steganalysis," *Proc. IEEE Int. Acoust., Speech Signal Process. (ICASSP)*, May 2017, pp. 2122–2126.

[66] R. Cogranne and F. Retraint, "An asymptotically uniformly most powerful test for LSB matching detection," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 464–476, Mar. 2013.

[67] E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*. Boca Raton, FL, USA: CRC Press, 2003.

[68] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[69] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, Apr. 1991.