JPEG Compatibility Attack Revisited

Eli Dworetzky, Jan Butora, and Jessica Fridrich, *Fellow, IEEE*

***Abstract*—The JPEG compatibility attack is a steganalysis method for detecting messages embedded in the spatial representation of images under the assumption that the cover is a decompressed JPEG. This paper focuses on improving the detection accuracy for the difficult case of high JPEG qualities and content-adaptive stego algorithms. Close attention is paid to the robustness of the detection with respect to the JPEG compressor and DCT coefficient quantizer. A likelihood ratio detector derived from a model of quantization errors of DCT coefficients in the recompressed image is used to explain the main mechanism responsible for detection and to understand the results of experiments. The most accurate detector is an SRNet trained on a two-channel input consisting of the image and its SQ error. The detection performance is contrasted with state of the art on four content-adaptive stego methods, wide range of payloads and quality factors.**

***Index Terms*—Steganography, steganalysis, JPEG, compatibility, robustness, rounding errors, deep learning, wrapped distributions**

I. Introduction

The JPEG Compatibility Attack (JCA) is a specialized image steganalysis method that can reliably detect messages embedded with spatial-domain steganography under the assumption that the cover image is a decompressed JPEG. The compression imposes strict constraints on the spatial domain representation, which allows very accurate detection of pixel modifications even for small payloads. The assumption that the cover was originally stored as JPEG is feasible as the vast majority of images are stored in the JPEG format. Steganographers might hide data in the spatial domain because it offers a larger embedding capacity or simply because the data hiding program cannot handle the JPEG format.

The attack was originally conceived in [11] based on the idea that one could prove that a given image contains blocks of 8 × 8 pixels that could not be obtained by decompressing any combination of 64 quantized Discrete Cosine Transform (DCT) coefficients. A brute force search in the form of a tree-pruning algorithm was proposed to obtain such proof. For larger quality factors (smaller JPEG quantization steps), the complexity of this search increases rapidly, which makes this attack impractical to use at scale. Moreover, since the original JPEG compressor is not available to the steganalyst, in practice the incompatibility of a block would also need to be verified w.r.t. all JPEG decompressors, which further increases the complexity and may not even be feasible.

A quantitative version of this attack that estimates the change rate introduced by Least Significant Bit (LSB) replacement was proposed in [4], [5], where a recompressed-decompressed version of the image was used as a pixel predictor in the weighted Stego-Image (WS) attack [20]. The detection accuracy of this attack is fairly robust w.r.t. errors in the estimated quantization table as well as different JPEG compressors. This approach is, however, fundamentally limited to LSB replacement and cannot detect embedding that uses LSB matching, which is the case of all modern content-adaptive stego algorithms. The same recompression predictor was also used in [27], where the number of pixels by which the stego image and its recompressed version differed was used as the detection statistic. The departure from the WS detector allowed detection of embedding operations other than LSB replacement.

An improved localized version of this attack was described in [22] by counting the number of different pixels between the image and the recompressed-decompressed version in each 8 × 8 block. A 65-dimensional histogram of these counts, which we call the *recompression residual histogram* (RRH), served as a feature vector for training a classifier. The authors reported a markedly improved detection accuracy especially for larger quality factors and small payloads.

In general, all forms of the JCA become less accurate for high qualities because the process of recompression-decompression, which is used as a powerful reference, is more affected by rounding in the spatial domain when decompressing the original cover image. The stego changes thus become harder to distinguish from recompression artifacts, which decreases the detection accuracy especially for content-adaptive steganography as the recompression artifacts and stego changes often occur in approximately the same areas of the image. Addressing these deficiencies is one of the main goals of this paper.

In the next section, we introduce the notation used throughout the paper and briefly discuss relevant background material from the field of directional statistics. Section III describes the processing pipeline considered in this paper, which involves the initial JPEG compression of the cover image, decompression, embedding, and subsequent recompression and decompression used by the steganalyst. This pipeline is analyzed in Section IV by modeling the quantization errors during the initial compression, which allows us to obtain a detector of steganography as a

The work on this paper was supported by NSF grant No. 2028119.

The authors are with the Department of Electrical and Computer Engineering, Binghamton University, NY, 13902, USA. Email: {edworet1,jbutora1,fridrich}@binghamton.edu

likelihood ratio test (LRT) in Section V. [1] The LRT is used to obtain insight into the inner workings of the JCA and also explain the trends in detection accuracy observed for detectors in the form of a Convolutional Neural Network (CNN) considered in Section VI. Section VII contains the results of the LRT and CNNs — contrasted with the performance of the previous art (RRH) — for a wide range of JPEG quality factors, payloads, and embedding schemes. Section VIII is devoted to an important practical aspect of the JCA, which is its robustness to various JPEG compressors and DCT quantizers, including the "trunc" quantizer in common use today [1], [9]. Since the JCA needs to estimate the quantization table of the original JPEG compression, in Section IX we demonstrate that the table can be accurately estimated from the decompressed cover / stego image while pointing out an important fact that, for the purpose of the JCA, only divisors of quantization steps (the so-called sufficient steps) need to be estimated. The paper is concluded in Section X.

II. Preliminaries

A. Notation

The operation of rounding $x \in \mathbb{R}$ to the nearest multiple of a positive integer q is denoted by $[x]_q \triangleq q \cdot [x/q]$, where the square bracket is the operation of integer rounding $[x]_1 = [x]$. The quantization (rounding) error is defined as $\mathrm{err}_q(x) \triangleq x - [x]_q$. Rounding x "towards zero" is denoted as $\mathrm{trunc}(x)$ and is defined as $\mathrm{trunc}(x) = \lfloor x \rfloor$ for $x \geq 0$ and $\mathrm{trunc}(x) = \lceil x \rceil$ for $x < 0$, where $\lfloor x \rfloor$ and $\lceil x \rceil$ represent flooring and ceiling. Clipping x to a finite dynamic range $[0, 255]$ is denoted $\mathrm{clip}(x)$ with $\mathrm{clip}(x) = x$ for $x \in [0, 255]$, $\mathrm{clip}(x) = 0$ for $x < 0$ and $\mathrm{clip}(x) = 255$ for $x > 255$. The symbol $\triangleq$ is used whenever a new concept is defined. The uniform distribution on the interval $[a, b]$ will be denoted $\mathcal{U}[a, b]$ while $\mathcal{N}(\mu, \sigma^2)$ is used for the Gaussian distribution with mean μ and variance σ^2. If X is a random variable, then f_X, $\mathbb{E}[X]$, and $\mathrm{Var}[X]$ denote the probability density (PDF), expectation, and variance of X, respectively.

Boldface symbols are reserved for matrices and vectors. The symbols $'\odot'$ and $'\oslash'$ denote element-wise product and division between vectors / matrices of the same dimensions. For readability, we slightly abuse notion when referring to the (element-wise) matrix extensions of the above operations. For example, rounding $\mathbf{x} \in \mathbb{R}^{m \times n}$ w.r.t. a matrix $\mathbf{q}$ is defined by $[\mathbf{x}]_{\mathbf{q}} \triangleq \mathbf{q} \odot [\mathbf{x} \oslash \mathbf{q}]$ where $[\cdot]$ denotes element-wise integer rounding in this context. Similarly, we define $\mathrm{err}_{\mathbf{q}}(\mathbf{x}) \triangleq \mathbf{x} - [\mathbf{x}]_{\mathbf{q}}$.

B. Directional statistics

Here, we recall some results from directional statistics needed for the JCA in this paper. For any real-valued random variable X and positive integer q, the distribution of the quantization error $\mathrm{err}_q(X)$ is obtained by wrapping the distribution of X onto a circle with circumference q. In other words, $\mathrm{err}_q(X)$ has a *wrapped* PDF of the form $\sum_{n \in \mathbb{Z}} f_X(x + qn)$ with a support confined to the half-open interval $[-q/2, q/2)$. In the case $X \sim \mathcal{N}(\mu, \sigma^2)$, the quantization error $\mathrm{err}_q(X)$ follows a wrapped Gaussian distribution $\mathcal{N}_{\mathcal{W}}(\mu, \sigma^2, q)$ whose PDF is given by

$$g(x; \mu, \sigma^2, q) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{n \in \mathbb{Z}} \exp\left(-\frac{(x - \mu + qn)^2}{2\sigma^2}\right), \quad (1)$$

when $-q/2 \leq x < q/2$ and $g(x; \mu, \sigma^2, q) = 0$ otherwise. We note that the wrapped Gaussian is equivalent to what was called a *folded* Gaussian in [8] and [10]. However, since the class of wrapped distributions is well studied and is the standard nomenclature in directional statistics [28], we use the term *wrapped* hereafter.

The wrapped Gaussian is adequately approximated by the truncated sum over the $2N + 1$ terms for which $n \in \{0, \pm 1, \ldots, \pm N\}$; the choice of N depends on μ, σ^2, q and the desired precision [28]. For example, $g(x; 0, 1/12, q)$ is well-approximated by one term ($n = 0$) for $q \geq 2$ and three terms ($n = -1, 0, 1$) for $q = 1$. General bounds for the approximation error are found in [8], [23], [28].

Finally, we recall a fundamental asymptotic result known as Poincaré's Limit Theorem (PLT) [28]. If X is an absolutely continuous random variable and q is fixed, then the distribution of $\mathrm{err}_q(cX)$ tends to the uniform distribution $\mathcal{U}[-q/2, q/2)$ as $c \to \infty$. The following extension of the PLT is developed in [18] for wrapping a joint distribution onto a torus. Let M be a n-torus, that is the set $\prod_{i=1}^{n} [-q_i/2, q_i/2)$ where $\prod$ denotes the cartesian product and $\mathbf{q} \in \mathbb{R}^n$. We can wrap $\mathbb{R}^n$ onto M via the map $\mathrm{err}_{\mathbf{q}}$. If X is an absolutely continuous random vector on $\mathbb{R}^n$, then the distribution of $\mathrm{err}_{\mathbf{q}}(cX)$ tends to the uniform distribution on M as $c \to \infty$.

III. Pipeline

In this section, we introduce the pipeline through which an originally uncompressed (raw) image is JPEG compressed and then decompressed for spatial domain embedding, and possibly embedded with a secret message. For clarity, all objects included in this initial compression-decompression will be denoted with a superscript $'(0)'$. JPEG compression proceeds by dividing the image into 8×8 blocks, applying the DCT to each block, dividing the DCT coefficients by quantization steps, and rounding to integers. The coefficients are then arranged in a zig-zag fashion and losslessly compressed to be written as a bitstream into the JPEG file together with a header. In this paper, we constrain ourselves to grayscale images. More details about the JPEG format can be found in [31].

The original uncompressed 8-bit grayscale image with $N_1 \times N_2$ pixels is an element of $\{0, 1, \ldots, 255\}^{N_1 \times N_2}$. Throughout this paper, $\mathbf{x}^{(0)} = (x_{ij}^{(0)})$ denotes one specific 8×8 block of uncompressed pixels where $0 \leq i, j \leq 7$. For clarity, we strictly use i, j to index pixels and k, l to index DCT coefficients.

During JPEG compression, the block of DCT coefficients before quantization, $\mathbf{y}^{(0)} \in \mathbb{R}^{8 \times 8}$, is obtained

[1] Research on quantization noise during recompression with high quality factors is potentially relevant to the forensics community [26], [30].

using the formula $y_{kl}^{(0)} = \mathrm{DCT}_{kl}(\mathbf{x}^{(0)}) \triangleq \sum_{i,j=0}^{7} f_{kl}^{ij} x_{ij}^{(0)}$, $0 \le k,l \le 7$, where

$$f_{kl}^{ij} = \frac{w_k w_l}{4} \cos\frac{\pi k(2i+1)}{16} \cos\frac{\pi l(2j+1)}{16}, \tag{2}$$

are the discrete cosines and $w_0 = 1/\sqrt{2}$, $w_k = 1$ for $0 < k \le 7$. The pair (k,l) is called the kl^{th} DCT mode. Before applying the DCT, each pixel is adjusted by subtracting 128 from it during JPEG compression, a step we omit here since it has no effect on our analysis. For brevity, we will also use matrix notation and denote the DCT of a block $\mathbf{u}$ as $\mathbf{v} = \mathbf{D}\mathbf{u}$ where $v_{kl} = \mathrm{DCT}_{kl}(\mathbf{u})$ for all k,l. Here, $\mathbf{D}$ is a 64×64 matrix of discrete cosines and $\mathbf{u}$, $\mathbf{v}$ are the blocks rearranged as column vectors. Note that $\mathbf{D}^{\top} = \mathbf{D}^{-1}$ due to orthonormality.

The block of quantized DCTs is $\mathbf{c}^{(0)} = [\mathbf{y}^{(0)} \oslash \mathbf{q}]$, $c_{kl}^{(0)} \in \{-1024, \dots, 1023\}$ where $\mathbf{q} = (q_{kl})$ is a luminance quantization matrix of quantization steps q_{kl} supplied in the header of the JPEG file. For a JPEG compressor that uses truncation instead of rounding, $\mathbf{c}^{(0)} = \mathrm{trunc}(\mathbf{y}^{(0)} \oslash \mathbf{q})$.

During decompression, the above steps are reversed. First, dequantizing $\mathbf{c}^{(0)}$ yields $\widetilde{\mathbf{y}}^{(0)} = \mathbf{q} \odot \mathbf{c}^{(0)}$. Applying the inverse DCT, the block $\widetilde{\mathbf{x}}^{(0)}$ of non-rounded pixels after decompression is obtained by $\widetilde{x}_{ij}^{(0)} = \mathrm{DCT}_{ij}^{-1}(\widetilde{\mathbf{y}}^{(0)}) \triangleq \sum_{k,l=0}^{7} f_{kl}^{ij} \widetilde{y}_{kl}^{(0)}$, where $\widetilde{x}_{ij}^{(0)} \in \mathbb{R}$, or in the matrix form $\widetilde{\mathbf{x}}^{(0)} = \mathbf{D}^{\top}\widetilde{\mathbf{y}}^{(0)}$. The pair (i,j) used to index $\widetilde{x}_{ij}^{(0)}$ is called the ij^{th} JPEG phase [14]. Finally, rounding $\widetilde{\mathbf{x}}^{(0)}$ to integers and clipping to a finite dynamic range $[0, 255]$ produces the fully decompressed block $\mathbf{x} = (x_{ij})$.

At this point, the steganographer may embed the cover image $\mathbf{x}$ with a secret message by introducing embedding changes $\boldsymbol{\eta}$ to produce the stego image $\mathbf{x}^{(s)} = \mathbf{x} + \boldsymbol{\eta}$. In the JCA, the (cover or stego) image is again JPEG compressed and decompressed to obtain a reference image. Since $\mathbf{q}$ is not available in a decompressed JPEG's file format, recompression is performed using a quantization matrix, $\widehat{\mathbf{q}}$, estimated directly from $\mathbf{x}$ or $\mathbf{x}^{(s)}$.

Figure 1 visually conveys the JCA pipeline considered in this paper. As shown, the recompressed blocks $\mathbf{y}$, $\widetilde{\mathbf{y}}$, $\widetilde{\mathbf{x}}$ are all defined by repeating the compression process. We omit $\mathbf{c}^{(0)}$ and $\mathbf{c}$ from Figure 1 since the operation $[\cdot]_{\mathbf{q}}$ combines quantizing and dequantizing into one step. All stego versions of the objects considered in the recompression will be denoted with a superscript $'(s)'$ — the cover versions do not have a superscript.

Moreover, we denote the initial quantization error by $\boldsymbol{\varepsilon}^{(0)} \triangleq \mathbf{y}^{(0)} - \widetilde{\mathbf{y}}^{(0)}$, the decompression (rounding) error in the spatial domain by $\boldsymbol{\delta} \triangleq \widetilde{\mathbf{x}}^{(0)} - \mathbf{x}$, and the recompression quantization error by $\boldsymbol{\varepsilon} \triangleq \mathbf{y} - \widetilde{\mathbf{y}}$. For brevity, we often refer to $\boldsymbol{\varepsilon}$ as the *Q error* and $\mathbf{D}^{-1}\boldsymbol{\varepsilon}$ as the spatial domain Q error, or *SQ error*. We refer to $\mathrm{clip}([\widetilde{\mathbf{x}}]) - \mathbf{x}$ as the *recompression residual* which was the object of focus in the previous art [22]. Ignoring clipping, the (negative) SQ error can be seen as the unrounded recompression residual since $\mathbf{x}$ is a block of integers:

$$[-\mathbf{D}^{-1}\boldsymbol{\varepsilon}] = [\widetilde{\mathbf{x}} - \mathbf{x}] = [\widetilde{\mathbf{x}}] - \mathbf{x}. \tag{3}$$

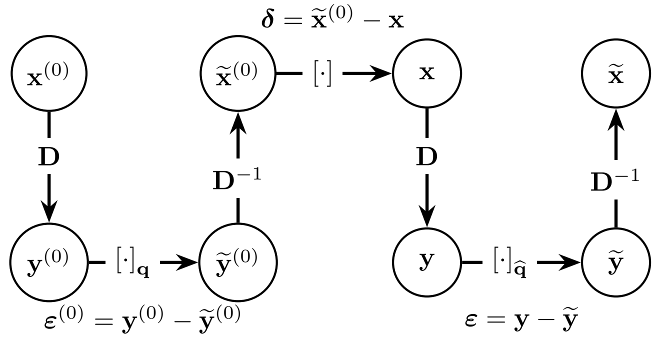
$\boldsymbol{\delta} = \widetilde{\mathbf{x}}^{(0)} - \mathbf{x}$
$\mathbf{x}^{(0)}$
$\widetilde{\mathbf{x}}^{(0)}$
$[\cdot]$
$\mathbf{x}$
$\widetilde{\mathbf{x}}$
$\mathbf{D}$
$\mathbf{D}^{-1}$
$\mathbf{D}$
$\mathbf{D}^{-1}$
$\mathbf{y}^{(0)}$
$[\cdot]_{\mathbf{q}}$
$\widetilde{\mathbf{y}}^{(0)}$
$\mathbf{y}$
$[\cdot]_{\widehat{\mathbf{q}}}$
$\widetilde{\mathbf{y}}$
$\boldsymbol{\varepsilon}^{(0)} = \mathbf{y}^{(0)} - \widetilde{\mathbf{y}}^{(0)}$
$\boldsymbol{\varepsilon} = \mathbf{y} - \widetilde{\mathbf{y}}$

Figure 1. JPEG compression - decompression - recompression pipeline. Adjusting pixels to $[-128, 127]$ and clipping to $[0, 255]$ are ignored.

IV. Pipeline analysis

Equipped with the tools introduced in Section II-B, we can now study the objects in Figure 1. We start by modeling the initial quantization error, $\boldsymbol{\varepsilon}^{(0)}$, as a random vector. We then derive the distributions of subsequent objects, ultimately formulating how a steganographic embedding impacts the distribution of the Q errors $\boldsymbol{\varepsilon}$. In summary, the analysis in this section will leverage these facts:

1) The (cover or stego) image is stored using integers which allows us to analytically isolate the rounding errors in each domain.
2) The dimensionality of the blocks is high enough to use the Central Limit Theorem (CLT) to approximate the marginals using Gaussians when switching between domains.
3) Poincaré's theorem tells us the distribution of $\boldsymbol{\delta}$ tends to a uniform distribution with jointly independent components as quality factor decreases.

A. Rounding errors in the spatial domain

By the linearity of the DCT, we can express the non-rounded block of pixels $\widetilde{\mathbf{x}}^{(0)}$ as

$$\begin{aligned} \widetilde{\mathbf{x}}^{(0)} &= \mathbf{D}^{-1}\widetilde{\mathbf{y}}^{(0)} \\ &= \mathbf{D}^{-1}\mathbf{y}^{(0)} - \mathbf{D}^{-1}\boldsymbol{\varepsilon}^{(0)} \\ &= \mathbf{x}^{(0)} - \mathbf{D}^{-1}\boldsymbol{\varepsilon}^{(0)}. \end{aligned} \tag{4}$$

Consider the case of the round quantizer; the values of $\varepsilon_{kl}^{(0)}$ are contained within $[-q_{kl}/2, q_{kl}/2)$.

Assumption 1. *For all modes (k,l), the DCT quantization errors $\varepsilon_{kl}^{(0)}$ are jointly independent and satisfy*

$$\varepsilon_{kl}^{(0)} \sim \mathcal{U}[-q_{kl}/2, q_{kl}/2). \tag{5}$$

Assumption 1 has been studied in [34], used in [8], [10], [30], and can be justified directly by the Poincaré Theorem for small quantization steps q_{kl}. By the joint independence of $\varepsilon_{kl}^{(0)}$ and the fact that $\mathbb{E}[\varepsilon_{kl}^{(0)}] = 0$ and $\mathrm{Var}[\varepsilon_{kl}^{(0)}] = q_{kl}^2/12$, Lindeberg's extension of the CLT implies that the marginals of $\widetilde{\mathbf{x}}^{(0)}$ approximately follow the Gaussian distribution

$$\widetilde{x}_{ij}^{(0)} \sim \mathcal{N}(x_{ij}^{(0)}, s_{ij}^{(0)}), \tag{6}$$

with variance

$$s_{ij}^{(0)} = \frac{1}{12} \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 q_{kl}^2. \tag{7}$$

The rounding error in the spatial domain has the form

$$\boldsymbol{\delta} = \widetilde{\mathbf{x}}^{(0)} - [\widetilde{\mathbf{x}}^{(0)}] = \mathrm{err}_1(-\mathbf{D}^{-1}\boldsymbol{\varepsilon}^{(0)}), \tag{8}$$

because $\mathbf{x}^{(0)}$ is a block of integers. We conclude that the marginals of $\boldsymbol{\delta}$ are approximately distributed by $\delta_{ij} \sim \mathcal{N}_{\mathcal{W}}(0, s_{ij}^{(0)}, 1)$ for all JPEG phases.

We note that when quantization steps are large or when an alternate quantizer such as trunc is used, Assumption 1 may no longer hold. Nonetheless, the PLT still allows us to say something about the joint distribution of the rounding errors $\boldsymbol{\delta}$. Looking at Eq. (7), notice that the probability mass of $\boldsymbol{\varepsilon}^{(0)}$ spreads out as the entries of $\mathbf{q}$ increase. Thus, the distribution of $\boldsymbol{\delta}$ is well-approximated by the joint uniform distribution on $[-1/2, 1/2)^{64}$ for sufficiently low enough quality factors. We experimentally observed that the marginals δ_{ij} are uniform for QFs 98 and below, and thus, we infer that the PLT has applied for these qualities.

Note that if the quantizer is trunc, the variance $\mathrm{Var}[\varepsilon_{kl}^{(0)}]$ is larger compared to round regardless of the distribution of uncompressed DCT coefficients $\mathbf{y}^{(0)}$. Hence, we also conclude that the PLT has applied for QFs 98 and below in the case of trunc.

B. Cover images

By reasoning similar to that of Eq. (4), the linearity of the DCT implies

$$\mathbf{y} = \widetilde{\mathbf{y}}^{(0)} - \mathbf{D}\boldsymbol{\delta}. \tag{9}$$

Assumption 2. *The cover block $\mathbf{x} = [\widetilde{\mathbf{x}}^{(0)}]$ has rounded to pixels all within the dynamic range $[0, 255]$. The rounding errors $\boldsymbol{\delta}$ are jointly independent for all JPEG qualities.*

If $\widetilde{x}_{ij}^{(0)}$ is outside the dynamic range, δ_{ij} will belong to an interval potentially much larger than $[-1/2, 1/2)$ with bounds dependent on image content. Using Assumption 2, we may ignore the effects of clipping and approximate the marginals of $\mathbf{y}$ using the CLT:

$$y_{kl} \sim \mathcal{N}(\widetilde{y}_{kl}^{(0)}, s_{kl}), \tag{10}$$

$$s_{kl} = \sum_{i,j=0}^{7} (f_{kl}^{ij})^2 \mathrm{Var}[\delta_{ij}]. \tag{11}$$

Note that for QFs 98 and below, the approximate uniformity of $\boldsymbol{\delta}$ implies $\mathrm{Var}[\delta_{ij}] \approx 1/12$, which yields $s_{kl} \approx 1/12$ by the orthonormality of the DCT. The Q error computed via the true quantization matrix $\mathbf{q}$ can be expressed as

$$\boldsymbol{\varepsilon} = \mathbf{y} - [\mathbf{y}]_{\mathbf{q}} = \mathrm{err}_{\mathbf{q}}(-\mathbf{D}\boldsymbol{\delta}), \tag{12}$$

since $\widetilde{y}_{kl}^{(0)}$ is an integer multiple of q_{kl} for all (k, l). Thus, we conclude that $\varepsilon_{kl} \sim \mathcal{N}_{\mathcal{W}}(0, s_{kl}, q_{kl})$.

C. Stego images

We model the embedding changes η_{ij} as content-adaptive ±1 noise in the spatial domain; we have $\mathbf{x}^{(s)} = \mathbf{x} + \boldsymbol{\eta}$. Specifically, we treat η_{ij} as a random variable supported on $\{-1, 0, 1\}$ with PMF $\mathbb{P}(\eta_{ij} = 1) = \mathbb{P}(\eta_{ij} = -1) = \beta_{ij}$, where β_{ij} are known as the *change rates* (or *selection channel*) determined by the stego scheme. Under this framework, the non-rounded recompressed DCTs have the form

$$\mathbf{y}^{(s)} = \widetilde{\mathbf{y}}^{(0)} - \mathbf{D}\boldsymbol{\delta} + \mathbf{D}\boldsymbol{\eta}. \tag{13}$$

Assumption 3. *The embedding changes η_{ij} are jointly independent and independent of the rounding errors δ_{ij}.*

This is a reasonable assumption for steganography that minimizes an additive distortion and does not use the rounding errors as side-information for embedding. Applying the CLT again, we have

$$y_{kl}^{(s)} \sim \mathcal{N}(\widetilde{y}_{kl}^{(0)}, s_{kl} + r_{kl}), \tag{14}$$

$$r_{kl} = \sum_{i,j=0}^{7} (f_{kl}^{ij})^2 \mathrm{Var}[\eta_{ij}]. \tag{15}$$

Thus, the Q error for a stego block can be written as

$$\boldsymbol{\varepsilon}^{(s)} = \mathrm{err}_{\mathbf{q}}(-\mathbf{D}\boldsymbol{\delta} + \mathbf{D}\boldsymbol{\eta}), \tag{16}$$

since $\widetilde{y}_{kl}^{(0)}$ is an integer multiple of q_{kl} for all modes. Hence, $\varepsilon_{kl}^{(s)} \sim \mathcal{N}_{\mathcal{W}}(0, s_{kl} + r_{kl}, q_{kl})$ which means the embedding increases the variance of the wrapped Gaussian.

V. Statistical Hypothesis Detector

The analysis carried out in the previous section allows us to formulate a statistical hypothesis test about the Q errors for detecting steganography. Then, we introduce rules for eliminating blocks from the test for a tighter fit of modeling assumptions in practice, which improves the detection accuracy. Afterwards, we briefly discuss other considerations for modeling assumptions. The analysis of this section is useful to obtain insight into why and how the JCA works and to explain trends observed for other types of detectors studied in Section VI.

All experiments in this section, and in this paper in general, were conducted on the union of the BOSSbase 1.01 [2] and BOWS2 [3] datasets, each with 10,000 grayscale images resized to 256 × 256 pixels with `imresize` in Matlab using default parameters. We refer to the union as BOSSBOWS2. This dataset is a popular choice for designing detectors with deep learning because small images are more suitable for training deep architectures [6], [36]–[39], [41]. The training set (TRN) contained all 10,000 BOWS2 images along with 4,000 randomly selected images from BOSSbase. The remaining images from BOSSbase were randomly partitioned to create the validation set (VAL) and the testing set (TST) containing 1,000 and 5,000 images, respectively.

A. Likelihood ratio test

Given a collection $\mathcal{B}$ of 8×8 blocks from an $N_1 \times N_2$ decompressed image, the steganalyst is faced with the following hypothesis test for all $0 \leq k, l \leq 7$ across all blocks $\mathbf{x} \in \mathcal{B}$:

$$\mathcal{H}_0 : \varepsilon_{kl} \sim \mathcal{N}_{\mathcal{W}}(0, s_{kl}, q_{kl}) \tag{17}$$

$$\mathcal{H}_1 : \varepsilon_{kl} \sim \mathcal{N}_{\mathcal{W}}(0, s_{kl} + r_{kl}, q_{kl}),\ r_{kl} > 0. \tag{18}$$

Assumption 4. *The Q errors ε_{kl} are jointly independent within and between blocks.*

This assumption allows us to construct a detector from the marginals; working with a joint density leads to similar computational complexity issues encountered in [10], [11]. Thus, the log-likelihood ratio test for an image is

$$\mathcal{L}(\mathcal{B}) = \sum_{\mathbf{x}\in\mathcal{B}} \sum_{k,l=0}^{7} \mathcal{L}_{kl}(\mathbf{x}) \tag{19}$$

$$= \sum_{\mathbf{x}\in\mathcal{B}} \sum_{k,l=0}^{7} \log \frac{g(\varepsilon_{kl}; 0, s_{kl} + r_{kl}, \widehat{q}_{kl})}{g(\varepsilon_{kl}; 0, s_{kl}, \widehat{q}_{kl})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma. \tag{20}$$

Assuming the change rates (and thus r_{kl}) are known, the steganalyst is faced with a simple hypothesis, for which the LRT is uniformly most powerful in the clairvoyant case according to the NP-lemma [19]. As a remark, we remind the reader that the quantization matrix must be estimated from the image first — a preanalytical step discussed in Section IX. Until then, we assume the true quantization matrix is known, i.e. $\widehat{\mathbf{q}} = \mathbf{q}$.

Moreover, the LRT is composite if r_{kl} is unknown, which would be the case when detecting multiple steganographic methods, an unknown payload size, or a steganographic method with unknown or partially known selection channel, e.g. side-informed steganography [12], [15], [17] or methods with synchronized embedding changes [7], [16], [25]. On the other hand, for detecting a known steganography and a known payload size, the selection channel is approximately available — the change rates β_{ij} can be computed from the analyzed stego image — which means that r_{kl} can also be approximately computed. By the Lindeberg's extension of the CLT, the normalized LRT

$$\Lambda(\mathcal{B}) = \frac{\mathcal{L}(\mathcal{B}) - \mathbb{E}_{\mathcal{H}_0}[\mathcal{L}(\mathcal{B})]}{\sqrt{\mathrm{Var}_{\mathcal{H}_0}[\mathcal{L}(\mathcal{B})]}} \tag{21}$$

follows the distribution $\mathcal{N}(0, 1)$ under $\mathcal{H}_0$, which allows setting a decision threshold for the normalized LRT that achieves the largest detection power for a fixed false-alarm probability. Figure 2 shows the distribution of $\Lambda(\mathcal{B})$ under $\mathcal{H}_0$ across images from the training and validation sets when ε_{kl} are sampled from their distributions (17).

B. Block elimination

In practice, blocks should be eliminated from hypothesis testing if they do not adhere to at least one of the assumptions above; there is no guarantee that the conclusions apply to such blocks. To this end, we formulate rules for rejecting a block $\mathbf{x}$ from $\mathcal{B}$ based on the following common phenomena.

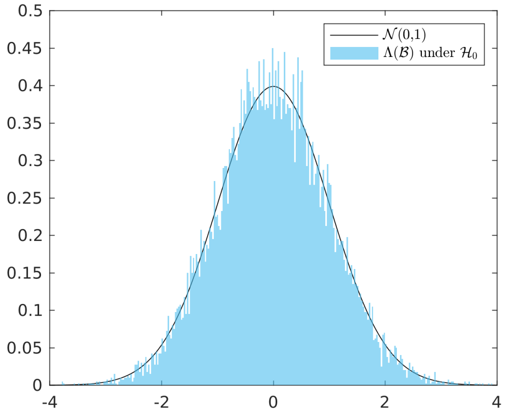

Figure 2. Distribution of LRT $\Lambda(\mathcal{B})$ under $\mathcal{H}_0$ for Monte-Carlo sampled ε_{kl} with s_{kl} and r_{kl} computed from images in the union of TRN and VAL. One sample of ε_{kl} was taken per DCT mode per block.

1) Block saturation: A block $\mathbf{x}$ with pixel values x_{ij} is *saturated* if there exists a phase (i, j) such that $x_{ij} = 0, 1, 254,$ or 255.
2) Block sparsity: A block $\mathbf{x}$ is *sparse* if the number of zero DCT coefficients in $\mathbf{y}$ is larger than or equal to 8. To account for floating-point error in the DCT, a coefficient y_{kl} is considered "zero" if $|y_{kl}| < 10^{-5}$.

Saturated blocks potentially violate Assumption 2 due to clipping. We include pixel values 1 and 254 to account for the possibility of embedding into pixels at the boundary of the dynamic range. As for sparse blocks, having 8 or more zero DCTs concentrate around zero is highly unlikely since the y_{kl} are Gaussian random variables.[2] Hence, we conclude that the CLT fails for sparse blocks. Therefore, if a block is deemed *saturated* or *sparse* (or both), then the block is rejected. Throughout the paper, all experiments with block elimination abide by this criteria.

We note that content-adaptive schemes tend to embed in non-saturated and non-sparse blocks. Thus, block elimination may artificially increase the image's overall change-rate which is to the steganalyst's benefit. On the other hand, we do not foresee steganographers intentionally embedding in rejected blocks since doing so would be highly detectable by methods outside the JCA and methods we introduce later in Section VI.

The BOSSBOWS2 dataset contains a small number of images (depending on JPEG quality) whose blocks were all eliminated due to lack of content. In our experiments, we eliminated these singular images entirely since they are known to be bad covers.

[2] The authors observed that zero DCTs typically occur in entire rows or columns of modes which is why the sparse block threshold was chosen to be 8.

C. Other considerations

We also experimented with modeling the marginals of the uncompressed DCT coefficients $\mathbf{y}^{(0)}$ as generalized Gaussian [29] random variables. It follows that the quantization error $\boldsymbol{\varepsilon}^{(0)}$ would be wrapped generalized Gaussian (WGG) distributed, and the phase-dependent variances, s_{kl}, would be computed by numerically integrating the WGG. In practice, though, the shape and width parameters would need to be estimated from $\mathbf{y}$ / $\mathbf{y}^{(s)}$, the unrounded DCTs of the cover / stego image, which complicates matters. However, even when estimating the parameters directly from the uncompressed image, we did not see the LRT benefit. Also, we noticed that the number of terms needed to approximate the WGG becomes unwieldy if the shape and width parameters are too small.

VI. Machine Learning Detectors

The LRT detector discussed above was derived in the DCT domain under the assumption that the distributions of different 8×8 blocks are independent. The embedding changes are, however, performed in the spatial domain, and the steganalyst can and should make use of dependencies between pixels across the block boundaries, which is ignored by the LRT test. Moreover, the heuristic block rejection rules were adopted based on experiments and are likely an additional source of suboptimality as the modeling assumptions, such as the validity of the CLT, will generally depend on the block content as well as the quality factor. Thus, the authors anticipate Convolutional Neural Network (CNN) detectors will provide better detection performance especially when supplying the image under investigation as one of the channels on top of the Q / SQ error during training. Such detectors could also potentially be more robust to differences between JPEG compressors simply by enlarging the training set. They can also more easily be made universal in the sense of covering both round and trunc DCT quantizers and possibly trained for unknown payloads.

These advantages motivated the authors to study deep learning based detectors. All previous art made use of the recompression residual $\mathrm{clip}([\tilde{\mathbf{x}}]) - \mathbf{x}$ as a reference signal, because recompressing the image and then decompressing to the spatial domain essentially erases the embedding changes for lower quality factors. For detecting content-adaptive stego schemes, however, the original image should be used as input so the network can properly learn the selection channel and form better detection statistics from dependencies between neighboring pixels.

Section VI-A and Section VI-B introduce the experimental setup for SRNet and the prior art, respectively.

A. SRNet

In this paper, we report the results for three flavors of SRNet [6]: an SRNet trained only on Q errors (Q-SRNet), on SQ errors (SQ-SRNet), and on two channels (SQY-SRNet) — the normalized image $\mathbf{x}/255$ (Y channel) and the SQ error — which provided by far the best overall performance especially for high quality factors. We also investigated an SRNet trained on both the image and its recompression residual but found that it performed worse than the LRT for high QFs. We hypothesize the recompression residual loses information about the embedding after rounding / clipping in the spatial domain.

Training was done for 50 epochs using mini-batches of size 64, the adamax optimizer [21], the one-cycle learning-rate (LR) scheduler with maximum LR 1×10^{-3} [33], and the cross-entropy loss function. All classifiers were trained using a pair-constraint, requiring batches to contain cover-stego pairs.

To augment the training data, a random dihedral group (D4) operation was applied to each cover-stego pair in the batch before extracting Q / SQ errors. Observe that the quantization table must be transposed when images are rotated by 90 or 270 degrees.

In experiments with multiple payloads, we trained networks from scratch on the largest payload with maximum LR 1×10^{-3}. The checkpoint with minimal validation loss was then used as a starting seed for training on smaller payloads with maximum LR 3×10^{-4}. Curriculum training in this manner significantly helped facilitate convergence.

B. RRH

For comparison against the prior art, we also implemented the RRH method [22] (see Section I) trained on the union of the TRN and VAL. The recompression residual was computed using Matlab's `imwrite` and `imread` to match the initial (de)compressor implementation.

VII. Experiments

In this section, our goal is to determine the best detector from Section V and VI. First, we compare the performance of the LRT and the three SRNets w.r.t. JPEG quality for a fixed stego scheme and payload. The best detector of these four will then be rigorously tested against the prior art, RRH, for a variety of stego schemes and payloads. Throughout the section, we present the results through the lens of our analysis in Section IV.

A. Methodology

As in Section V, we used the same split 14,000 / 1,000 / 5,000 for TRN / VAL / TST. Images were initially compressed and decompressed using Matlab's `imwrite` and `imread`. In order to compare the LRT to the machine learning detectors, we first choose the decision threshold that minimized P_{E} on the union of TRN and VAL. The measurement P_{E} is the probability of error under equal priors defined by $P_{\mathrm{E}} = (P_{\mathrm{MD}} + P_{\mathrm{FA}})/2$, where P_{MD} and P_{FA} are the probabilities of missed detection and false alarm. The test accuracy of the LRT is then computed on TST using this fixed threshold. Cover-stego pairs were generated using the MiPOD [32] simulator at 0.01 bits per pixel (bpp).

B. Performance w.r.t quality

In Figure 3, the left plot visualizes the trends for the LRT and all versions of SRNet. Since SQY-SRNet outperformed the other detectors especially for high qualities, we continued by testing SQY-SRNet and the prior art on the following four content-adaptive steganographic schemes: S-UNIWARD [15], HILL [24], MiPOD [32], and WOW [13]. These schemes were tested on the following range of payloads: 0.02, 0.01, 0.005, and 0.002 bpp. We refer the reader to Tables VI and VII in the appendix for the full results for SQY-SRNet and the prior art. A subset of these results are shown in the right plot of Figure 3. The SQY-SRNet significantly outperforms the RRH especially for small payloads for QFs above 93.

Note that the model-based MiPOD is consistently more secure that the other three cost-based stego algorithms. The difference is most pronounced for the smallest payloads and largest qualities. We were able to trace the reason for this to the average number of modified pixels by these four schemes. For QF100 and payload 0.002 bpp, the average number of changed pixels for MiPOD, S-UNIWARD, WOW, and HILL are 9.7, 12.2, 13.8, and 14.1, which matches the trend in increased detectability with SQY-SRNet: 0.689, 0.811, 0.863, and 0.872.

We note that the performance of the LRT matches the performance of Q-SRNet except for QFs 99–100. We interpret this overlap as an indication that our modelling assumptions take into account all relevant information contained in the Q error representation of the image (besides inter-block dependencies). We hypothesize that the deviation for QFs 99–100 occurs due to $\boldsymbol{\delta}$ not being jointly independent since the PLT does not apply for these qualities as per Section IV-A. This implies the CLT may not apply to the marginals of $\mathbf{y}$, hence the ε_{kl} is not guaranteed to follow the wrapped Gaussian in Section IV-B.

We note that SRNet generally has trouble forming inter-block statistics in DCT domain representations [40] which is likely why we see a jump in performance when the SQ error is used instead.

In [11], QF100 is deemed the hardest quality for the JCA due to search complexity. This hints at the existence of suboptimality in the prior art for which QF97 is empirically the hardest quality. Note that SQY-SRNet closely matches the monotonic behavior we intuitively expect.

VIII. Robustness to JPEG Compressors

There exist many variants of JPEG compressors, which can differ in the implementation of the DCT, the quantizer, and the internal number representation. If two compressors differ, they may produce different JPEG images from the same raw image. Similarly, if two decompressors differ, they may produce different decompressed images from the same JPEG file. As a result, a cover image can potentially originate from a vast number of JPEG compressor-decompressor combinations. In addition, the steganalyst must use a JPEG variant for recompression and decompression to compute, e.g. the SQ errors. Any mismatch of JPEG combination may complicate the distribution of rounding errors and potentially dramatically decrease the performance of the JCA. Also, machine-learning detectors may perform poorly on a variant not seen during training. In this section, we pinpoint the JPEG compressor variant that should be used for training in order to maximize the robustness of SQY-SRNet.

Since the recompression method for the JCA is the steganalyst's choice, we are free to select the one that works the best overall. Since rounding errors are not easily attainable using off-the-shelf JPEG compressors, we manually recompress via SciPy's `dct` to compute Q / SQ errors for all experiments. To simplify matters, we exlusively use Matlab's `imwrite` to compute the recompression residual for the prior art [22] since this variant was used for benchmarking in Section VI.

On the other hand, the steganalyst does not know the compressor-decompressor pair used to obtain the spatial representation of the cover image. The following (de)compressor implementations were considered: Matlab's `imwrite/imread`, Python3 library PIL (PIL), ImageMagick's Convert (Convert), Int and Float DCT compressors in libjpeg (version 6b).[3] Fast DCT compression in libjpeg has not been included in our tests because it is not recommended for QFs larger than 97 since the compression is then slower and more lossy than on smaller QFs.[4]

For experimental feasibility, we reduced the number of compressor-decompressor pairs tested by restricting our attention purely to differences between quantizers used for the initial compression. We specifically use Matlab's `imwrite` for its round quantizer and a manually implemented trunc compressor in Python3 using SciPy's `dct`.

A. Mismatching the decompressor

First, we try to determine the best JPEG decompressor for the steganalyst under the assumption 1) that the original JPEG cover was obtained using a round quantizer for the DCTs and 2) the steganographer was free to choose any of the decompressors. Table I shows the testing accuracies for SQY-SRNet trained and tested on mismatched decompressors for QFs 95, 99, 100. While a loss can indeed be observed especially in the case when the detector was built with images generated by 'Float' and 'Convert', the detector trained on images from Python's PIL and Matlab's `imread` generalized overall very well when evaluated on images from all five compressors.

We also studied the prior art's robustness to decompressor since no benchmarking exists in [22]. The testing accuracies for the RRH are shown in Table II. We observed that QF99 and 100 had the same pattern in the results (with accuracies in the range [.7486, .7616] for QF100), so we report the results for QFs 90, 95, 99.

[3] http://libjpeg.sourceforge.net/

[4] Taken from libjpeg documentation https://manpages.ubuntu.com/manpages/artful/man1/cjpeg.1.html.

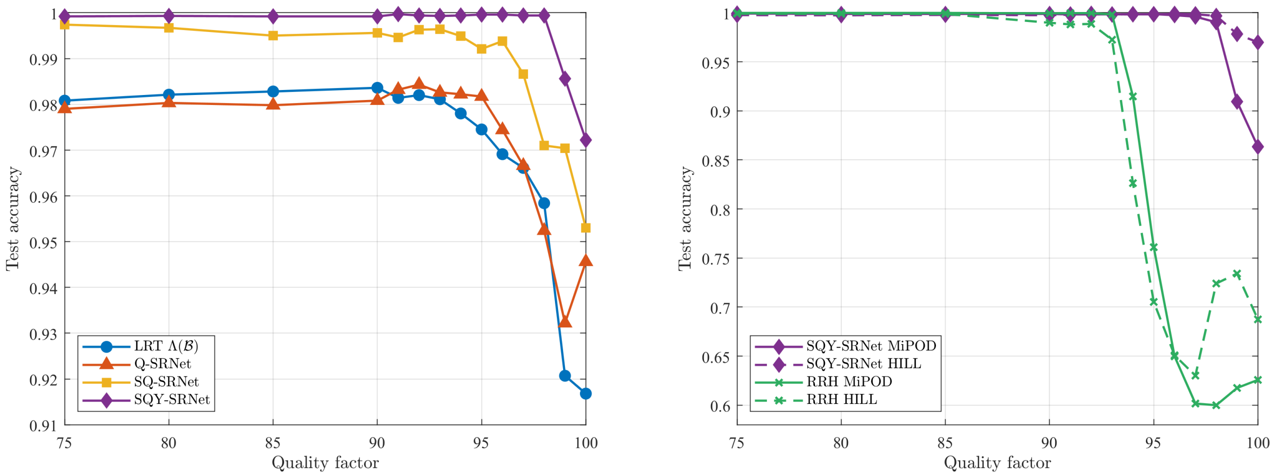

Figure 3. Left: Testing accuracy as a function of JPEG quality for LRT (21) and all flavors of SRNet. Embedded using MiPOD at 0.01 bpp. Right: Testing accuracy as a function of JPEG quality for SQY-SRNet (purple) and RRH (green). Embedded using MiPOD (solid) and HILL (dashed) at 0.005 bpp.

Table I
Testing accuracy for SQY-SRNet trained and tested on combinations of decompressors. Each row / column corresponds to the decompressor used for training / testing, respectively. Initially compressed with Matlab's `imwrite`. Embedded using MiPOD at 0.01 bpp.

QF	TRN	TST decompressor				
	decomp.	imread	float	int	convert	PIL
100	imread	.9721	.9556	.9716	.9568	.9729
	float	.9500	.9742	.9491	.9739	.9491
	int	.9695	.9587	.9682	.9570	.9685
	convert	.9461	.9732	.9455	.9742	.9456
	PIL	.9721	.9633	.9706	.9635	.9708
99	imread	.9856	.9846	.9870	.9849	.9859
	float	.9781	.9875	.9784	.9899	.9777
	int	.9845	.9833	.9856	.9832	.9838
	convert	.9760	.9878	.9770	.9885	.9771
	PIL	.9843	.9864	.9849	.9860	.9844
95	imread	.9996	.9997	.9993	.9994	.9996
	float	.9991	.9992	.9991	.9992	.9992
	int	.9996	.9996	.9993	.9992	.9995
	convert	.9996	.9994	.9993	.9993	.9996
	PIL	.9994	.9995	.9994	.9992	.9995

Table II
Testing accuracy for RRH trained and tested on combinations of decompressors. Matlab's `imwrite` is used for the initial compressor and used to compute the recompression residual. Embedded using MiPOD at 0.01 bpp.

QF	TRN	TST decompressor				
	decomp.	imread	float	int	convert	PIL
99	imread	.7538	.7315	.7523	.7341	.7522
	float	.7481	.7453	.7451	.7485	.7437
	int	.7552	.7323	.7518	.7342	.7512
	convert	.7486	.7446	.7460	.7480	.7448
	PIL	.7540	.7339	.7518	.7363	.7517
95	imread	.9042	.5000	.9031	.5000	.9041
	float	.5288	.6813	.5281	.6828	.5281
	int	.9035	.5000	.9032	.5000	.9041
	convert	.5166	.6834	.5159	.6834	.5172
	PIL	.9022	.5000	.9019	.5000	.9029
90	imread	.9993	.5000	.9993	.5000	.9993
	float	.4819	.6349	.4816	.6359	.4817
	int	.9993	.5000	.9992	.5000	.9993
	convert	.4831	.6350	.4828	.6350	.4823
	PIL	.9993	.5000	.9994	.5000	.9993

The recompression residual will typically contain blocks with no pixel changes or blocks with large patterns of changes; residual blocks will rarely contain single pixel changes especially for QFs with no 1's in the quantization table [22]. Thus, for QF92 and below, embedding is highly detectable since single pixel changes will appear in the recompression residual. We observed, however, that having mismatched JPEG variants in the JCA pipeline commonly creates salt-and-pepper noise artifacts in the recompression residual, which the RRH misinterprets as steganography. For example, the accuracy of RRH for QF90 trained and tested on the float decompressor only has an accuracy of .6349 because the compressor is Matlab's `imwrite`. For QFs above 92, mismatching is less problematic since the RRH classifier gets trained on covers that more commonly produce salt-and-pepper noise.

B. Mismatching the quantizer

Having seen that training on MATLAB's `imread` or PIL generalize the best for decompressor robustness, we turn to investigating robustness to a compressor's quantizer. Table III shows that training on either the `imread` or PIL decompressor gives similar accuracies when images are initially compressed with a trunc quantizer. Overall, the accuracies are somewhat lower compared to when quantized with round (see Table I) with the largest difference for QF100. This is related to the differences between both quantizers, namely the way they affect the distribution of δ_{ij}. Except for QFs 99–100, $\boldsymbol{\delta}$ is well-approximated by the uniform distribution for both quantizers (see Section IV-A). Therefore, the SQ errors for both quantizers approximately follow the same distribution under assumptions of Section IV which explains the matching accuracies for QF95. For QFs 99–100, however, the DCT quantization errors for the trunc quantizer $\varepsilon_{kl}^{(0)} \in [0, q_{kl})$ for positive

Table III
Testing accuracy for SQY-SRNet trained and tested on the trunc quantizer and combinations of decompressors. Embedded using MiPOD at 0.01 bpp.

QF	TRN	TST decompressor				
	decomp.	imread	float	int	convert	PIL
100	imread	.8894	.8641	.8918	.8664	.8897
	PIL	.8906	.8608	.8940	.8642	.8923
99	imread	.9830	.9775	.9830	.9770	.9834
	PIL	.9842	.9777	.9824	.9767	.9833
95	imread	.9993	.9993	.9992	.9993	.9996
	PIL	.9994	.9994	.9996	.9994	.9996

DCTs and $\varepsilon_{kl}^{(0)} \in (-q_{kl}, 0]$ for negative DCTs. Thus, any asymmetry in the distribution of the DCT coefficients in the cover image transfers to an asymmetry of the quantization errors, giving them a non-zero mean. In contrast, the distribution of quantization errors for the round quantizer is much less affected by such asymmetries.[5] Consequently, the rounding errors δ_{ij} in the spatial domain for the trunc quantizer are wrapped Gaussians with non-zero means, which has an effect on the accuracy of the LRT (not shown in this paper) and, apparently, also on the CNN detectors.

Next, we investigate what happens when there is a mismatch between the quantizer used to obtain the original cover JPEG and the quantizer used by the steganalyst for training their detectors. In Table IV, SQY-SRNet exhibits no loss of accuracy for mismatched quantizers at QF95, a noticeable loss for QF99, and a catastrophic loss for QF100. As explained in the paragraph above, this demonstrates the utilility of the PLT when steganalyzing (lower quality) images compressed with quantizers not seen during training. For QFs 99–100, however, the distribution of $\boldsymbol{\delta}$ is quantizer-dependent, which implies the SQ errors are quantizer-dependent.

Since the JPEG quantizers can be distinguished quite accurately with machine-learning tools, we decided to address the performance loss simply by training on images obtained using both quantizers. As Table V portrays, training in this fashion resolves the problem with an unknown quantizer; the detection accuracies are now comparable to those of the detectors trained and tested on images obtained with matching quantizers (as shown in Tables I and III). Overall, training on the `imread` decompressor generalizes slightly better than training on PIL.

IX. Estimating the Quantization Table

As mentioned earlier, the steganalyst must estimate the true quantization table $\mathbf{q}$ directly from the image under investigation since it is not provided in the decompressed JPEG. Ideally, and for the most general case, each quantization step should be estimated separately for each DCT mode k, l since JPEG images can have non-standard quantization tables. This problem belongs to the field of image forensics and is well studied [11], [35]. However, the exact quantization steps are not needed to apply the JCA because estimating the Q errors is an *easier* task compared to estimating the exact quantization steps. Instead, we need only find a table $\widehat{\mathbf{q}}$ such that the estimated Q errors $\widehat{\boldsymbol{\varepsilon}} \triangleq \mathrm{err}_{\widehat{\mathbf{q}}}(\mathbf{y})$ are close in distribution to the true Q errors $\boldsymbol{\varepsilon}$. In particular, it is enough to estimate a divisor of the true quantization step — the so-called "sufficient" steps defined in Appendix A. Additionally, indeterminable steps [35] that may occur for high frequencies k, l do not pose a problem for the JCA either. Both cases are explained and discussed in more detail in Appendix A.

A comprehensive study of the effects of incorrectly estimated quantization steps on the accuracy of machine learning based JCAs is well beyond the scope of this paper mostly due to the enormous diversity of custom quantization tables in use today. Due to space limitations in this paper, we postpone such study to future work and limit ourselves to standard tables so we may estimate the QF instead of individual quantization steps. We propose a simple maximum likelihood estimator (MLE) and show that its estimation accuracy is high enough so the effects of estimating incorrect tables / steps on steganalysis can be ignored. The reader is referred to [11], [35] for further discussion on incorrectly estimated steps and for estimation techniques more powerful than the MLE proposed.

Given a collection of observed blocks $\mathcal{B}$ (after block elimination), we can estimate the standard quantization table by maximizing the log-likelihood over all qualities $QF \in \{1, \ldots, 100\}$:

$$\widehat{\mathbf{q}} = \underset{QF}{\operatorname{argmax}} \sum_{\mathbf{x} \in \mathcal{B}} \sum_{k,l=0}^{7} \log f_{y_{kl}}\left((\mathbf{D}\mathbf{x})_{kl}\right), \qquad (22)$$

where $(\mathbf{D}\mathbf{x})_{kl} = y_{kl}$ denotes the kl^{th} non-rounded recompressed DCT coefficient for a block. From Eq. (10), the PDF of y_{kl} can be expressed as

$$f_{y_{kl}}(u) = \sum_{n \in \mathbb{Z}} \frac{\mathbb{P}(\widetilde{y}_{kl}^{(0)} = n q_{kl})}{\sqrt{2\pi s_{kl}}} \exp\left(-\frac{(u - n q_{kl})^2}{2 s_{kl}}\right), \quad (23)$$

where $\mathbb{P}(\widetilde{y}_{kl}^{(0)} = n q_{kl})$ is the prior probability that $y_{kl}^{(0)}$ had quantized to $n q_{kl}$. Each step q_{kl} is computed as per the JPEG standard for every quality factor. In practice, for each mode (k, l) we must estimate $\mathbb{P}(\widetilde{y}_{kl}^{(0)} = n q_{kl})$ using a quantity $\widehat{P}_{kl}(n q_{kl})$ derived from the decompressed JPEG itself. For simplicity, if $|n q_{kl}| \leq M_{kl}$, we set

$$\widehat{P}_{kl}(n q_{kl}) = 1/(2M_{kl} + 1) \qquad (24)$$

and $\widehat{P}_{kl}(n q_{kl}) = 0$ otherwise where $M_{kl} = \max_{\mathbf{x} \in \mathcal{B}} |\widetilde{y}_{kl}|$ is the maximum realization of $|\widetilde{y}_{kl}|$ attained across all blocks. Figure 4 shows the accuracy of estimating the correct QF from cover (solid line) and stego (dashed line) images. The authors deem this accuracy to be high enough to have a minimal effect on steganography detection in practice.

X. Conclusions

This paper revisits the JPEG Compatibility Attack in light of the most recent advancements in steganalysis

[5] Also note that this effect of non-zero mean for $\varepsilon_{kl}^{(0)}$ is mitigated for lower qualities – the increased variance of $\varepsilon_{kl}^{(0)}$ makes the wrapped Gaussian uniform.

Table IV
Testing accuracy for SQY-SRNet trained and tested on mismatched quantizers and combinations of decompressors. Embedded using MiPOD at 0.01 bpp.

		Train quantizer: round Test quantizer: trunc					Train quantizer: trunc Test quantizer: round				
QF	Train decomp.	Test decompressor					Test decompressor				
		imread	float	int	convert	PIL	imread	float	int	convert	PIL
100	imread	.5004	.5007	.5004	.5007	.5004	.5049	.5040	.5049	.5045	.5047
	PIL	.5004	.5005	.5004	.5005	.5004	.5050	.5044	.5049	.5052	.5052
99	imread	.8095	.8969	.8093	.8962	.8098	.9335	.9141	.9362	.9133	.9351
	PIL	.7595	.8546	.7591	.8541	.7586	.9204	.8979	.9247	.8984	.9245
95	imread	.9992	.9995	.9995	.9993	.9995	.9993	.9991	.9993	.9993	.9994
	PIL	.9991	.9993	.9994	.9994	.9995	.9996	.9996	.9993	.9994	.9994

Table V
Testing accuracy for SQY-SRNet trained on both round and trunc at QF 100. SQY-SRNet is tested on two sets: TST only quantized with round and TST only quantized with trunc.

Test quant.	Train decomp.	Test set decompressor				
		imread	float	int	convert	PIL
round	imread	.9719	.9545	.9690	.9548	.9735
	PIL	.9676	.9487	.9674	.9503	.9695
trunc	imread	.8829	.8590	.8857	.8634	.8818
	PIL	.8738	.8452	.8762	.8478	.8705

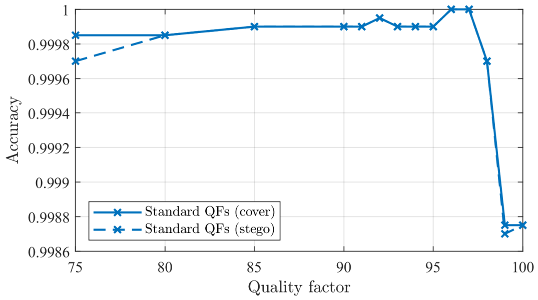

Figure 4. The ratio of images in BOSSBOWS2 whose quality factors were correctly estimated from covers (solid) and stegos (dashed) embedded using MiPOD at 0.01 bpp.

as well as steganography. The focus is on detection of modern content-adaptive embedding schemes and high quality factors when previous state-of-the-art methods experience computational complexity issues and loss of accuracy. Close attention is paid to the robustness of the proposed detectors to JPEG compressors and DCT coefficient quantizers. To better understand the observed trends in accuracy of various implementations of the JCA w.r.t. the quality factor and the effects of different JPEG quantizers, the authors derived a likelihood ratio test under mild modeling assumptions.

To summarize, the best detector was a SQY-SRNet, a two-channel SRNet trained on the image and its SQ error. It exhibited a markedly better accuracy than previous art especially for high JPEG qualities and small payloads. Since the DCT quantizer used for the cover JPEG image and the decompressor are not available to the steganalyst to build the training datasets, this paper includes a comprehensive study of the robustness of the SQY-SRNet w.r.t. these unknowns. We found that training SQY-SRNet on images obtained using both DCT quantizers and using Matlab's `imread` for decompression gave the best generalized results. This detector enjoys a similiar level of accuracy as the clairvoyant detectors informed by and trained on the right combination of cover JPEG quantizer and decompressor.

Our future effort will be directed towards extending the JCA to color images and to make it robust to errors when estimating custom quantization tables.

References

[1] S. Agarwal and H. Farid. Photo forensics from rounding artifacts. In C. Riess and F. Schirrmacher, editors, *The 8th ACM Workshop on Information Hiding and Multimedia Security*, Denver, Colorado, June 22–25, 2020. ACM Press.

[2] P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

[3] P. Bas and T. Furon. BOWS-2. http://bows2.ec-lille.fr, July 2007.

[4] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Science, pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[5] R. Böhme. *Advanced Statistical Steganalysis.* Springer-Verlag, Berlin Heidelberg, 2010.

[6] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019.

[7] M. Boroumand and J. Fridrich. Synchronizing embedding changes in side-informed steganography. In *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2020*, San Francisco, CA, January 26–30 2020.

[8] J. Butora and J. Fridrich. Reverse JPEG compatibility attack. *IEEE Transactions on Information Forensics and Security*, 15:1444–1454, 2020.

[9] J. Butora and J. Fridrich. Steganography and its detection in JPEG images obtained with the "trunc" quantizer. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 4–8, 2020.

[10] R. Cogranne. Selection-channel-aware reverse JPEG compatibility for highly reliable steganalysis of JPEG images. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, pages 2772–2776, Barcelona, Spain, May 4–8, 2020.

[11] J. Fridrich, M. Goljan, and R. Du. Steganalysis based on JPEG compatibility. In A. G. Tescher, editor, *Special Session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications IV*, volume 4518, pages 275–280, Denver, CO, August 20–24, 2001.

[12] L. Guo, J. Ni, and Y. Q. Shi. Uniform embedding for efficient JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 9(5):814–825, May 2014.

[13] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

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

[15] V. Holub, J. Fridrich, and T. Denemark. Universal distortion design for steganography in an arbitrary domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop*, 2014:1, 2014.

[16] X. Hu, J. Ni, W. Su, and J. Huang. Model-based image steganography using asymmetric embedding scheme. *Journal of Electronic Imaging*, 27(4):1 – 7, 2018.

[17] F. Huang, W. Luo, J. Huang, and Y.-Q. Shi. Distortion function designing for JPEG steganography with uncompressed side-image. In W. Puech, M. Chaumont, J. Dittmann, and P. Campisi, editors, *1st ACM IH&MMSec. Workshop*, Montpellier, France, June 17–19, 2013.

[18] P. E. Jupp. A Poincaré limit theorem for wrapped probability distributions on compact symmetric spaces. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 95, pages 329–334. Cambridge University Press, 1984.

[19] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*, volume II. Upper Saddle River, NJ: Prentice Hall, 1998.

[20] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–17, San Jose, CA, January 27–31, 2008.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. http://arxiv.org/abs/1412.6980.

[22] J. Kodovský and J. Fridrich. JPEG-compatibility steganalysis using block-histogram of recompression artifacts. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 78–93, Berkeley, California, May 15–18, 2012.

[23] G. Kurz, I. Gilitschenski, and U. D. Hanebeck. Efficient evaluation of the probability density function of a wrapped normal distribution. In *2014 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–5. IEEE, 2014.

[24] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[25] B. Li, M. Wang, X. Li, S. Tan, and J. Huang. A strategy of clustering modification directions in spatial image steganography. *IEEE Transactions on Information Forensics and Security*, 10(9):1905–1917, September 2015.

[26] Bin Li, Tian-Tsong Ng, Xiaolong Li, Shunquan Tan, and Jiwu Huang. Revealing the trace of high-quality jpeg compression through quantization noise analysis. *IEEE Transactions on Information Forensics and Security*, 10(3):558–573, 2015.

[27] W. Luo, Y. Wang, and J. Huang. Security analysis on spatial ±1 steganography for JPEG decompressed images. *IEEE Signal Processing Letters*, 18(1):39–42, 2011.

[28] K. V. Mardia and P. E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistic. John Wiley & Sons, Inc., 1999.

[29] V. K. Nath, , and D. Hazarika. Comparison of generalized Gaussian and Cauchy distributions in modeling of dyadic rearranged 2D DCT coefficients. In *3rd National Conference on Emerging Trends and Applications in Computer Science (NCETACS)*, pages 89–92, March 2012.

[30] C. Pasquini and R. Böhme. Towards a theory of JPEG block convergence. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 550–554. IEEE, 2018.

[31] W. Pennebaker and J. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[32] V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[33] L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018.

[34] A. Sripad and D. Snyder. A necessary and sufficient condition for quantization errors to be uniform and white. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(5):442–448, 1977.

[35] Thanh H. Thai, R. Cogranne, F. Retraint, and Thi-Ngoc-Canh Doan. JPEG quantization step estimation and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):123–133, 2017.

[36] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[37] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, November 2017.

[38] M. Yedroudj, M. Chaumont, and F. Comby. How to augment a small learning set for improving the performances of a CNN-based steganalyzer? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

[39] M. Yedroudj, F. Comby, and M. Chaumont. Yedroudj-net: An efficient CNN for spatial steganalysis. In *IEEE ICASSP*, pages 2092–2096, Alberta, Canada, April 15–20, 2018.

[40] Y. Yousfi and J. Fridrich. An intriguing struggle of CNNs in JPEG steganalysis and the one-hot solution. *IEEE Signal Processing Letters*, 27:830–834, 2020.

[41] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, May 2018.

Appendix

In this appendix, we explain why the steganalyst only needs to estimate sufficient steps for the JCA to apply as they provide approximately the same Q errors. We also touch upon indeterminable steps.

Suppose q_{kl} is the true quantization step, and let $f_{\widehat{\varepsilon}_{kl}}$ and $f_{\varepsilon_{kl}}$ denote the PDFs of the estimated Q error $\widehat{\varepsilon}_{kl}$ and true Q error ε_{kl}, respectively. We say an estimated quantization step $\widehat{q}_{kl}$ is *sufficient* if 1) $\widehat{q}_{kl} = q_{kl}$, or 2) $q_{kl} > \widehat{q}_{kl} \geq 2$ and $\widehat{q}_{kl}$ divides q_{kl}.

Proposition 5. *If $\widehat{q}_{kl}$ is sufficient, then $|f_{\widehat{\varepsilon}_{kl}}(u) - f_{\varepsilon_{kl}}(u)| \leq C \doteq 3.43 \times 10^{-3}$ for all $u \in \mathbb{R}$.*

Informally, Proposition 5 gives a sufficient condition under which $f_{\widehat{\varepsilon}_{kl}}(u) \approx f_{\varepsilon_{kl}}(u)$ (meaning "approximately equal") within some negligibly small uniform error C. The proposition is trivial to prove under the condition $\widehat{q}_{kl} = q_{kl}$, so we turn to the case $q_{kl} > \widehat{q}_{kl} \geq 2$ and $\widehat{q}_{kl}$ divides q_{kl}. The density $f_{\widehat{\varepsilon}_{kl}}$ is obtained by wrapping $f_{y_{kl}}$ (23) onto a circle of circumference $\widehat{q}_{kl}$: $f_{\widehat{\varepsilon}_{kl}}(u) = \sum_{m \in \mathbb{Z}} f_{y_{kl}}(u + m\widehat{q}_{kl})$ for $u \in [-\widehat{q}_{kl}/2, \widehat{q}_{kl}/2)$ and $f_{\widehat{\varepsilon}_{kl}}(u) = 0$ otherwise.

When $|u| \geq \widehat{q}_{kl}/2$, observe that $f_{\varepsilon_{kl}}(u) \approx 0 = f_{\widehat{\varepsilon}_{kl}}(u)$.[6] In particular, $|f_{\widehat{\varepsilon}_{kl}}(u) - f_{\varepsilon_{kl}}(u)| = f_{\varepsilon_{kl}}(u) \leq C$ by direct evaluation of the maximum.[7]

[6] This is due to the fact that $s_{kl} \leq 1/12$ and $q_{kl} > \widehat{q}_{kl} \geq 2$.

[7] $f_{\varepsilon_{kl}}(u)$ is maximized when $|u| = 1$, $\widehat{q}_{kl} = 2$, $q_{kl} = 4$, $s_{kl} = 1/12$.

Table VI
TESTING ACCURACY FOR SQY-SRNET. (DE)COMPRESSED WITH MATLAB'S IMWRITE.

	Payload	QF										
	(bpp)	90	91	92	93	94	95	96	97	98	99	100
MiPOD	0.02	.9996	.9996	.9996	.9995	.9994	.9996	.9997	.9996	.9999	.9924	.9899
	0.01	.9992	.9997	.9994	.9993	.9994	.9996	.9995	.9994	.9994	.9856	.9721
	0.005	.9988	.9983	.9983	.9984	.9983	.9986	.9975	.9960	.9903	.9094	.8634
	0.002	.9918	.9905	.9916	.9874	.9856	.9791	.9747	.9587	.9231	.7512	.6891
HILL	0.02	.9998	.9997	.9997	.9997	.9996	.9997	.9998	.9998	.9998	.9981	.9982
	0.01	.9994	.9993	.9995	.9994	.9995	.9996	.9996	.9997	.9996	.9964	.9950
	0.005	.9975	.9981	.9990	.9981	.9986	.9983	.9989	.9985	.9968	.9783	.9698
	0.002	.9926	.9948	.9926	.9927	.9906	.9885	.9885	.9841	.9683	.8958	.8717
S-UNI	0.02	.9995	.9995	.9997	.9996	.9996	.9997	.9998	.9999	.9999	.9970	.9959
	0.01	.9994	.9995	.9997	.9994	.9996	.9997	.9990	.9997	.9996	.9934	.9918
	0.005	.9991	.9988	.9985	.9985	.9989	.9994	.9983	.9976	.9968	.9650	.9498
	0.002	.9934	.9923	.9910	.9921	.9908	.9895	.9831	.9721	.9603	.8511	.8109
WOW	0.02	.9997	.9997	.9997	.9998	.9997	.9996	.9998	.9995	.9998	.9990	.9981
	0.01	.9996	.9996	.9996	.9996	.9997	.9991	.9999	.9993	.9995	.9965	.9959
	0.005	.9987	.9991	.9985	.9988	.9983	.9990	.9986	.9987	.9972	.9804	.9725
	0.002	.9920	.9942	.9917	.9929	.9912	.9888	.9872	.9813	.9726	.8978	.8630

Table VII
TESTING ACCURACY FOR RRH. (DE)COMPRESSED WITH MATLAB'S IMWRITE.

	Payload	QF										
	(bpp)	90	91	92	93	94	95	96	97	98	99	100
MiPOD	0.02	.9995	.9994	.9987	.9970	.9924	.9778	.9189	.8722	.9052	.9239	.9290
	0.01	.9993	.9998	.9991	.9984	.9826	.9035	.7778	.7114	.7172	.7543	.7577
	0.005	.9994	.9996	.9996	.9983	.9147	.7610	.6497	.6016	.5999	.6175	.6257
	0.002	.9939	.9942	.9982	.9519	.7247	.6157	.5597	.5370	.5342	.5390	.5438
HILL	0.02	.9906	.9903	.9891	.9763	.9440	.9087	.8974	.9242	.9676	.9796	.9651
	0.01	.9926	.9902	.9881	.9807	.9044	.8165	.7613	.7551	.8745	.8736	.8312
	0.005	.9898	.9881	.9886	.9725	.8263	.7052	.6508	.6301	.7240	.7341	.6874
	0.002	.9817	.9830	.9837	.9219	.6971	.5987	.5623	.5505	.5721	.5872	.5705
S-UNI	0.02	.9980	.9982	.9977	.9924	.9819	.9562	.9078	.8816	.9368	.9536	.9501
	0.01	.9984	.9979	.9961	.9939	.9598	.8744	.7776	.7303	.7930	.8142	.7964
	0.005	.9978	.9974	.9967	.9939	.8884	.7503	.6497	.6216	.6363	.6649	.6572
	0.002	.9930	.9931	.9965	.9540	.7306	.6137	.5680	.5472	.5488	.5614	.5575
WOW	0.02	.9932	.9929	.9911	.9754	.9448	.9127	.8919	.9210	.9665	.9787	.9677
	0.01	.9931	.9911	.9893	.9766	.9136	.8248	.7709	.7547	.8608	.8756	.8353
	0.005	.9905	.9898	.9899	.9766	.8440	.7208	.6526	.6268	.7070	.7260	.6873
	0.002	.9840	.9881	.9868	.9354	.7112	.6104	.5668	.5511	.5621	.5852	.5699

For $u \in [-\widehat{q}_{kl}/2, \widehat{q}_{kl}/2)$, observe that the Gaussian terms in $f_{\widehat{\varepsilon}_{kl}}$ are offset by integer multiples of $\widehat{q}_{kl}$ because

$$m\widehat{q}_{kl} - nq_{k\ell} = m\widehat{q}_{kl} - nj\widehat{q}_{kl} = (m-nj)\widehat{q}_{kl}, \qquad (25)$$

for some $j \in \mathbb{Z}_{>0}$. By swapping the sums in $f_{\widehat{\varepsilon}_{kl}}$, we can re-index the sum over m according to Eq. (25) to produce

$$\begin{aligned} f_{\widehat{\varepsilon}_{kl}}(u) &= \sum_{n\in\mathbb{Z}} \frac{\mathbb{P}(\widetilde{y}_{kl}^{(0)} = nq_{kl})}{\sqrt{2\pi s_{kl}}} \sum_{m\in\mathbb{Z}} \exp\left(-\frac{(u+m\widehat{q}_{kl})^2}{2s_{kl}}\right) \\ &= \frac{1}{\sqrt{2\pi s_{kl}}} \sum_{m\in\mathbb{Z}} \exp\left(-\frac{(u+m\widehat{q}_{kl})^2}{2s_{kl}}\right), \end{aligned} \qquad (26)$$

for $u \in [-\widehat{q}_{kl}/2, \widehat{q}_{kl}/2)$. The last line in Eq. (26) follows from $\sum_{n\in\mathbb{Z}} \mathbb{P}(\widetilde{y}_{kl}^{(0)} = nq_{kl}) = 1$. Observe that $|f_{\widehat{\varepsilon}_{kl}}(u) - f_{\varepsilon_{kl}}(u)|$ is upper bounded by $g(u; 0, s_{kl}, \widehat{q}_{kl})$ without the $n = 0$ term which has a maximum of C when $u \in [-\widehat{q}_{kl}/2, \widehat{q}_{kl}/2)$. Thus, we get $f_{\widehat{\varepsilon}_{kl}}(u) \approx f_{\varepsilon_{kl}}(u)$, proving Proposition 5 as desired.[8]

In practice, there is another (content-dependent) sufficient condition that commonly holds for lower qualities: $\widehat{q}_{kl}, q_{kl} \geq 2$ and the DCTs y_{kl} are contained within the interval $[-1, 1)$ across all sampled blocks. This condition is known as the "indeterminable" case in [35] and is a point of failure for many quantization step estimation methods. However, this case benefits the steganalyst since $\mathrm{err}_{\widehat{q}_{kl}}(y_{kl}) = y_{kl} = \mathrm{err}_{q_{kl}}(y_{kl})$ for any chosen step $\widehat{q}_{kl} \geq 2$.

Observe that Proposition 5 holds when either the round or the trunc quantizer is used for the initial JPEG compression; the differences in quantization bins only affect the values of $\mathbb{P}(\widetilde{y}_{kl}^{(0)} = nq_{kl})$ and s_{kl}. Also note that the proposition considered only cover images. When estimating the steps from stego images, the variance s_{kl} is replaced with $s_{kl}+r_{kl}$, which has a negligible effect on the accuracy of the Q error for the most relevant case of small payloads $r_{kl} \ll 1$. Finally, quantization step estimation methods such as the one proposed in [35] will often select a divisor of the true step when wrong, which tells us that steps are commonly sufficient in practice.

[8] $g(u; 0, s_{kl}, \widehat{q}_{kl})$ without the $n = 0$ term is maximized at $|u| = 1$, $\widehat{q}_{kl} = 2$, $s_{kl} = 1/12$.