Multivariate Side-Informed Gaussian Embedding Minimizing Statistical Detectability

Quentin Giboulot, Patrick Bas, Rémi Cogranne

Multivariate Side-Informed Gaussian Embedding Minimizing Statistical Detectability

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

***Abstract*—Steganography schemes based on a deflection criterion for embedding posses a clear advantage against schemes based on heuristics as they provide a direct link between theoretical detectability and empirical performance. However, this advantage depends on the accuracy of the cover and stego model underlying the embedding scheme. In this work we propose an original steganography scheme based on a realistic model of sensor noise, taking into account the camera model, the ISO setting and the processing pipeline. Exploiting this statistical model allows us to take correlations between DCT coefficients into account. Several types of dependency models are presented, including a very general lattice model which accurately models dependencies introduced by a large class of processing pipelines of interest. We show in particular that the stego signal which minimizes the KL divergence under this model has a covariance proportional to the cover noise covariance. The resulting embedding scheme achieves state-of-the-art performances which go well beyond the current standards in side-informed JPEG steganography.**

I. Introduction

One of the stated goals of steganography is to design algorithms able to hide information in an innocuous medium, henceforth named a cover object. As of today, steganography as a discipline has concentrated its effort on using digital media as covers, particularly digital images which will be the focus of this work. Modern steganography is based on two main ingredients: a cost function and a coding scheme. On the one hand, coding schemes improve steganography performance by hiding more data with fewer changes. As of today, it is considered as a mostly solved problem since the Syndrome-Treillis Coding [1] scheme is very close to Shannon efficiency. On the other hand, cost functions associate to each cover element a cost of modification. The goal of the steganographer is then to minimize the overall cost under the constraint of hiding messages of a given payload size. This allows for steganography to be adaptive; the underlying heuristic being that hiding in a smooth region should be more costly than hiding in a textured region because the former should be intuitively more detectable than the latter. It should be noted that cost functions currently used in the literature are almost always additive. This means that the cost of modifying one element does not depend on modifications on other elements. This key assumption allows for tremendous simplifications when designing steganographic algorithms [1]. However, we need to forego this assumption in this work to be able to leverage correlations between DCT coefficients.

The other side of the game – detecting objects in which information has been hidden – is named steganalysis. As of today, state-of-the-art methods are mainly based on supervised machine learning techniques. Until recently, the favored approach was the use of handcrafted, high-dimensional features specifically designed to capture artifacts introduced by steganographic schemes. These features were then fed to classifiers designed to handle such high-dimensional feature sets, the most popular being an ensemble of FLD classifiers [2] and a fast ridge linear classifier [3]. However, these techniques are now being largely superseded by deep neural networks. Those were at first specifically designed for steganography [4], [5], but recent advances (see the results of the ALASKA2 competition [6]) showed that using neural networks pre-trained on ImageNet such as Efficient-Net [7] can lead to similar or better performance than these specialized networks.

A. State of the art

As was alluded to earlier, steganography is now mainly concerned with the design of effective cost functions. There exists currently two main paths for their design: heuristic and statistical. The heuristic path has been by far the most popular. The approach is based on designing cost functions able to defeat the most effective steganalyzer available on a given standard dataset. The performance of these schemes is consequently only empirically validated. This approach gave rise to the most successful schemes both in the spatial domain [8], [9] and in the JPEG domain [8], [10]. Despite this success, this approach has several limitations. First, because these techniques are only validated empirically, one can observe significant differences in performance depending on how the steganalysis is performed [11], [12]. Secondly, and more generally, this approach uses cost functions that do not *apriori* have a clear link with theoretical or even empirical detectability. Consequently, this approach gives very little insight on why a strategy works and why another does not. Worse, it makes the approach incapable of giving any theoretical guarantees except under the precise setting for which it has been designed. The second path is based on minimizing a quantity which should be *apriori* linked to theoretical and/or empirical detectability. The most recent strategy following this path, adversarial embedding, directly tries to minimize empirical detectability on a given dataset by iteratively modifying the cost function to defeat a classifier which also updates at each

The work presented in this paper was founded by the European Union's Horizon 2020 project UNCOVER (under grant 101021687), the ANR/DGA Project ALASKA (under grant ANR-18-ASTR-0009) and the ANR Project PACeS (grant ANR-21-CE39-0002).

Quentin Giboulot and Rémi Cogranne are with the LIST3N research unit, Troyes University of Technology. Patrick Bas is with the CNRS, École Centrale de Lille.

step to defeat the steganographer. Though this approach has been proven to be quite successful [13], [14] it has limitations of its own, namely its high computational cost and the need for large datasets, necessary to obtain a high-quality cost function. Furthermore, it does not solve the problem of theoretical guarantees for unseen datasets and there has not been, to the best of our knowledge, a systematic study of the impact cover-source mismatch for this strategy. For these reasons, we work with another strategy in this paper which works by bounding the theoretical power of an optimal detector. This approach first appeared in [15] and culminated in the design of the MiPOD algorithm [16]. The underlying framework is based on hypothesis testing theory. The idea is to cast the steganalysis problem as a simple test between two hypotheses: $\mathcal{H}_0$, the image under scrutiny is a cover, or $\mathcal{H}_1$, it is a stego. Using this theory, one can show under some conditions that an optimal detector exists and analytically compute its statistical performance. The goal of the steganographer is then to minimize the power of this detector under the constraint of embedding a payload of a given size. However, to be able to cast the problem in this setting, one must have a model of both the cover and the stego images. The limits of MiPOD and the MG algorithm which preceded it mostly came from their choice of the noise model. Indeed both of these algorithms modeled natural images as pixels corrupted by an independent, though not identical, Gaussian noise. Furthermore, it relied on the so-called fine quantization limit assumption, which states that the variances of the pixels are greater than one. These two assumptions are both erroneous in practice. First, it is known that the neighboring pixels are, more often than not, correlated due to the impact of the processing pipeline [17], [18], [19]. Secondly, the fine quantization limit is often violated, especially in dark areas of an image due to the heteroscedastic nature of the noise [20], [21, Chapter 5, Section 5] which implies that the variance in these zones is quite small. This assumption is even more problematic if one wants to extend the approach in the JPEG domain as the quantization of DCT coefficients can greatly lower the variance before rounding.

Despite these limitations, MiPOD still enjoys close to state-of-the-art performances in the spatial domain, hence demonstrating the merit of the approach.

B. Contributions and comparisons to current approaches

The work presented in this paper is the logical continuation of our previous works on Gaussian Embedding [18], [19]. This series of work has its roots in [12] where it was observed that empirical steganalysis performance is mainly determined by only three factors: sensor, ISO, and processing pipeline. This motivated the construction of a statistical model of the noise that was only based on these three parameters. The importance of such a model was demonstrated by the success of the work on Natural Steganography culminating in [17]. In particular, this work highlighted the importance of the covariance matrix of the cover noise for the security of a steganographic scheme. However, the idea behind Natural Steganography is to imitate the noise of an image if this image was taken at a higher ISO. Its methodology thus relies on cover generation whereas our approach is based on minimizing the statistical detectability of a given stego signal to provide security guarantees to the steganographer. To leverage the covariance of the noise, [22] builds a model of the sensor noise in the JPEG domain using a very general linear model of the processing pipeline. This allows improvements on MiPOD by allowing a better estimation of the variance maps. This work also innovated by modeling the stego signal in the continuous domain (DCT domain before rounding) as a Gaussian random variable while specifying the payload constraint in the discrete domain (DCT domain after rounding). This allows foregoing the fine quantization limit assumption entirely. Despite these improvements, this algorithm only used the covariance matrix to improve its cost estimation. It does not use a multivariate Gaussian signal which, as has been shown by Natural Steganography, has a tremendous impact on security. Furthermore, recent heuristic approaches [23], [9], [24] have shown that taking into account embedding modifications performed on neighboring cover elements can lead to a significant increase in security. However, until our work, there was no statistically founded approach to explain how to take neighboring modifications into account for imperfect steganography. The subsequent work [19] improved on [18] by allowing the use of a multivariate Gaussian stego signal. This is achieved by using the Cholesky decomposition of the covariance matrix which allows recasting the difficult multivariate problem as a sequence of simple univariate ones. This paper is an extension of our paper [19] and solves many of the limitations of this works, namely:

- Our previous works assumed that the steganographer has access to the RAW image to estimate the covariance matrix of the noise. We relax this assumption greatly by requiring only the knowledge of the camera and ISO used – information usually available in the EXIF metadata – as well as black-box access to the processing pipeline.
- Our previous work assumed macro-blocks of DCT coefficients to be independent. This means that dependencies between neighboring blocks not belonging to the same macro-block were not taken into account. We solve this limitation in this work using a lattice embedding strategy akin to what was used in Natural Steganography [17].
- We extended our model to take saturation of pixels into account.
- Last but not least, numerous new experimental results, including results using recent steganalysis tools such as Efficient-Net, are now given. Full proofs of all mathematical results are also now available in the appendices.

C. Notation and terminology

When referring to *blocks* of elements in an image, we always refer to 8×8 blocks of elements. When referring to "blocks of blocks" we use the term *macro-block*. Except when explicitly stated, we manipulate blocks in their **vectorized representation** arranged lexicographically. Vectors and matrices are always written in $\mathbf{boldface}$. Vectors use small case letters while matrices use uppercase, except for collections of vectors which are denoted by bold small case letters without indices. When indexing individual elements of an image we

use the letter i. When indexing vectors of elements of an image we use the letter k. Similarly, when referring to the number of individual elements in an image we use the letter n, whereas we use the letter m when referring to the number of blocks or macro-blocks of a specified size. For the sake of clarity, we use the same symbol for a random variable and its realization. The Gaussian distribution is designated with $\mathcal{N}$. We do not differentiate the symbol between the univariate and multivariate cases as it can be inferred from the typesetting of the parameters. When referring to the distribution which is obtained by quantizing a Gaussian random variable with a uniform quantizer of step 1, we use the symbol $\mathbf{N}$. We refer to the diagonal matrix constructed with individual elements from a vector $\boldsymbol{\sigma}$ as $\mathrm{diag}(\boldsymbol{\sigma})$. Similarly, we refer to the vector constructed with the diagonal of a matrix Σ as $\mathrm{diag}(\Sigma)$.

II. Statistical model of the noise in the developed domain

In this section, we derive a model of the noise of a developed image starting from the RAW image. We summarize each step of the model derivation in Figure 1. Note that we model the noise only up to the DCT transform of the JPEG compression but not including the rounding operation as we will mostly work with a model in the continuous domain throughout this paper.

A. Model of the RAW image

We begin by giving a model of the noise of the image in the RAW domain, that is, before any processing is applied. A widely adopted model of the sensor noise is the heteroscedastic Gaussian noise model studied by Foi et al. [20]. Under this model, a RAW image is composed of n photo-sites where each photo-site x_i follows a Gaussian distribution :

$$\begin{aligned} x_i &\sim \mathcal{N}\left(\mu_i, \sigma_i^2\right), \\ \sigma_i^2 &= c_1 \mu_i + c_2, \end{aligned} \tag{1}$$

where μ_i is the value the sensor would have registered in the absence of noise. The variance of the noise, σ^2, depends linearly on μ_i through two parameters c_1 and c_2. These two parameters depend on the camera sensor and the ISO setting – see [25], [26]. At his point, note that the noise is considered **independent** between photo-sites.

B. Model of the processing pipeline

We now go on to model the processing pipeline which takes a RAW image $\mathbf{x}$ as input and outputs a developed image $\mathbf{y}$ (in the continuous domain). Many operations in image processing can be modeled as convolutions (e.g demosaicking, sub-sampling, DCT transform) or sample-wise linear functions (e.g white balance, RGB to greyscale conversion). Even though non-linearities are also present, such as in the gamma correction, in order to keep our model computationally tractable, we model the processing pipeline as a stationary linear map acting on vectors of photo-sites. By stationary, we mean that the linear map is the same for every vector of elements of the input image. Formally, we represent the linear map as a **full-rank matrix** $\mathbf{H} \in \mathbb{R}^{N \times M}$ where M and N are freely chosen by the modeler depending on trade-off between the model accuracy and computational complexity: larger M and N allows for a model of more far-reaching dependencies between DCT coefficients but will be more computationally expensive due to the increased size of the processing pipeline matrix. Note that some values are optimal for certain kinds of pipelines, for example, the work in [17] shows that choosing $M = 26^2$ and $N = 24^2$ is sufficient to model all the dependencies introduced by a bilinear demosaicking followed by a DCT transform. Finally, we require N to be a perfect square integer which is also a multiple of 8 since the DCT transform acts on 8×8 blocks of pixels.

C. Model of the developed image

To apply the processing pipeline matrix $\mathbf{H}$ as modeled in the previous subsection, we first write each macro-block of the RAW image as $\mathbf{x}_k$:

$$\mathbf{x}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \mathrm{diag}(\boldsymbol{\sigma}_k)\right). \tag{2}$$

By multiplying the matrix $\mathbf{H}$ with a macro-block $\mathbf{x}_k$ containing M elements, we obtain a new developed macro-block $\mathbf{y}_k$ which follows a multivariate Gaussian (MVG) distribution:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k, \tag{3}$$

$$\sim \mathcal{N}\left(\mathbf{H}\,\boldsymbol{\mu}_k, \Sigma_k\right), \tag{4}$$

with the covariance of $\mathbf{y}_k$ simply obtained as:

$$\Sigma_k = \mathbf{H}\mathrm{diag}(\boldsymbol{\sigma}_k)\mathbf{H}^T. \tag{5}$$

Depending on the processing pipeline, the dependency structure between the $\mathbf{y}_k$ might differ. In this paper we will only treat two models of dependency:

1) **Independent model** : Macro blocks of size $\sqrt{N} \times \sqrt{N}$ are considered independent.
2) **Lattice model** : Each block is considered to be dependent on its neighboring block – including the diagonal ones.

We illustrate each type of dependency model in Figure 2. .

Finally, note that contrary to the independent model, which is heuristic in nature, the lattice model has been extensively studied and justified in the work of Natural Steganography [17] as the optimal dependency model for certain types of linear pipeline.

III. Estimation and approximation of the covariance matrix without the RAW file

Our previous work [18] gave a methodology to estimate the covariance matrix when the RAW file and processing pipeline are available. This section provides a novel method to estimate the covariance matrix without having access to the RAW file. Note that this section is an option However, we still assume access to the processing pipeline, at least as a black box, since the estimation method mostly relies on having a linear approximation of this pipeline. We will also assume that the steganographer knows the c_1 and c_2 parameters of the heteroscedastic model of the cover. This knowledge does not

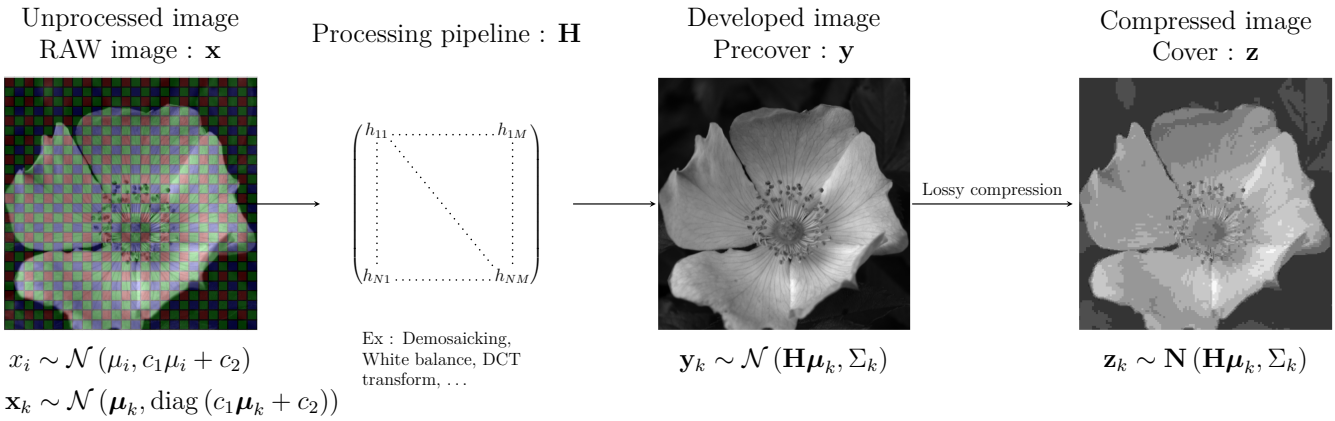


Fig. 1: Summary of the model of natural images that is described in Section II.

(a) Independent macro-block model: every non-overlapping macro-blocks of a given size is considered to be independent with the others. In this figure,the chosen size is 24×24, hence every block is only dependent with blocks with which it shares its color.

(b) Lattice model: every block is considered dependent only with its direct neighboring blocks. This induces four sets of blocks, called lattices, $\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \boldsymbol{\Lambda}_3, \boldsymbol{\Lambda}_4$ where each block in a given set is independent of every other block in the same set.

Fig. 2: The two dependency models studied in this paper

necessarily require access to the RAW image, it is sufficient to know the camera and ISO which were used to capture the cover; this information is usually available in the EXIF metadata of an image.

The estimation method presented in this section is based on estimating a generic correlation matrix which depends only on the processing pipeline. The idea is then to compute the variance of every DCT coefficient using an approximate heteroscedastic model of the DC coefficients before computing an estimation of the covariance of each block by scaling the correlation matrix using these estimated variances.

A. Heteroscedastic model of the DC coefficients

Our goal here is to show that, under some additional assumptions on the processing pipeline, the variance of the DC coefficients of each block is linear with respect to the expectation of this DC coefficient. Note that, for this section only, we distinguish between $\mathbf{H}^{DCT}$, the 64×64 matrix representing the DCT transform and $\mathbf{H}^s$ the $64 \times M$ matrix representing all operations performed in the RAW and spatial domain. We therefore have: $\mathbf{H} = \mathbf{H}^{DCT}\mathbf{H}^s$.

We assume, as usual, that the processing pipeline is both linear and stationary. Furthermore, we assume the RAW image is almost constant by block, that is, for all k we have $\boldsymbol{\mu}_k = \hat{\mu} + \mathbf{e}_k$ with $|e_{k,i}| \ll \hat{\mu}, 1 \leq i \leq M$.

First of all, let us rewrite the models of the block of photosites and of DCT coefficients:

$$\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, c_1 \mathrm{diag}(\boldsymbol{\mu}_k) + c_2), \tag{6}$$

$$\mathbf{y}_k \sim \mathcal{N}\left(\mathbf{H}^{DCT}(\mathbf{H}^s\,\boldsymbol{\mu}_k - 128), \mathbf{H}\mathrm{diag}(c_1\,\boldsymbol{\mu}_k + c_2)\,\mathbf{H}^T\right). \tag{7}$$

Note here that, contrary to Eq 3, we take into account the fact that we subtract 128 to each pixel before the DCT transform as this has an impact on the estimation method of this section (whereas it does not when using the method which uses RAW file). We can express the first two moments of the DC coefficient $y_{k,1}$ of each block:

$$\mathbb{E}[y_{k,1}] = \left(\sum_{l=1}^{M} H_{1,l}(\hat{\mu} + e_{k,l})\right) - 1024, \tag{8}$$

$$\mathrm{Var}[y_{k,1}] = c_1 \sum_{l=1}^{M} H_{1,l}^2 \hat{\mu} + H_{1,l}^2 c_2 + c_1 \sum_{l=1}^{M} H_{1,l}^2 e_{k,l}. \tag{9}$$

Let us note the following quantities:

$$\bar{H}_{i,j} \triangleq \frac{\sum_{l=1}^{M} H_{i,l}^2}{\sum_{l=1}^{M} H_{j,l}}, \bar{H}_{i,j}^{(2)} \triangleq \frac{\sum_{l=1}^{M} H_{i,l}^2}{\sum_{l=1}^{M} H_{j,l}^2}. \tag{10}$$

The variance of $y_{k,1}$ can be expressed using its expectation:

$$\begin{aligned}\mathrm{Var}[y_{k,1}] &= c_1 \bar{H}_{1,1} \mathbb{E}[y_{k,1}] + \sum_{l=1}^{M} H_{1,l}^2 c_2 + 1024 \cdot c_1 \bar{H}_{1,1} \\ &\quad + c_1 \left(\sum_{l=1}^{M} H_{1,l}^2 e_{k,l} - \bar{H}_{1,1} \sum_{l=1}^{M} H_{1,l} e_{k,l}\right) \\ &\triangleq c_1^{DC} \mathbb{E}[y_{k,1}] + c_2^{DC} + \mathrm{error},\end{aligned} \tag{11}$$

with c_1^{DC} and c_2^{DC} defined as:

$$c_1^{DC} = c_1 \bar{H}_{1,1}, \tag{12}$$

$$c_2^{DC} = \left(\sum_{l=1}^{M} H_{1,l}^2 c_2 \right) + 1024 \cdot c_1 \bar{H}_{1,1}. \tag{13}$$

This shows that the variance of the DC coefficients is linear with their expectation up to an error which is small as long as the ratio $\bar{H}_{1,1}$ is small. As a particular case, note that if all the $H_{1,l}$ are constant, for example if the only operation of the processing pipeline performed is the DCT transform, then the error is simply 0. In this paper, we assume that the error is negligible in practice.

Note however, that the $H_{i,l}$ usually alternate sign for $i > 1$ because of the structure of the DCT transform. This leads to the ratio $\bar{H}_{i,l}$ possibly exploding and the error can not be considered small anymore for AC coefficients in general. However we can still express the variance of the AC coefficients as a function of the variance of the DC coefficient:

$$\begin{aligned} \mathrm{Var}\left[y_{k,i}\right] &= c_1 \hat{\mu} \sum_{l=1}^{M} H_{i,l}^2 + c_2 \sum_{l=1}^{M} H_{i,l}^2 + c_1 \sum_{l=1}^{M} H_{i,l}^2 e_{k,l} \\ &\simeq \bar{H}_{i,1}^{(2)} \mathrm{Var}\left[y_{k,1}\right], \end{aligned} \tag{14}$$

using the fact that $|e_{k,l}|$ is small compared to $\hat{\mu}$.

B. Processing pipeline and correlation matrix estimations

The first step is to estimate the processing pipeline matrix $\mathbf{H}$ since we only assumed a black-box access to the processing pipeline.

To do so, we first generate an image $\bar{\mathbf{x}}$ so that:

$$\bar{x}_i \sim \mathcal{N}\left(0, \sigma^2\right), \tag{15}$$

where the value of σ^2 can be freely chosen and does not impact the estimation.

Using the black-box access to the pipelines, we develop this image to obtain the developed image $\bar{\mathbf{y}}$.

Using Eq (3), we know that for a given macro-block size M, the processing pipeline outputs a macro-block of size N and that the k-th macro-block of the image follows:

$$\bar{\mathbf{y}}_k = \mathbf{H} \bar{\mathbf{x}}_k, \tag{16}$$

which can be solved using any type of linear regression method. For example, we can solve for $\mathbf{H}$ using a least-square estimation:

$$\mathbf{H} = \bar{\mathbf{y}} \bar{\mathbf{x}}^T \left(\bar{\mathbf{x}} \bar{\mathbf{x}}^T \right)^{-1}. \tag{17}$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ here correspond respectively to the $M \times m$ and $N \times m$ matrices where each line corresponds to a vectorized macro-block. Note that, here, $\mathbf{H}$ is the closest solution (in the least-square sense) to the set of equations in Eq (16) and not an exact solution.

To approximate the covariance matrices, we will also use a correlation matrix $\boldsymbol{\rho}$ based on the processing pipeline. First we compute the sample covariance matrix of $\bar{\mathbf{y}}$:

$$\Sigma_{\bar{\mathbf{y}}} = \frac{1}{m-1} \sum \bar{\mathbf{y}}_k \bar{\mathbf{y}}_k^T, \tag{18}$$

and obtain the correlation matrix:

$$\boldsymbol{\rho} = \mathrm{diag}\left(\varrho_{\bar{\mathbf{y}}}^2\right)^{-\frac{1}{2}} \Sigma_{\bar{\mathbf{y}}} \mathrm{diag}\left(\varrho_{\bar{\mathbf{y}}}^2\right)^{-\frac{1}{2}}. \tag{19}$$

where $\varrho_{\bar{\mathbf{y}}}^2 \triangleq \mathrm{diag}\left(\Sigma_{\bar{\mathbf{y}}}\right)$.

C. Approximation of the covariance matrix

With the use of the heteroscedastic parameters c_1^{DC} and c_2^{DC}, it is now possible to compute an approximation of the covariance matrix using the correlation matrix $\boldsymbol{\rho}$ computed in Section III-B. In this subsection, we denote $\mathrm{Var}\left[y_{k,i}\right]$ as $\varrho_{k,i}^2$.

We first compute an approximation of the true variance map of the DCT coefficients and use it to scale the correlation matrix:

1) Compute the variance $\varrho_{k,1}^2$ of the DC coefficient of the k-th block $\mathbf{y}_k$ as:
$$\varrho_{k,1}^2 = c_1^{DC} y_{k,1} + c_2^{DC}. \tag{20}$$
2) Compute the variances of the l-th coefficients of the k-th block simply as:
$$\varrho_{k,i}^2 = \bar{H}_{i,1}^{(2)} \varrho_{k,1}^2. \tag{21}$$
.
3) Compute the covariance matrix of the k-th block by scaling the correlation matrix $\boldsymbol{\rho}$ with the variances of the DCT coefficients of the block using the standard formula:
$$\hat{\Sigma}_k = \mathrm{diag}(\boldsymbol{\varrho}_k)\ \boldsymbol{\rho}\ \mathrm{diag}(\boldsymbol{\varrho}_k), \tag{22}$$
where $\boldsymbol{\varrho}_k$ is the vector of standard deviation of the k-th block of DCT coefficients and $\hat{\Sigma}_k$ the resulting estimation of the covariance matrix.

IV. Optimal detector

The goal of the steganographer is to find the optimal prestego signal to use in order to evade the steganalyst. By prestego, we refer to the stego signal in the continuous domain, which will be the only domain of interest in this section. Section V addresses the transition to the quantized domain in detail.

To find the form of an optimal prestego signal, three ingredients are needed:

1) A model of both the precover, in our case, the cover after the DCT transform but before rounding – see $\mathbf{y}_k$ in Figure 1 and the prestego,
2) A criterion of optimality,
3) An optimal detector built from the two first items.

Once these ingredients are available, the optimal prestego signal is the signal which minimizes the power of the optimal detector under a given payload size constraint.

In this section we provide these three elements, beginning with a choice of prestego model. The optimal detector is then derived using the precover and the prestego model under the Neyman-Pearson of optimality.

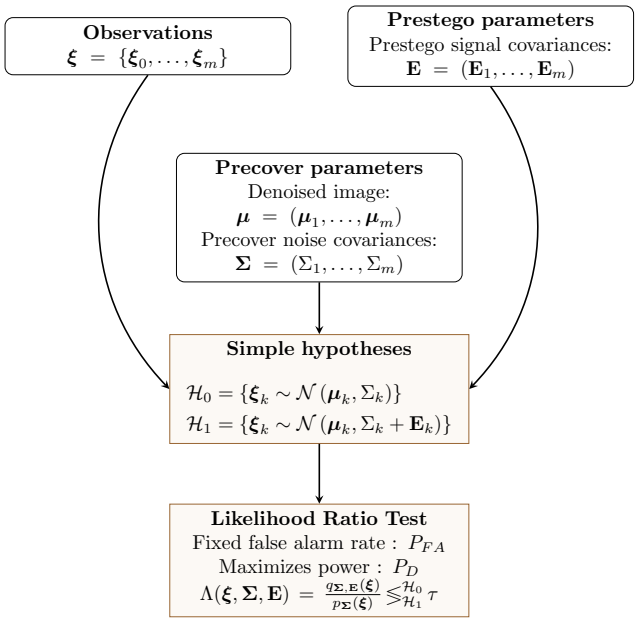


Fig. 3: Summary of the construction of the Likelihood Ratio Test for two simple hypotheses $\mathcal{H}_0$, the image is cover, $\mathcal{H}_1$, the image is stego.

A. Prestego model

Despite the fact that the precover model has been fully defined in Section II, the steganographer still needs a model of the prestego signal.

We will assume for the rest of the paper that the steganographer uses a centered multivariate Gaussian signal:

$$\mathbf{s}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{E}_k), \tag{23}$$

where $\mathbf{E}_k$ is the covariance of the prestego signal $\mathbf{s}_k$.

The signal $\mathbf{s}_k$ is then simply added to the k-th macro-block of the precover, creating the prestego macro-block $\boldsymbol{\gamma}_k$:

$$\begin{aligned} \boldsymbol{\gamma}_k &= \mathbf{y}_k + \mathbf{s}_k, \\ &\sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k + \mathbf{E}_k). \end{aligned} \tag{24}$$

This choice of distribution, though we do not prove its optimality, is motivated by three nice properties of this distribution. First, the sum of two Gaussian random variables is also a Gaussian random variable which obviously facilitates the derivations in this paper. Secondly, for a given expectation and variance, the Gaussian distribution is the maximum entropy distribution in the continuous domain [27], we can thus expect the distribution to maximize the embedded payload for a given KL-divergence. Finally, using a multivariate Gaussian distribution allows us to construct extremely efficient algorithms to compute the stego signal is the quantized domain – see Section V.

B. Optimal detector

Now that both the precover and prestego model are defined, it is possible to construct the optimal detector. To do so, we use the Neyman-Pearson criterion of optimality [28, Chapter 3, Section 2]. In this setting the steganalyst constructs a test $\delta : \mathbb{R}^n \to \{\mathcal{H}_0, \mathcal{H}_1\}$ which maximizes the power of all possible tests $P_D \triangleq \mathbb{P}(\delta(x) = \mathcal{H}_1 | \mathcal{H}_1)$ whose false-alarm probability $P_{FA} \triangleq \mathbb{P}(\delta(x) = \mathcal{H}_1 | \mathcal{H}_0)$ is upper bounded by a chosen α_0.

We also assume a worst-case adversary for the steganographer and as such consider that the steganalyst has access to all the model parameters : $\mathbf{E} = (\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_m)$, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_m)$ and $\boldsymbol{\Sigma} = (\Sigma_1, \Sigma_2, \ldots, \Sigma_m)$. Finally, the image under scrutiny by the steganalyst, which can be either a precover or a prestego, is written as $\boldsymbol{\xi}$.

The problem of the steganalyst reduces to a choice between two hypotheses: $\mathcal{H}_0$, the image under consideration is cover, $\mathcal{H}_1$, the image under consideration is stego:

$$\begin{cases} \mathcal{H}_0 & = \{\boldsymbol{\xi}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)\}, \\ \mathcal{H}_1 & = \{\boldsymbol{\xi}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k + \mathbf{E}_k)\}. \end{cases} \tag{25}$$

Because under our model the two hypotheses are simple, the Neyman-Pearson Lemma states that the most-powerful test is the likelihood ratio test (LRT) which maximizes the power P_D for a given $P_F A \leq \alpha_0$, defined, in our case as follows:

$$\Lambda(\boldsymbol{\xi}, \boldsymbol{\Sigma}, \mathbf{E}) = \frac{q_{\boldsymbol{\Sigma},\mathbf{E}}(\boldsymbol{\xi})}{p_{\boldsymbol{\Sigma}}(\boldsymbol{\xi})} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \tau, \tag{26}$$

where $p_{\boldsymbol{\Sigma}}$ and $q_{\boldsymbol{\Sigma},\mathbf{E}}$ are the joint pdf of all macro-blocks of the precover and of the prestego respectively. The threshold τ is fixed in advance by the steganalyst depending on her choice of $P_{FA} = \mathbb{P}(\delta(x) > \tau | \mathcal{H}_0) = \alpha_0$.

The study of the performance of the LRT is detailed in Appendix A and B.

V. Design of a side-informed multivariate steganographic scheme

A. Problem specification and optimal stego signal

At this point of the paper, the steganographer has access to a model of both the precover and of the prestego as well as to the detector of the worst-case adversary. His goal is then to design a steganographic scheme which minimizes the power of this detector while still hiding a secret message of a given size in the cover.

Until now, all of our models have been specified in the continuous domain. However, the payload has to be embedded in the cover in the quantized domain. We make the assumption that minimizing the power of the LRT in the continuous domain also minimizes the power of the optimal detector in the quantized domain. As such we specify the problem of the steganographer as minimizing the power P_D of the optimal detector in the continuous domain under a payload constraint R in the quantized domain:

$$\begin{cases} \min\limits_{\mathbf{E}} & P_D(\mathbf{E}) \\ R & = \sum\limits_{i=0}^{n} \sum\limits_{j \in \mathbb{Z}} \beta_i^{(j)} \log\left(\beta_i^{(j)}\right) \end{cases} \tag{27}$$

where $\beta_i^{(j)}$ is the probability of modifying the i-th DCT coefficient by $+j$. Note that the payload constraint is written under the assumption that we embed at the optimal rate.

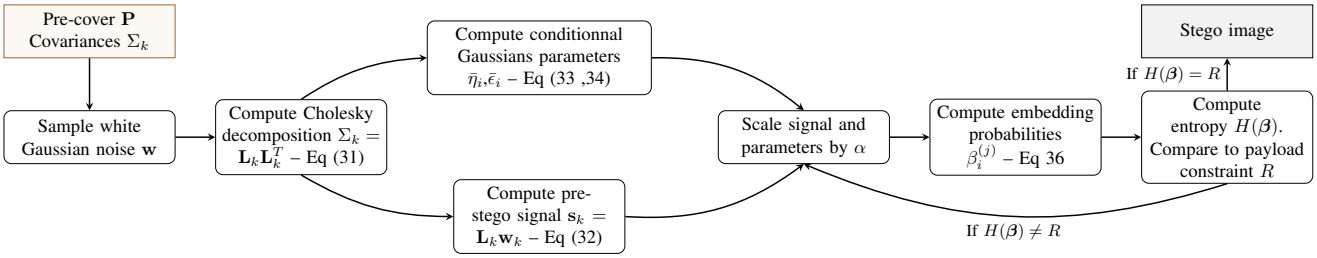


Fig. 4: Principle of all variants of Multivariate Gaussian Embedding (MGE) presented in this paper. The main idea is to generate a multivariate Gaussian stego signal with covariance proportional to the covariance of the cover noise. The Cholesky decomposition of the cover noise covariance matrix is used to generate the stego signal as well as to compute the entropy of the signal.

Due to the difficulty of obtaining an exact expression of the power of the LRT in our setting – see Appendix A – we simplify this optimization problem by minimizing the KL-divergence between the precover and prestego instead:

$$\begin{cases} \min\limits_{\mathbf{E}} & D_{KL}\left(p_{\mathbf{\Sigma}} | q_{\mathbf{\Sigma},\mathbf{E}}\right) \\ R & = \sum\limits_{i=0}^{n}\sum\limits_{j\in\mathbb{Z}} \beta_i^{(j)} \log\left(\beta_i^{(j)}\right) \end{cases} \tag{28}$$

This simplification is justified by a data-processing inequality stating that the KL-divergence between the precover and the prestego provides an upper bound on the power of the LRT [29]. Consequently, we sacrifice the guarantee of optimality while conserving a guarantee of security. We now use the key result of this paper – proven in Appendix B – which states that the prestego signal which minimizes the KL-divergence between the precover and the prestego under a given entropy in the continuous domain has the following form:

$$\mathbf{s}_k \sim \mathcal{N}\left(\mathbf{0}, \alpha\Sigma_k\right), \forall k \tag{29}$$

with $\alpha \in \mathbb{R}^+$. In other words, the prestego signal has a covariance proportional to the covariance of the precover noise. In particular, α is the same for all macro-blocks of the precover.

However, the reader should be aware that the form of the matrix in Eq (29) has only been shown to be optimal – in the KL-divergence sense – for an entropy constraint in the continuous domain. We have to assume this result to translate for an entropy constraint in the quantized domain. This is not always guaranteed. Indeed, in the case where we have $\alpha\bar{\epsilon}_i < 0.5$ for some i (see Eq (33) for a definition of $\bar{\epsilon}_i$), the expression of the entropy is a complicated function of the rounding errors r_i, $\bar{\epsilon}_i$ and $\bar{\eta}_i$ (again see Eq (34) for a definition of $\bar{\eta}_i$) and might not be strictly increasing.

In practice, these cases are rare enough in a single image, for payloads of interest, that we have not observed their impact. Therefore, we assume Eq (29) to hold for payload constraint in the quantized domain and, in particular, that $\sum_{i=0}^{n}\sum_{j\in\mathbb{Z}} \beta_i^{(j)} \log\left(\beta_i^{(j)}\right)$ is strictly increasing in α. Notice that, in the particular case where all the $\bar{\epsilon}_i$ are large enough, the entropy is well approximated by $\sum_{i=1}^{n} \frac{1}{2}\log\left(2\pi e\left(\bar{\epsilon}_i + \frac{1}{12}\right)^2\right)$.

Following this assumption, the system in Eq (28) is easily solved by a simple bisection search on α.

However, to be able to compute the value of the payload size for a given α it is necessary to compute all individual $\beta_i^{(j)}$ which is not an obvious task since we want to compute an individual value in the quantized domain from a multivariate signal in the continuous domain. The next subsection addresses this difficulty and presents a solution that does not rely on expensive Monte-Carlo simulations.

Note however, that for the rest on this paper, we only discuss methods related to **simulating** the pre-stego signal as simply and effectively as possible. These methods can be adapted to practical uses with a multi-layered STC by using the rejection sampling method used in [17, Section V.C].

B. Computing embedding probabilities

First, let $\mathbf{s}_k$ be a pre-stego signal macro-block. It follows a centered multivariate Gaussian (MVG) random variable (rv) of N elements with a full-rank covariance $\mathbf{E}_k$. Denote the i-th element of $\mathbf{s}_k$ as $s_{k,i}$. Then we have, for all i:

$$s_{k,i} | s_{k,1}, s_{k,2}, ...s_{k,i-1} \sim \mathcal{N}\left(\bar{\eta}_{k,i}, \bar{\epsilon}_{k,i}^2\right). \tag{30}$$

In other words, every element of an MVG rv conditioned on all its previous elements follows a univariate Gaussian distribution with mean $\bar{\eta}_{k,i}$ and variance $\bar{\epsilon}_{k,i}^2$.

Secondly, since $\mathbf{E}_k$ is a full-rank covariance matrix, it is symmetric positive definite. Consequently it can factorized uniquely by a lower triangular matrix $\mathbf{L}_k$ with positive diagonal entries [30, Corollary 7.2.9]:

$$\mathbf{E}_k = \mathbf{L}_k\mathbf{L}_k^T, \tag{31}$$

which corresponds to the Cholesky decomposition of the covariance matrix.

Finally, let $\mathbf{w}_k$ be a vector of M univariate standard Gaussian rvs. We can correlate white noise by multiplying it by the Cholesky decomposition of a chosen covariance matrix:

$$\mathbf{L}_k\mathbf{w}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{E}_k\right). \tag{32}$$

The parameters $\bar{\boldsymbol{\eta}}$ and $\bar{\boldsymbol{\epsilon}}$ can be computed efficiently using the Cholesky decomposition and the realization of $\mathbf{s}$ using the following equations:

$$\bar{\boldsymbol{\epsilon}}_k = \operatorname{diag}(\mathbf{L}_k), \tag{33}$$

$$\bar{\boldsymbol{\eta}}_k = \left(\mathbf{L}_k \mathbf{w}_k - \operatorname{diag}(\mathbf{L}_k)\mathbf{w}_k\right). \tag{34}$$

Now observe that all of this methodology can be applied to every macro-block of the prestego signal $\mathbf{s}_k$ as defined in Eq (29). If we apply Eq (33 ,34) to every $\mathbf{s}_k$, we obtain a vector $\bar{\mathbf{s}}$ of n elements such that:

$$\bar{s}_i \sim \mathcal{N}\left(\bar{\eta}_i, \bar{\epsilon}_i\right). \tag{35}$$

Finally, using the chain rule of probability on each $\beta_i^{(j)}$, the embedding probabilities $\beta_i^{(j)}$ are obtained by:

$$\begin{aligned}\beta_i^{(j)} = &\boldsymbol{\Phi}\left(\frac{j - r_i - \bar{\eta}_i + 0.5}{\bar{\epsilon}_i}\right) \\ &- \boldsymbol{\Phi}\left(\frac{j - r_i - \bar{\eta}_i - 0.5}{\bar{\epsilon}_i}\right),\end{aligned} \tag{36}$$

where $\boldsymbol{\Phi}(\cdot)$ represents the cumulative distribution function of the standard normal distribution and $r_i = y_i - [y_i]$ denotes the rounding error of i-th DCT coefficient. In practice, we perform a $2q+1$-ary embedding. Consequently, the alphabet size of the embedding scheme is finite; j is thus constrained to a finite range q and the $\beta_i^{(j)}$ must be normalized accordingly:

$$\beta_{i,\text{normalized}}^{(j)} = \frac{\beta_i^{(j)}}{\sum_{j=-q}^{q} \beta_i^{(j)}}. \tag{37}$$

To understand Equation (36), observe that after adding the prestego signal, the new value will either stay in the same integer bin after rounding or fall into a neighboring one. The probability of falling into one bin or another depends on the original rounding error r_i, the conditioned mean $\bar{\eta}_i$ and variance $\bar{\epsilon}_i^2$ of the prestego signal – see Figure 5. Computing $\beta_i^{(j)}$ is then simply a matter of computing the probability of falling into each bin which is simply the area under the curve of the prestego signal centered at the original rounding error inside each integer bin.

Note that for the STC implementation as described in [17, Section V.C], one would have to convert the embedding probabilities into costs using the following equation:

$$\beta_i^{(j)} = \frac{e^{-\lambda \rho_i^{(j)}}}{1 + \sum_{j \neq 0} e^{-\lambda \rho_i^{(j)}}}, \tag{38}$$

where $\rho_i^{(j)}$ is the cost of modifying the i-th DCT coefficient by $+j$ and λ is the Lagrange multiplier which allows solving Eq (28).

VI. Practical implementation

We are now ready to give the algorithms for the simulators of the steganographic scheme outlined in Section V. One algorithm is given for each of the two dependency models defined in Section II-C, namely the independent macro-block model and the lattice model.

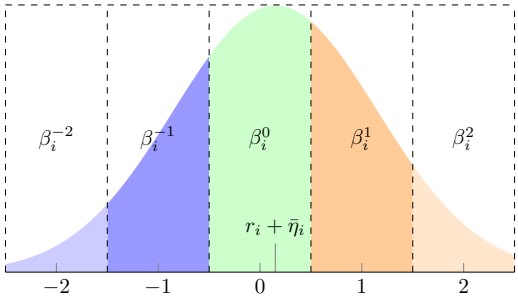


Fig. 5: Figure explaining how $\beta_i^{(j)}$ are computed with a prestego signal centered at the original rounding error r_i. The dashed boxes represent the integer bins after rounding the DCT coefficient whereas the colored parts represent the area under the curve of the pre-stego signal which will be taken into account for each $\beta_i^{(j)}$.

All the algorithms specified in this section assume that the steganographer has access to a precover $\mathbf{y}$, its corresponding RAW file $\mathbf{x}$ and full knowledge of the processing pipeline. It is assumed that the covariance matrices of the cover are estimated using the method given in [18] which will not be repeated here due to space considerations.

The effect of pixel saturation, which is not directly taken into account by our model, is discussed a the end of the section.

A. Independent macro-block model

The first dependency model assumes that all the elements in the same $\sqrt{N} \times \sqrt{N}$ macro-block are possibly dependent while elements in two different macro-blocks are considered independent. As shown in Figure 2a, only non-overlapping macro-blocks are considered.

To build the macro-blocks in this model, one just has to split the whole image $\mathbf{y}$ into non-overlapping macro-blocks $\mathbf{y}_k$ of the same size so that every element y_i of the precover belongs to one and only one macro-block.

Consequently we have each $\mathbf{y}_k$ independent from every other $\mathbf{y}_k$ and following a MVG distribution:

$$\mathbf{y}_k \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \Sigma_k\right). \tag{39}$$

To simulate the stego signal, we follow Eq (28). The steganographer first samples the prestego signal according to the precover distribution and secondly, performs a bisection search on α in order to scale the signal so that it matches the payload constraint. The signal is finally added to the precover and quantized.

B. Lattice model

The second dependency model we study in this paper is the lattice model. In this model, we assume dependencies between DCT coefficients within the same block as well as among DCT coefficients with neighboring blocks.

First, we introduce some notation that will be used throughout this subsection. We write $\boldsymbol{\Lambda}$ to denote a set of blocks,

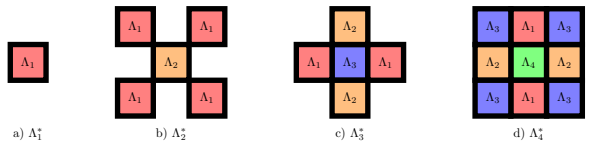


Fig. 6: Graphical representation of the blocks used to construct each $\mathbf{\Lambda}_i^*$. Note that the central block of each figure, corresponding to the last block of each $\mathbf{\Lambda}_i^*$ is sampled conditionally on each of its drawn neighboring blocks.

which we call a lattice, such that all blocks in the set are independent from one another.

For a given block $\mathbf{y}_k$ we write $\mathbf{y}_k^{\text{cardinal}}$ with $\text{cardinal} \in \{N, S, E, W, NE, NW, SE, SW\}$ to designate the block which is respectively above, below, right of, left of, etc... of $\mathbf{y}_k$.

Now, from Figure 2b, observe that the lattice model assumptions lead to a natural decomposition of the image into four lattices $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4$.

The steganographer's goal is, first, to sample a MVG signal $\mathbf{s}$ with the same covariance as the noise of the precover. To do so, we use the fact that the pdf of the whole image can be decomposed using the chain rule of probability so that the pdf of $\mathbf{s}$ is:

$$\begin{aligned} p(\mathbf{s}) =& p\left(\mathbf{s}^{\mathbf{\Lambda}_1}\right) p\left(\mathbf{s}^{\mathbf{\Lambda}_2}|\mathbf{s}^{\mathbf{\Lambda}_1}\right) p\left(\mathbf{s}^{\mathbf{\Lambda}_3}|\mathbf{s}^{\mathbf{\Lambda}_1}, \mathbf{s}^{\mathbf{\Lambda}_2}\right) \\ & p\left(\mathbf{s}^{\mathbf{\Lambda}_4}|\mathbf{s}^{\mathbf{\Lambda}_1}, \mathbf{s}^{\mathbf{\Lambda}_2}, \mathbf{s}^{\mathbf{\Lambda}_3}\right). \end{aligned} \tag{40}$$

Consequently, we can first sample the prestego signal in $\mathbf{\Lambda}_1$ by sampling from $p\left(\mathbf{s}^{\mathbf{\Lambda}_1}\right)$, then sample the prestego signal in $\mathbf{\Lambda}_2$ according to $p\left(\mathbf{s}^{\mathbf{\Lambda}_2}|\mathbf{s}^{\mathbf{\Lambda}_1}\right)$ and so on for the two other lattices.

Now let $\mathbf{\Lambda}_1^*, \mathbf{\Lambda}_2^*, \mathbf{\Lambda}_3^*, \mathbf{\Lambda}_4^*$ be such that:

$$\mathbf{\Lambda}_1^* = \mathbf{\Lambda}_1, \tag{41}$$
$$\mathbf{\Lambda}_2^* = \{[y^{NE}, y^{NW}, y^{SE}, y^{SW}, y] | y \in \mathbf{\Lambda}_2]\}, \tag{42}$$
$$\mathbf{\Lambda}_3^* = \{[y^{N}, y^{S}, y^{E}, y^{W}, y] | y \in \mathbf{\Lambda}_3]\}, \tag{43}$$
$$\mathbf{\Lambda}_4^* = \{[y^{N}, y^{S}, y^{E}, y^{W}, y^{NE}, y^{NW}, y^{SE}, y^{SW}, y] | y \in \mathbf{\Lambda}_4]\}. \tag{44}$$

In other words, each $\mathbf{\Lambda}^*$ contains vectors built from each block in $\mathbf{\Lambda}$ along with the blocks from previous lattices on which it is dependent. See Figure 6 for a graphical representation of the dependency structure of each lattice used to construct each $\mathbf{\Lambda}^*$. In what follows, we refer to the covariance matrix of the k-th vector of the i-th lattice $\mathbf{\Lambda}_i^*$ as $\Sigma_k^{\mathbf{\Lambda}_i^*}$

By using exactly the same reasoning as in Section V-B, adapted to a multivariate setting, we can easily sample the blocks in each lattice by using the Cholesky decomposition of the covariance of each vector in each $\mathbf{\Lambda}_i^*$:

$$\mathbf{s}_k^{\mathbf{\Lambda}_i} = \mathbf{L}_k^{\mathbf{\Lambda}_i^*} \mathbf{w}_k^{\mathbf{\Lambda}_i}, \forall i \in \{1, 2, 3, 4\} \tag{45}$$

with $\mathbf{L}_k^{\mathbf{\Lambda}_i^*}$ being the rectangular submatrix of the Choleksy decomposition of $\mathbf{\Sigma}_k^{\mathbf{\Lambda}_i^*}$ referring to the central block of the k-th vector of $\mathbf{\Lambda}_i^*$. In practice, using the convention of Eq (41-44), where the central block is always at the end of each vector of $\mathbf{\Lambda}_i^*$, $\mathbf{L}_k^{\mathbf{\Lambda}_i^*}$ would be constructed by taking the 64 last lines of the Cholesky decomposition of $\mathbf{\Sigma}_k^{\mathbf{\Lambda}_i^*}$.

Once every $\mathbf{s}_k^{\mathbf{\Lambda}_i}$ is computed, the algorithm to sample the pre-stego signal is exactly the same as for the independent macro-block model.

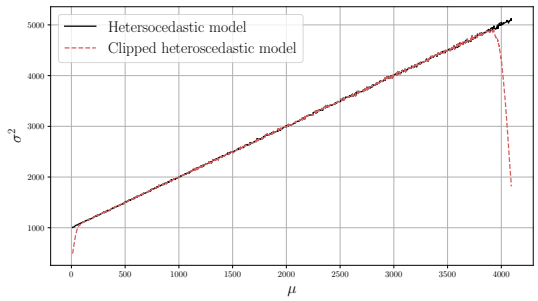


Fig. 7: Variance σ^2 of 10000 samples of $\mathcal{N}(\mu, c_1\mu + c_2)$ as function of μ with $c_1 = 1$ and $c_2 = 1000$. The curve in red is obtained when the samples are clipped at 0 and 4096 while the curve in black is obtained when the samples are not clipped.

.

C. Impact of pixel saturation

An important feature of the noise model in the RAW domain, which was not taken into account in Section II, is the saturation of photo-sites. It is known from the work of Foi et al. [25] that the heteroscedastic model breaks near saturating values of camera sensor. The variance decreases sharply in this regime due to the clipping of the photo-sites values – see Figure 7 for an example.

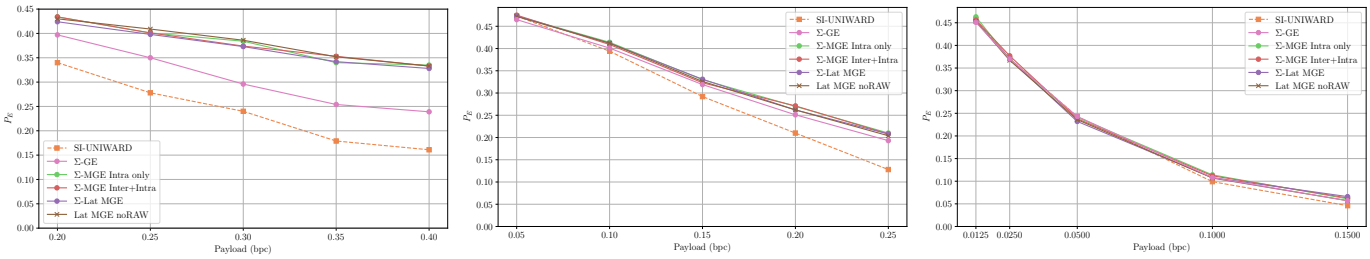
In order to solve this problem while keeping the simplicity of our model, we have experimentally found that forbidding some embedding locations is counter-productive as it destroys the covariance structure of the block where the changes are forbidden.

A better solution is to modify directly the covariance matrix in the RAW domain during the estimation phase. In particular, we empirically found that a relevant heuristic is to set the variance to 1 when the mean value of the photo-site μ_i is greater than $0.95S$ where S is the saturating value of the camera sensor.

Similarly, it is important to set a threshold for variances near zero so that the Cholesky decomposition can still be numerically computed. As long as the value is small we have not found the exact value of the threshold to matter; as a consequence, we fixed it to 10^{-5}, that is we fix the variance σ_i^2 to 10^{-5} if $\sigma_i^2 < 10^{-5}$.

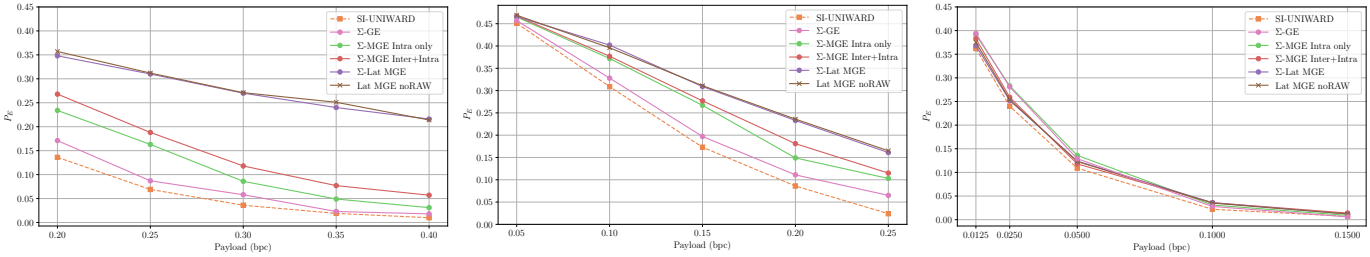
VII. Numerical evaluations and comparisons

In this section, we study the performance of our different extensions of Gaussian Embedding in the JPEG domain. To have access to a precise estimation of the covariance matrix, we use the estimation method described in our previous work [18, Section II]. Consequently, we use the knowledge of the RAW file and the processing pipeline in those cases and prefix the name of the embedding scheme with Σ. When



(a) Bosslike/QF100 (b) Bosslike/QF95 (c) Bosslike/QF75

Fig. 8: P_E as function of payload size for BossBase developed with the BOSS pipeline using Efficient-Net-b3.



(a) LIN/QF100 (b) LIN/QF95 (c) LIN/QF75

Fig. 9: P_E as function of payload size for BossBase developed with the Linear pipeline using Efficient-Net-b3.

TABLE I: Names and operations of the processing pipelines used in the experiments. The operations are performed in the order they are presented in the table

Pipeline name	**Demosaicking**	**White Balance**	**RGB to grey**	**Downsampling method**
Linear Pipeline	Bilinear	No	Yes	Edge crop, 264×264
BOSS Pipeline	PPG	Yes, Camera	Yes	Resize from 792×792 (Edge crop) to 264×264, Lanczos kernel

TABLE II: Nomenclature of the embedding schemes

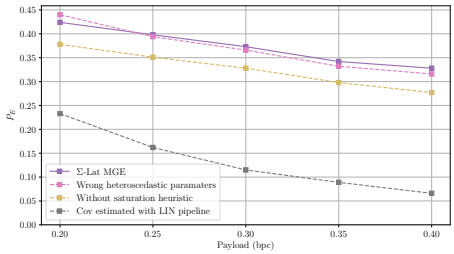
Name	**Meaning**
GE	Minimizes the power of of the MP detector in the continuous domain supposing every DCT coefficients to be independent as described in [18].
MGE Intra Only	Minimizes the KL divergence detector in the continuous domain supposing 8×8 DCT blocks to be independent.
MGE Intra+Inter	Minimizes the KL divergence in the continuous domain supposing 24×24 DCT macro-blocks to be independent.
Lat MGE	Minimizes the power of the MP detector in the continuous domain using lattice embedding as described in Section VI-B.
SI-UNIWARD	Side informed distortion based schemes as described in [8].

using the estimation method described in Section III which does not require access to the RAW dataset, we add the suffix *noRAW* at the end of the embedding scheme name. We use the BOSS RAW dataset excluding the M9 camera because of the peculiar distribution of its photonic noise (see [31], Fig. 2) which would lead to an imprecise estimation of the covariance matrix. From this dataset comprising 7240 RAW images taken with 6 different cameras, we produce two new datasets using two different processing pipelines: a linear processing pipeline and a processing pipeline close to the original BOSSBase. Both these pipelines output 264×264 greyscale JPEG images. The details are exposed in Table I. Note that for cropping we used an algorithm – Edge crop[1] – which selects crops containing the greatest number of edges – that is the zone that should contain the most textured areas. This choice of cropping was made because it is known that SI-UNIWARD does not perform well on smooth images [18, Section V.A]; using such a cropping algorithm allows us to compare our embedding schemes in a situation where SI-UNIWARD is not disfavored due to a suboptimal content choice.

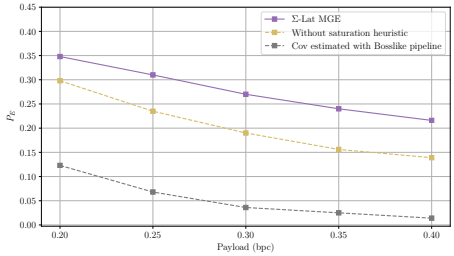
The parameters of the photonic noise c_1 and c_2 were estimated as described in [18, Section II] using the algorithm described in [26]. The $\mathbf{H}$ matrix is estimated once for each processing pipeline using the method described in Section III-B. The different embedding schemes used as well as their parameters are described in Table II. Since we performed optimal simulations of the embedding, which finds the optimal stego-signal in the continuous domain, we always chose the smallest alphabet necessary to encode all amplitudes of the stego-signal once discretized.

Steganalysis was performed with Efficient-Net-b3 [13] modified so that the stride of the stem is equal to 1. The network was trained by training from the highest payload for 30 epochs, then 10 epochs for other payloads. The model for the highest payload was initialized with ImageNet weights. The base learning rate was fixed at 0.0005 and divided by 2 on loss plateau. The batch size was fixed to 24. This configuration

[1]Available at *https://alaska.utt.fr/#material*



(a) Bosslike/QF100



(b) LIN/QF100

Fig. 10: P_E as function of payload size for different the different heuristics and design decisions made in this paper.

of Efficient-Net-b3 was the one used during the ALASKA2 competition. The rest of the parameters are initialized in the same way as in the original paper.

A. Performance evaluation

We now proceed to discuss the performance of the embedding schemes developed in this paper with SI-UNIWARD on the two datasets presented at the beginning of the section. The results are presented in Figure 8a-9c.

Beginning with QF100, regardless of the processing pipeline, there is a clear gap between the performance of SI-UNIWARD and the different variants of MGE. For the linear pipeline, the gap is, on average, of 6% and 9% in terms of absolute P_E for Σ-MGE Intra only and Σ-MGE intra + inter respectively. However, it is of 22% on average when using Σ-Lat MGE, showing the importance of using the most precise model for this pipeline. The difference between the different models is less pronounced for the Bosslike pipeline with an average gain of 13.5% in terms of absolute P_E wrt SI-UNIWARD for the MGE schemes whatever the chosen correlation model. This is due to the fact that the downsampling operation removes most correlations between blocks due to the removal of neighboring pixels before the DCT transform.

At QF95, the difference between the different schemes becomes less pronounced for the Linear pipeline. However, there is still a gap in performance between these schemes and SI-UNIWARD. Compared with Σ-Lat MGE, there is an average gain of 10.5% wrt to SI-UNIWARD for the Linear pipeline and 4% for the BOSS pipeline. At QF75, every scheme performs approximately the same irrespective of the pipeline. This is most likely due to the fact that, at such a low QF, most of the covariances are now close to 0 – see [18][Section VI.B]. As such most of the performance of the steganography is likely due to the side-information related to the rounding errors.

We also note that the implementation of Σ-Lat MGE which does not use the RAW file has a performance virtually identical with the original implementation showing that the assumptions on the processing pipeline we used in Section III are quite practical for a steganography context.

Finally, note that when using Σ-GE, which does not use any correlation between DCT coefficient, there is always a small gain with respect to SI-UNIWARD for both QF100 and QF95. However, its performance are always subpar even compared to Σ-MGE intra only, showing the importance of taking these correlations into account.

One interesting thing to note here is the impact of the processing pipeline on the correlation structure necessary to obtain good performances. In the Bosslike case, the downsampling operation has made most inter-block dependencies very small compared to intra-block dependencies. Consequently, using a more sophisticated model of dependencies does not bring any gain in performance. On the other hand, in the Linear pipeline case where all dependencies are preserved until the end of the pipeline since no downsampling is performed, there is quite a substantial gain when using the most sophisticated model.

B. Impact of the estimation of the covariance matrix

In this subsection we study the impact of errors on the estimation of the covariance matrix. Three main types of error can occur on the estimation: errors on the heteroscedastic parameters c_1 and c_2, errors on the pipeline matrix $\mathbf{H}$ and errors due to the saturation of the pixels (see Section VI-C).

Errors on the heteroscedastic parameters can be studied analytically. Let c_1 and c_2 be the true parameters; $\hat{c}_1$ and $\hat{c}_2$ are the estimated parameters. The variances σ_i and estimated variances $\hat{\sigma}_i$ in the RAW domain are given by $\sigma_i = c_1\mu_i + c_2$ and $\hat{\sigma}_i = \hat{c}_1\mu_i + \hat{c}_2$ respectively. Let $\alpha' = \frac{\hat{c}_1}{c_1}$ be relative estimation error on the c_1 parameter. We have that $\hat{\sigma}_i = \alpha'\sigma_i + C$ where $C = \hat{c}_2 - \alpha' c_2$ is a constant which does not depend on the photo-site. Consequently, the resulting estimated covariance $\hat{\Sigma}_k$ in the developed domain is given by $\hat{\Sigma}_k = \alpha'\Sigma_k + C\mathbf{H}\mathbf{H}^T$. As a particular case, note that if we have $\frac{\hat{c}_1}{\hat{c}_2} = \frac{c_1}{c_2}$, then the estimation error has no impact on the MGE steganography schemes, because a multiplicative error on the covariance does not change the optimal solution. If we take the Bosslike pipeline as an example, we computed $\mathbf{H}\mathbf{H}^T$ for each camera and found that the average absolute value of the non-diagonal entries is of the order of 10^{-7} and 10^{-4} for the diagonal elements. If we set $\hat{c}_2 = 0$, the relative estimation error would then have to be very high (i.e, in the order of at least 100) to begin to have an impact. To validate this analysis on the Bosslike pipeline, we repeated the experiments of Section VII-A except that the covariance matrices were estimated by fixing $c_1 = 0.5$ and $c_2 = 0$ for every image. The results are given in Figure 10a. As expected they are extremely close to the original Σ-Lat-MGE results on this pipeline.

The two other types of error are an error on the estimation of the processing pipeline and the error due to saturation of the pixels. In this paper, we always assumed the steganographer had access to the correct processing pipeline for the estimation of the covariance matrix, even in the "no RAW" case presented in Section III. It would then be interesting to see the impact on security when using mismatched pipelines for the covariance estimation. To do so, we repeated the experiments of Section VII-A, but this time using the covariance matrices estimated with the Boss pipeline on the Linear dataset and vice versa. Similarly, we also repeated the experiments when using the heuristic presented in Section VI-C and when not using it. The results are presented in Figure 10a-10b.

The results are clear: using strongly mismatched pipelines or not taking the saturation of pixels into account leads to near useless schemes, showing the importance of having a good model of the pipeline in the first place. Note, however, that such a mismatch is quite significant since one pipeline uses cropping and the other downsampling, leading to very different correlation structures. A complete study of this phenomenon should find a good measure of mismatch using the covariance matrices and link this measure to empirical detectability but it is out of the scope of this paper.

VIII. Conclusion

This paper is the conclusion to a series of work on Gaussian Embedding. Through the development of a multivariate Gaussian model of the sensor noise and the processing pipeline, we derived the optimal detector when the sensor noise follows this very general model. We showed that in this setting, the signal which minimizes the KL-divergence under a given payload constraint has a covariance matrix proportional to the covariance matrix of the cover noise. With these results, we designed the general form of Gaussian Embedding using a lattice embedding strategy. This allows us to use an embedding scheme taking into account correlations between every neighboring block. Furthermore, this algorithm does not need access to a RAW file. This yields an algorithm that beats the previous state-of-the-art by an important margin, even when our model does not match exactly the real distribution of the sensor noise.

References

[1] T. Filler *et al.*, "Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[2] J. Kodovsky *et al.*, "Ensemble Classifiers for Steganalysis of Digital Media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

[3] R. Cogranne *et al.*, "Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?" in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. Roma, Italy: IEEE, Nov. 2015, pp. 1–6.

[4] G. Xu *et al.*, "Structural Design of Convolutional Neural Networks for Steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, May 2016.

[5] M. Boroumand *et al.*, "Deep Residual Network for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.

[6] R. Cogranne *et al.*, "ALASKA#2: Challenging Academic Research on Steganalysis with Realistic Images," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. New York, NY, USA: IEEE, Dec. 2020, pp. 1–5.

[7] M. Tan *et al.*, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv:1905.11946 [cs, stat]*, Sep. 2020, arXiv: 1905.11946.

[8] V. Holub *et al.*, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, Dec. 2014.

[9] B. Li *et al.*, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing (ICIP)*. Paris, France: IEEE, Oct. 2014, pp. 4206–4210.

[10] L. Guo *et al.*, "Using Statistical Image Model for JPEG Steganography: Uniform Embedding Revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, Dec. 2015.

[11] V. Sedighi *et al.*, "Toss that BOSSbase, Alice!" *Electronic Imaging*, vol. 2016, no. 8, pp. 1–9, Feb. 2016.

[12] Q. Giboulot *et al.*, "Effects and solutions of Cover-Source Mismatch in image steganalysis," *Signal Processing: Image Communication*, vol. 86, p. 115888, Aug. 2020.

[13] W. Tang *et al.*, "CNN-Based Adversarial Embedding for Image Steganography," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074–2087, Aug. 2019.

[14] S. Bernard *et al.*, "Explicit Optimization of min max Steganographic Game," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 812–823, 2021.

[15] J. Fridrich *et al.*, "Multivariate gaussian model for designing additive distortion for steganography," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE, May 2013, pp. 2949–2953.

[16] V. Sedighi *et al.*, "Content-Adaptive Steganography by Minimizing Statistical Detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, Feb. 2016.

[17] T. Taburet *et al.*, "Natural Steganography in JPEG Domain With a Linear Development Pipeline," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 173–186, 2021.

[18] Q. Giboulot *et al.*, "Detectability-Based JPEG Steganography Modeling the Processing Pipeline: The Noise-Content Trade-off," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2202–2217, 2021.

[19] ——, "Synchronization Minimizing Statistical Detectability for Side-Informed JPEG Steganography," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. New York, NY, USA: IEEE, Dec. 2020, pp. 1–6.

[20] A. Foi, "Clipped noisy images: Heteroskedastic modeling and practical denoising," *Signal Processing*, vol. 89, no. 12, pp. 2609–2629, Dec. 2009.

[21] B. Widrow *et al.*, *Quantization noise: roundoff error in digital computation, signal processing, control, and communications*. Cambridge ; New York: Cambridge University Press, 2008, oCLC: ocn183916250.

[22] Q. Giboulot *et al.*, "JPEG Steganography with Side Information from the Processing Pipeline," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 2767–2771.

[23] T. Denemark *et al.*, "Improving Steganographic Security by Synchronizing the Selection Channel," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*. Portland Oregon USA: ACM, Jun. 2015, pp. 5–14.

[24] X. Hu *et al.*, "Model-based image steganography using asymmetric embedding scheme," *Journal of Electronic Imaging*, vol. 27, no. 04, p. 1, Jul. 2018.

[25] A. Foi *et al.*, "Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[26] T. H. Thai *et al.*, "Statistical Model of Quantized DCT Coefficients: Application in the Steganalysis of Jsteg Algorithm," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 1980–1993, May 2014.

[27] T. M. Cover *et al.*, *Elements of information theory*, 2nd ed. Hoboken, N.J: Wiley-Interscience, 2006, oCLC: ocm59879802.

[28] E. L. Lehmann *et al.*, *Testing statistical hypotheses*, 3rd ed., ser. Springer texts in statistics. New York, NY: Springer New York, 2010.

[29] C. Cachin, "An Information-Theoretic Model for Steganography," in *Information Hiding*, D. Aucsmith, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, vol. 1525, pp. 306–318, series Title: Lecture Notes in Computer Science.

[30] R. A. Horn *et al.*, *Matrix analysis*, second edition, corrected reprint ed. New York, NY: Cambridge University Press, 2017.

[31] T. Denemark *et al.*, "Natural Steganography in JPEG Compressed Images," *Electronic Imaging*, vol. 2018, no. 7, pp. 316–1–316–10, Jan. 2018.

[32] H. L. Van Trees *et al.*, *Detection estimation and modulation theory*, second edition ed. Hoboken, N.J: John Wiley & Sons, Inc, 2013.

APPENDIX

In this appendix, we provide proofs and derivations of the main results of the paper. First, we derive the optimal detector of the hypothesis testing problem given in Section IV. Secondly we simplify the power minimization problem as a minimization of the KL-divergence of the likelihood-ratio and show that the optimal covariance in this setting is a scaling of the covariance of the cover noise.

A. Derivation of the optimal detector

To be as general as possible, we will not consider any particular dependency model in this appendix. As such we will consider that the sample $\boldsymbol{\xi}$ is the whole image under scrutiny. Consequently we try to derive the optimal test which discriminate between the two following hypotheses:

$$\begin{cases} \mathcal{H}_0 & = \{\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)\}, \\ \mathcal{H}_1 & = \{\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma + \mathbf{E})\}. \end{cases} \tag{46}$$

Notice that we consider here the covariance of the whole image Σ and not the covariance of the blocks. Now recall that the optimal test under our setting is the likelihood-ratio test given by Eq (26). To compute the power of this test, we need to compute the distribution of the likelihood ratio under both hypotheses. To do so, we use the fact that the statistic of the LRT can be written as a quadratic form – see [32][Chapter 3, Section 3] for a derivation:

$$\frac{1}{2}\left(\boldsymbol{\xi}^T\left(\Sigma^{-1} - (\Sigma + \mathbf{E})^{-1}\right)\boldsymbol{\xi} + \log\left(\frac{|\Sigma|}{|\Sigma + \mathbf{E}|}\right)\right) \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \tau. \tag{47}$$

This test can be simplified as:

$$\hat{\Lambda}(\boldsymbol{\xi}, \boldsymbol{\Sigma}, \mathbf{E}) = \boldsymbol{\xi}^T\left(\Sigma^{-1} - (\Sigma + \mathbf{E})^{-1}\right)\boldsymbol{\xi} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \tau'. \tag{48}$$

by putting the contribution of the constant values in the threshold.

We will show that this test can be rewritten as a sum of weighted independent standard chi-squared rv.s. We will develop only the case under $\mathcal{H}_0$; the case for $\mathcal{H}_1$ follows exactly from the same method.

Let:

$$\mathbf{A} = \Sigma^{-1} - (\Sigma + \mathbf{E})^{-1}, \tag{49}$$

and decompose Σ using the symmetric square-root matrix :

$$\Sigma = \Sigma^{1/2}\Sigma^{1/2}. \tag{50}$$

Using the spectral theorem we can write:

$$\Sigma^{1/2}\mathbf{A}\Sigma^{1/2} = \mathbf{U}\mathbf{K}\mathbf{U}^T, \tag{51}$$

where $\mathbf{U}$ is an orthogonal matrix and $\mathbf{K}$ a diagonal matrix.

Finally, let $\boldsymbol{\xi}_u = \mathbf{U}^T\Sigma^{-1/2}\boldsymbol{\xi}$ and note that it follows a centered multivariate Gaussian distribution with covariance the identity matrix (by the orthogonality of $\mathbf{U}$).

Now write:

$$\boldsymbol{\xi}^T\mathbf{A}\boldsymbol{\xi} = \boldsymbol{\xi}^T\Sigma^{-1/2}\mathbf{U}\mathbf{K}\mathbf{U}^T\Sigma^{-1/2}\boldsymbol{\xi} \tag{52}$$

$$= \boldsymbol{\xi}_u^T\mathbf{K}\boldsymbol{\xi}_u = \sum_{i=0}^{M}\mathbf{K}_{ii}\xi_{u,i}^2. \tag{53}$$

The statistic in Eq (48) is thus a realization of a sum of **independent** standard chi-squared random variable weighted by the eigenvalues of $\mathbf{A}\Sigma$ and $\mathbf{A}(\Sigma + \mathbf{E})$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ respectively.

B. Form of the covariance matrix which minimizes the KL-divergence

Computing the power of the LRT when the distribution under both hypotheses are weighted sum of chi-square distribution is difficult in the general case. Furthermore, we cannot appeal to the central limit theorem since we have no information on the distribution of the weights $\mathbf{K}_{ii}$.

If we cannot give the form of the optimal pre-stego signal due to this fact, we can however give security guarantees for a given pre-stego signal. To do so, we use the result of Cachin's work on steganographic security [29], which shows, using a simple data-processing inequality, that the performance of an optimal detector is upper bounded by the KL divergence between the distributions of the two hypotheses.

Instead of minimizing the power of LRT under an entropy constraint we will thus minimize the KL-divergence $D_{\mathrm{KL}}(p||q)$ under an entropy constraint:

$$\begin{cases} \min\limits_{\mathbf{E}} & D_{\mathrm{KL}}(p_{\boldsymbol{\Sigma}}||q_{\boldsymbol{\Sigma},\mathbf{E}}) \\ R & = \frac{1}{2}\log(2\pi e|\mathbf{E}|) \end{cases} \tag{54}$$

where R is the payload constraint in the continuous domain and $|.|$ the matrix determinant. Recall that we assume here $\boldsymbol{\Sigma}$ and $\mathbf{E}$ to be both positive definite matrices.

Note that the KL divergence between p and q is given by:

$$\frac{1}{2}(\mathrm{trace}\left((\Sigma + \mathbf{E})^{-1}\Sigma\right) + \log\left(\frac{|\Sigma + \mathbf{E}|}{|\Sigma|}\right) - n). \tag{55}$$

The main idea of this proof is that we can simplify the problem by working in a basis where the covariance of the cover noise is the identity matrix. This is done by showing that the KL-divergence is invariant when changing to this basis. We then express the Lagrangian of the system as a function of the eigenvalues of $\mathbf{E}$ and show that there exists a unique minimum for our system.

First, let $\mathbf{L}$ be the Cholseky decomposition of Σ such that:

$$\Sigma = \mathbf{L}\mathbf{L}^T. \tag{56}$$

Let us also define $\mathbf{E}_w$ as:

$$\mathbf{E}_w \triangleq \mathbf{L}^{-1}\mathbf{E}\left(\mathbf{L}^{-1}\right)^T. \tag{57}$$

We now want to show the following equality:

$$D_{\mathrm{KL}}(p_{\boldsymbol{\Sigma}} \,||\, q_{\boldsymbol{\Sigma},\mathbf{E}}) = D_{\mathrm{KL}}(p_{\mathbf{I}} \,||\, q_{\mathbf{I},\mathbf{E}_w}), \tag{58}$$

where I is the identity matrix (with relevant dimensions).

To do so, we begin by showing the equality for the trace term:

$$\begin{aligned}\operatorname{trace}\left((\Sigma+\mathbf{E})^{-1}\Sigma\right) &= \operatorname{trace}\left(\mathbf{L}^T(\Sigma+\mathbf{E})^{-1}\mathbf{L}\right)\\ &= \operatorname{trace}\left(\left(\left(\mathbf{L}^T(\Sigma+\mathbf{E})^{-1}\mathbf{L}\right)^{-1}\right)^{-1}\right)\\ &= \operatorname{trace}\left(\left(\mathbf{L}^{-1}\left(\Sigma+\mathbf{E}\right)\left(\mathbf{L}^{-1}\right)^T\right)^{-1}\right)\\ &= \operatorname{trace}\left(\left(\mathbf{I}+\mathbf{L}^{-1}\,\mathbf{E}\left(\mathbf{L}^{-1}\right)^T\right)^{-1}\right)\\ &= \operatorname{trace}\left(\left(\mathbf{I}+\mathbf{E}_w\right)^{-1}\right).\end{aligned} \tag{59}$$

and secondly for the determinant term:

$$\begin{aligned}|\Sigma+\mathbf{E}\,|\cdot|\Sigma^{-1}| &= |\mathbf{I}+\mathbf{E}\,\Sigma^{-1}|\\ &= |\mathbf{I}+\mathbf{E}\,\mathbf{L}^{-1}(\mathbf{L}^{-1})^T|\\ &= |\mathbf{I}+\mathbf{E}_w\,|,\end{aligned} \tag{60}$$

with the last line obtained using Sylvester's Law of determinant. This validates the invariance of the KL-divergence to our change of basis.

Using Eq (58), we rewrite the system in Eq (54) as:

$$\begin{cases}\min\limits_{\mathbf{E}_w} & D_{\mathrm{KL}}\left(p_{\mathbf{I}}\;||\;q_{\mathbf{I},\mathbf{E}_w}\right)\\ R & = \frac{1}{2}\left(\log\left(2\pi e|\mathbf{E_w}|\right)-\log\left(2\pi e|\Sigma^{-1}|\right)\right).\end{cases} \tag{61}$$

Now, let k_i be the i-th eigenvalues of $\mathbf{E}_w$, we have that:

$$\begin{aligned}D_{\mathrm{KL}}\left(p_{\mathbf{I}}\;||\;q_{\mathbf{I},\mathbf{E}_w}\right) &= \frac{1}{2}\left(\operatorname{trace}\left((\mathbf{I}+\mathbf{E}_w)^{-1}\right)\right)\\ &\quad+\frac{1}{2}\left(\log\left(|\mathbf{I}+\mathbf{E}_w\,|\right)-n\right)\\ &= \frac{1}{2}\left(\sum_{i=1}^{n}\frac{1}{1+k_i}+\sum_{i=1}^{n}\log\left(1+k_i\right)-n\right)\end{aligned} \tag{62}$$

With a change of variable such that $k_i' = \log\left(k_i\right)$ we obtain the following optimization problem:

$$\begin{cases}\min\limits_{k_i'} & \frac{1}{2}\left(\sum_{i=1}^{n}\frac{1}{e^{k_i'}+1}+\log(e^{k_i'}+1)\right)-n\\ R' & = \sum_{i=1}^{n}k_i'.\end{cases} \tag{63}$$

where $R' = -2R+\log\left(2\pi e|\mathbf{E_w}|\right)-n\log\left(2\pi e\right)$.

Since the objective function is a sum of identical convex functions and the constraint is the sum of k_i', the problem is convex and consequently admits a single, global minimum which is attained when all k_i' are equal.

From this observation, it follows that $\mathbf{E}_w$ is a a matrix where all eigenvalues are equal. Since, $\mathbf{E}_w$ is positive definite (hence normal), this implies that it is proportional to the identity matrix:

$$\mathbf{E}_w = \alpha\mathbf{I}, \tag{64}$$

for some $\alpha > 0$.

Going back to the original system in Eq (54), the solution is obtained by:

$$\begin{aligned}\mathbf{E} &= \mathbf{L}\,\mathbf{E}_w\,\mathbf{L}^T\\ &= \mathbf{L}\alpha\mathbf{I}\mathbf{L}^T\\ &= \alpha\Sigma.\end{aligned} \tag{65}$$