https://doi.org/10.2352/EI.2023.35.4.MWSF-374

# Cost Polarization by Dequantizing for JPEG Steganography

*Edgar Kaziakhmedov, Yassine Yousfi, Eli Dworetzky, and Jessica Fridrich, Department of ECE, SUNY Binghamton, NY, USA, {ekaziak1, yyousfi1, edworet1, fridrich}@binghamton.edu*

## Abstract

*In this article, we study a recently proposed method for improving empirical security of steganography in JPEG images in which the sender starts with an additive embedding scheme with symmetrical costs of* $\pm 1$ *changes and then decreases the cost of one of these changes based on an image obtained by applying a deblocking (JPEG dequantization) algorithm to the cover JPEG. This approach provides rather significant gains in security at negligible embedding complexity overhead for a wide range of quality factors and across various embedding schemes. Challenging the original explanation of the inventors of this idea, which is based on interpreting the dequantized image as an estimate of the precover (uncompressed) image, we provide alternative arguments. The key observation and the main reason why this approach works is how the polarizations of individual DCT coefficients work together. By using a MiPOD model of content complexity of the uncompressed cover image, we show that the cost polarization technique decreases the chances of "bad" combinations of embedding changes that would likely be introduced by the original scheme with symmetric costs. This statement is quantified by computing the likelihood of the stego image w.r.t. the multivariate Gaussian precover distribution in DCT domain. Furthermore, it is shown that the cost polarization decreases spatial discontinuities between blocks (blockiness) in the stego image and enforces desirable correlations of embedding changes across blocks. To further prove the point, it is shown that in a source that adheres to the precover model, a simple Wiener filter can serve equally well as a deep-learning based deblocker.*

## Introduction

Steganography by minimizing an additive distortion or detectability function is a well established and successful approach to building data hiding schemes with a high level of empirical security in practice [21, 20, 26, 34, 18, 17, 35]. Fundamentally, this direction was enabled by the invention of syndrome-trellis codes [13] that operate near the rate–distortion bound. An additive distortion function, though, cannot account for the effect of interaction of individual embedding changes, which requires the use of non-additive distortion functions. A general framework for embedding with non-additive distortion is the so-called Gibbs construction [12], which is applicable whenever the distortion can be written as a sum of locally supported potentials. The first embedding schemes with non-additive distortion were HUGO [32] (which used a heuristic iterative cost adjustment) and HUGO-BD [12] based on the bounding distortion of HUGO. Unfortunately, it is apparently very difficult to design non-additive distortion that properly captures the interaction of embedding changes and their impact on detectability. In [33], the authors introduced a greedy minimization technique that they applied to embedding with the non-additive UNIWARD distortion [21]. Disappointingly, a smaller total embedding distortion did not correlate with empirical detectability.[1]

It has only gradually been recognized by the community that taking into account mutual interaction of embedding changes can have a significant impact on empirical security. The first work in this direction was heuristic and restricted to the spatial (pixel) domain. The Clustering Modification Directions (CMD) [27] employed additive embedding on four interleaved sub-lattices with a step that included slashing the costs of embedding modifications on sub-lattices that have not yet been embedded to encourage neighboring embedding changes to be correlated. A different method for turning an additive scheme into non-additive [9] leveraged the Gibbs construction for practical embedding. A model-based approach was described in [23] by minimizing the variational approximation of the KL divergence for an asymmetric Gaussian mixture by postulating that the neighboring modifications change the local pixel mean. A heuristic method that combined side-information with clustering modification direction was described in [6].

Interaction of embedding changes is far stronger in the JPEG domain because modification patterns of DCT coefficients from the same JPEG $8 \times 8$ block overlap and because the modifications are stronger in terms of their energy due to quantization. The impact on detectability is further increased at the block boundaries where the embedding increases spatial discontinuity also called blockiness [14]. It is consequently completely unclear how to properly contain all this complexity heuristically via a non-additive distortion function. Assessing the cost of simultaneous modifications of a large number of DCT coefficients is understandably significantly more difficult than finding heuristic costs of individual embedding changes that correlate with detectability in practice. In [29], the authors show that non-additive distortion in JPEG domain needs to properly capture correlations (and anticorrelations) of embedding changes of DCT coefficients from neighboring blocks. A more feasible and theoretically well founded ap-

[1] This observation was already made in [19].

proach to this problem, however, is to start with a cover source model.

Natural Steganography (NS) [1] starts by considering the heteroscedastic model of sensor photonic noise (also called ISO noise). For a sufficiently simple development pipeline, it is possible to derive the statistical distribution of pixels (and DCT coefficients for the JPEG domain [38]), so that the embedding impact can be masked as taking the same cover image with a higher ISO setting. While NS can embed very large payloads with negligible detectability by modern steganalysis tools, it comes with rather significant limitations and overhead complexity. In particular, the sender needs access to the RAW undeveloped image, then estimate the parameters of the heteroscedastic noise (which depend on the ISO settings), and finally needs to adopt a rather simplistic development, essentially creating a special source where steganography is easier. To remove the need for unrealistically simple development pipeline and access to the RAW image, in a series of papers [16, 15] the authors showed that a significant security boost can be obtained for side-informed UNIWARD (SI-UNIWARD) [21] by estimating the photonic sensor noise model and by using a linearized model of the development pipeline. This approach cannot be adapted when the sender only has a JPEG file of unknown pedigree. Moreover, considering only ISO noise prevents embedding in lower quality JPEGs (e.g., lower than 95) due to the fact that this noise component is largely decimated due to harsh quantization. For lower quality JPEGs, the embedding needs to "hide behind content complexity," which can be thought of as the inability to estimate content [16]. Content complexity is notoriously difficult to model using statistical approaches at least to a degree that is required for building secure steganographic schemes. Without a model or any form of side-information, only rather incremental improvement has been reported over classical JPEG steganography with additive distortion, such as the popular J-UNIWARD algorithm [37, 22, 36].

A simple way to force an additive ternary embedding scheme to consider interactions among embedding changes is via cost polarization, which we define here as purposely breaking the symmetry of costs of opposite modification directions. This way, certain combinations of embedding changes will be occurring with higher probability than some other combinations. In other words, we probabilistically enforce certain relationships among embedding changes while keeping the simplicity of embedding with an additive distortion function. The authors of [10] showed that a quite significant improvement in empirical security of J-UNIWARD can be obtained by utilizing a second JPEG image of the same scene by decreasing the costs of changing the DCT coefficient in the direction of the value from the second JPEG while keeping the costs unchanged for all coefficients that were the same for both exposures. The second exposure is a form of side-information, which makes this method also limited in its applicability.

Recently, a cost-polarization method was proposed [39, 28] based on creating side-information from the cover JPEG by applying to it a JPEG deblocker.[2] The deblocked image is used as side-information for embedding in a manner similar to how side-informed steganography with precover has been used in the past – by decreasing the costs of modifications towards the deblocked image. With the right deblocker, the improvement in security of this simple method over the original embedding scheme can be quite remarkable. The authors of this idea view the deblocked image as an estimate of the unquantized cover image (precover), explaining thus the technique as embedding with side-information. In this paper, we critique this explanation, pointing out that even the best deblockers are only slightly better than a random guesser in determining the signs of quantization errors. Instead, we hypothesize that the deblocked image exerts pressure on the embedding to generate a stego image that is more compatible with the distribution of the precover. To this end, we adopt MiPOD's content complexity model in the spatial domain, port it to the DCT domain, and inspect the likelihood under the precover distribution. This likelihood correlates with empirical detectability in practice. To further prove the point, we show that in a cover source that follows the adopted model perfectly, a simple Wiener filter can serve as a deblocker as efficiently in terms of empirical security as more complex deep learning deblockers.

After introducing basic notational conventions and symbols, in Section "Cost Polarization" we describe the cost polarization method as introduced in [39, 28]. Then, we give details of the datasets used for experiments and the detectors used to evaluate empirical detectability. To motivate the research, in Section "Base Experiments" we report the performance of cost polarization with a variety of JPEG deblockers applied to J-UNIWARD and contrast them with SI-UNIWARD and BACKPACK [4]. The following section contains a critique of the explanation of why cost polarization works as presented by its inventors. In Section "Insight from modeling content complexity," we lay out arguments that the internal mechanism is rooted in the way the polarities work together. A numerical measure based on a precover model is proposed and shown to correlate with empirical detectability. Additional insight is obtained with a deblocker implemented as a Wiener filter. The Section "Interblock Relationships" inspects the impact of embedding on relationships among DCTs across blocks. Section "Deblockers" contains implementation details of all deblockers and their mutual comparison in terms of PSNR w.r.t. the uncompressed image. Section "BACKPACK" provides the reader with the details of how it used in this paper. The paper is concluded in the last Section "Conclusions."

## Notation

In this section, we introduce basic notational conventions. Vectors and matrices will be typed in boldface while reserving uppercase symbols for matrices. Transpose of

[2] A JPEG deblocker attempts to "dequantize" a decompressed JPEG (bring it closer to the uncompressed image within certain metric) in order to improve the visual quality and partially undo the compression loss.

matrix $\mathbf{A}$ will be denoted $\mathbf{A}^T$. Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$; uniform distribution on interval $[a,b]$ is denoted $\mathcal{U}[a,b]$. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$.

In this paper, we work solely with grayscale images. Quantized DCT coefficients in a JPEG file will be represented in a block-by-block fashion by a integer-valued matrix $\mathbf{d}$ of the same dimensions as the image. When addressing a specific $8\times 8$ DCT block, we will use the same symbol as the meaning should be clear from context. A steganographic scheme for JPEG images typically modifies the values of $\mathbf{d}$ by ±1. The dequantized cofficients are captured with matrix $\mathbf{c}=\mathbf{q}\cdot\mathbf{d}$ whose elements are multiples of the quantization steps ($\mathbf{q}$ is the $8\times 8$ luminance quantization matrix and the operation $'\cdot'$ is the elementwise multiplication).

When the spatial positions of DCT coefficients within the image are not relevant, we will assume that $\mathbf{d}$ or $\mathbf{c}$ are vectors obtained by unwrapping the corresponding matrices according to some fixed order.

## Cost polarization

In this section, we describe the cost polarization algorithm that is studied in this paper. We adopt the same acronym for this approach as in the original work [28], where it was called SIEp (Side Information Estimation Polarity), and also use the same language.

This method starts with an existing additive ternary steganographic scheme, which changes cover DCT coefficients $d_i$ by ±1 to embed the secret message. Let $\hat{d}_i$ be the corresponding non-rounded DCT coefficients of the same cover JPEG image after processing it by a JPEG deblocking algorithm. To remove any source of ambiguity, $\hat{d}_i$ are obtained by applying the DCT blockwise to the deblocked image in spatial domain and dividing by the quantization matrix without rounding to integers. Denoting the costs of the ternary steganographic scheme with $\rho_i$, the SIEp method costs are modulated (slashed) by a factor $0<\eta<1$ ($\eta=0.65$ everywhere in this paper)

$$\begin{aligned}\rho_i^{\mathrm{SIEp}}(s_i) &= \eta\rho_i\\ \rho_i^{\mathrm{SIEp}}(-s_i) &= \rho_i. \end{aligned}\tag{1}$$

Here, $s_i=\mathrm{sign}(\hat{e}_i)$ is the sign of the "estimated rounding error",$\hat{e}_i$, given by

$$\hat{e}_i=\hat{d}_i-d_i.\tag{2}$$

Based on the interpretation in [28], $\hat{e}_i$ is an estimate of the rounding error during the original JPEG compression that produced the cover $d_i$. Note that the resulting embedding scheme will work with asymmetric costs $\rho_i^{\mathrm{SIEp}}(\pm s_i)$ of changing $c_i$ by $\pm s_i$. Also note that the cost modulation is different from typical side-informed schemes, which modulate costs by $1-2|e_i|$, where $e_i\in(-1/2,1/2]$ is the true rounding error during JPEG compression. As the authors of [28] point out, the modulation (1) does not "trust" the estimated rounding errors as much as a typical side-informed scheme because the estimated rounding error is only an approximation and also because it may not be in the range $(-1/2,1/2]$.

We included in our study two more versions of the SIEp algorithm: SITp (side information true polarity) and SIRp (side information random polarity) . The first one uses the uncompressed image instead of a deblocked image, and thus knows the signs of the true quantization errors $e_i$. This version was added to see what would happen if the deblocker gave a perfect output. SIRp selects the directions for modulation randomly with equal probability 1/2. This algorithm was added to see the impact of cost slashing with a deblocker that randomly guesses the directions for slashing.

## Datasets and detectors

All experiments in this paper are conducted on four datasets, one containing natural images and three artificial sources to gain insight.

The dataset of natural images is the union of the BOSSbase 1.01 [2] and BOWS2 [3], each with 10,000 grayscale images resized to $256\times 256$ pixels with `imresize` in Matlab using default parameters and stored them as uncompressed images. To generate JPEG images, each image is processed with in a blockwise manner with the DCT transform with `dct2` in MATLAB. The DCT coefficients are then quantized with a quantization matrix, which depends on a quality factor (QF), and rounded to the closest integer. We use the same pipeline for obtaining JPEGs to avoid the cover-source mismatch (CSM) [25, 30]. We refer to the union for brevity as BB. This dataset is a popular choice for designing detectors with deep learning because small images are more suitable for training deep architectures [41, 5, 42, 43, 40, 44]. The training set (TRN) contains 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) with 1,000 and 5,000 images, respectively.

An artificial version of this dataset [7] was prepared to allow statistical analysis to provide insight. Since we intend to study a wide range of quality factors, instead of using photonic sensor noise as a source of randomness for modeling as in [15, 16], we decided to model content complexity as in the embedding algorithm MiPOD [34]. Thus, we model precover ($p$) pixels in a single $8\times 8$ block as a multi-variate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^{(p)},\boldsymbol{\Sigma}^{(p)})$ with $\boldsymbol{\mu}^{(p)}\in\{0,\ldots,255\}^N$ and a diagonal covariance $\boldsymbol{\Sigma}^{(p)}=\mathrm{diag}(\sigma_1^2,\ldots,\sigma_N^2)$, with pixel variances $\sigma_i^2$ estimated from the uncompressed image using MiPOD's variance estimator. The superscript $p$ is to remind the reader that the objects are in the pixel domain. Since the MiPOD model is really a model of pixel noise residuals, each image from BB was first denoised using the wavelet denoising filter [31] with $\sigma_{\mathrm{Den}}=10$, then its dynamic range was narrowed (and the values rounded to the closest integers $\boldsymbol{\mu}^{(p)}$) to make sure the pixel values after noisification fit within the [0,255] range with high probability. Subsequently, a precover in the pixel domain was obtained by sampling from the MVG, which was achieved by adding to $\boldsymbol{\mu}_i^{(p)}$ independent samples from $\mathcal{N}(0,\sigma_i^2)$. The sampled precovers are rounded to the closest integer. We abbreviate this artificial version of BB as BB1 (and BB1/2), depending on whether we use the exact value of $\sigma_i$ from MiPOD or $\sigma_i/2$. More details describing the creation of this dataset appear in [7].[3]

The third artificial dataset we generated contains multiple samples of the same scene; we call it BB-MS (BB with multiple samples). Note, that the dataset uses only the exact value of $\sigma_i$ from MiPOD. It contains 80 scenes randomly selected from TRN, VAL, and TST set and each scene is sampled 50 times, which amounts to $80 \times 3 \times 50 = 12,000$ images. The main purpose for adding this dataset is to mitigate the impact of acquisition noise on experiments carried out in Section "Insight from modeling content complexity".

Everywhere in this paper, we evaluate the security of embedding algorithms empirically with $P_{\mathrm{E}}$ – the minimum average total detection error under equal priors on the testing set. All deep learning detectors were initialized with JIN-SRNet [8], which is the SRNet [5] pretrained on ImageNet [11] and its stego version embedded with J-UNIWARD [21] with payloads uniformly randomly selected from the interval [0.4, 0.6] bpnzac. The networks were then refined for a given steganalysis task via transfer learning as described in [8]. For artificial datasets BB1 and BB1/2, SRNet is seeded with JIN-SRNet only for detecting J-UNIWARD. Detectors for the other algorithms are seeded with SRNet pretrained on the corresponding artificial dataset and its stego version embedded with J-UNIWARD.

## Base experiments

To ease the reader into the subject and to motivate this study, in this section we report the empirical security of various versions of SIEp applied to J-UNIWARD and contrast with other methods for increasing security, such as SI-UNIWARD [21] and BACKPACK [4]. Table 1 shows the detection error $P_{\mathrm{E}}$ of SRNet for J-UNIWARD and various versions of SIEp for payloads 0.2 and 0.4 bpnzac on BB with costs modulated with $\eta = 0.65$. Five deblockers are studied: Wiener filter with a $3 \times 3$ window implemented as `wiener2` in Matlab with default parameters (image local means and variances are estimated from $3 \times 3$ neighborhood, noise variance is an average of local image variances), jpeg2png,[4] SSRQC [46], and two deep learning deblockers DnCNN [45], and FBCNN [24]. The implementation details of all deblockers used in this paper and their performance are reported in Section "Deblockers."

The results clearly show a substantial gain in empirical security across all quality factors and both payloads. The deblockers are listed from top to bottom according to their ability to deblock in terms of PSNR between the uncompressed image and the deblocked image (see Section "Deblockers"). It is generally true that better deblockers lead to more secure steganographic schemes. The two deep learning deblockers perform practically the same with the exception of QF98 when FBCNN achieves a better performance. Also note that even the simplest deblocker, the $3 \times 3$ Wiener filter, does provide a non-negligible security boost.

In Figure 1 (and Table 1), we compare empirical security of J-UNIWARD in terms of $P_{\mathrm{E}}$ with SIEp($DnCNN$) as the deblocker w.r.t. SI-UNIWARD [21], SITp, SIRp, and BACKPACK [4] with two global iterations against XuNet [40]. The implementation details of BACKPACK for the BB dataset are given in Section "BACKPACK". As expected, SI-UNIWARD offers the best security because it has access to the uncompressed cover image. Having said this, the difference between SIEp with the FBCNN deblocker and SI-UNIWARD is less than 5%. Moreover, SIEp outperforms BACKPACK with two iterations, which is quite remarkable considering the immense computational requirements this advanced algorithm needs.

We also observed that SIEp($DnCNN$) experiences a performance drop for the largest quality factor, while SIEp($FBCNN$) does not suffer from this problem. We hypothesize that this is due to the fact that FBCNN is seeded with weights trained for multiple quality factors. This technique only makes sense for FBCNN, since it contains attention layer parametrized by the quality factor predictor embedded into the network, while DnCNN is a simple encoder-decoder convolutional network.

Table 2 shows two more interesting results. When SIEp is given the true directions of rounding errors (SITp), its security is markedly lower than when the directions are estimated. Apparently, the way the costs are modulated and the accuracy of estimating the quantization error signs need to be optimized jointly. The "harsh" modulation by $\eta$ slashes the costs irrespectively of the magnitude of the quantization error, unlike SI-UNIWARD, which explains why it performs worse than SI-UNIWARD. The second curious result is the fact that when the directions to be modulated are selected randomly (SIRp), the empirical security of the embedding algorithm is unaffected w.r.t. the original scheme, J-UNIWARD. This is an indirect indication of a space for improvement should these directions be selected in a more judicious manner.

## Critique

In the original paper describing SIEp [28], the authors explain their method by referring to the principle of side-informed embedding. The deblocked image is considered as an estimate of the unquantized image and the differences between the DCT coefficients between this image and the cover image are taken as estimates of the quantization errors. However, when we inspect how accurately the deblockers estimate the rounding error, we discover that their ability to guess even the sign of the rounding error is only slightly better than random guessing. Figure 2 shows the accuracy of estimating the rounding error sign as a function of the cost percentile from JPEG images stored at quality 95 and 75. To be more precise, for example the

[3] In contrast to [7], we skipped the step that clips the pixel values to [1,254] since we evaluate the cover likelihood in the DCT domain.

[4] https://github.com/victorvde/jpeg2png

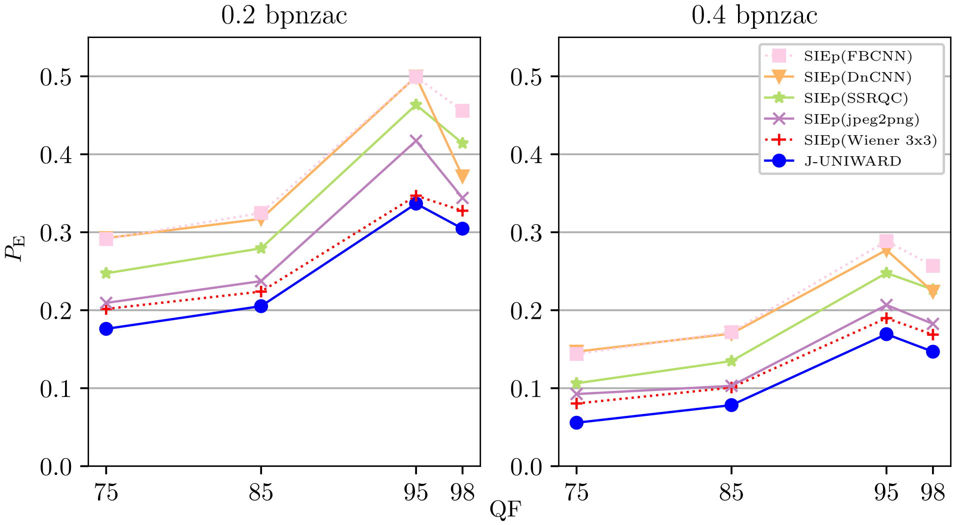


***Figure 1.*** *Detection error $P_{\mathrm{E}}$ of SRNet for J-UNIWARD for four quality factors, two payloads, and five different deblockers on BB.*

| | 0.2 bpnzac | | | | 0.4 bpnzac | | | |
|---|---|---|---|---|---|---|---|---|
| QF | 75 | 85 | 95 | 98 | 75 | 85 | 95 | 98 |
| J-UNIWARD | 0.1761 | 0.2052 | 0.3367 | 0.3046 | 0.0557 | 0.0783 | 0.1693 | 0.1468 |
| SIEp(*Wiener 3×3*) | 0.2016 | 0.2239 | 0.3469 | 0.3276 | 0.0804 | 0.1008 | 0.1898 | 0.1687 |
| SIEp(*jpeg2png*) | 0.2094 | 0.2373 | 0.4173 | 0.344 | 0.0925 | 0.103 | 0.2067 | 0.1825 |
| SIEp(*SSRQC*) | 0.2474 | 0.2794 | 0.4635 | 0.4141 | 0.1063 | 0.1348 | 0.2477 | 0.2265 |
| SIEp(*DnCNN*) | **0.2925** | 0.3172 | **0.4996** | 0.3717 | **0.1469** | 0.17 | 0.2772 | 0.2237 |
| SIEp(*FBCNN*) | 0.2913 | **0.3247** | 0.4993 | **0.4557** | 0.1439 | **0.1719** | **0.2886** | **0.2569** |

**Table 1. Detection error $P_{\mathrm{E}}$ of SRNet for J-UNIWARD for four quality factors, two payloads, and five different deblockers on BB.**

| QF | J-UNI | SIRp | SITp | SIEp(*DnCNN*) | SI-UNI | BP#2 |
|---|---|---|---|---|---|---|
| 95 | 0.1672 | 0.1608 | 0.2022 | 0.2772 | **0.3234** | 0.2461 |
| 75 | 0.0557 | 0.0562 | 0.0783 | 0.1469 | **0.2481** | 0.1215 |

**Table 2. Empirical security measured as $P_{\mathrm{E}}$ with SRNet for J-UNIWARD, SIRp with randomly selected directions for modulation, SITp with the precover as the "deblocked" image, SIEp(*DnCNN*) with DnCNN deblocker, SI-UNIWARD, and BACKPACK with two iterations w.r.t. XuNet. Payload 0.4 bpnzac, BB.**

point on the curve for percentile 25 is the sign prediction accuracy for the 25% of the smallest costs, which correspond to pixels where the majority of embedding changes occur. Figure 3 shows that on average 95% of the embedding changes are made to pixels with 20% of the smallest costs for J-UNIWARD 0.4 bpnzac at JPEG quality 95.

With such low sign estimation accuracy, the cost modulation is essentially random for the DCT coefficients that are primarily used for embedding in terms of the actual embedding changes, which points to a different mechanism responsible for the security boost. We note that a similar level of sign accuracy can be observed for other quality factors and deblockers.

## Insight from modeling content complexity

To obtain insight into the internal mechanism of SIEp, we generate from BB an artificial cover source where the content complexity in pixel domain is modeled as a multivariate Gaussian (MVG) distribution $\mathcal{N}(\boldsymbol{\mu}^{(p)},\boldsymbol{\Sigma}^{(p)})$ with $\boldsymbol{\Sigma}^{(p)} = \mathrm{diag}(\sigma_1^2,\ldots,\sigma_N^2)$. The variances are estimated from the uncompressed BB image using MiPOD's variance estimator and the mean is the BB image processed with a denoising filter, scaled to a narrower dynamic range, and rounded to integers as described in Section Datasets and Detectors and detailed in [7].

Since the spatial model is a collection of independent Gaussian random variables, we can transfer the spatial model to the DCT domain by blocks. Restricting $\boldsymbol{\mu}^{(p)},\boldsymbol{\Sigma}^{(p)}$ to one $8\times 8$ block but keeping the notation for simplicity, $\boldsymbol{\mu}^{(p)} \in \{0,\ldots,255\}^{64}, \boldsymbol{\Sigma}^{(p)} \in \mathbb{R}^{64\times 64}$, by transforming the quantities to the DCT domain, we obtain the mean and covariance matrix for the DCT coefficients (not divided by quantization steps or rounded)

$$\boldsymbol{\mu} = \mathbf{D}\boldsymbol{\mu}^{(p)}$$

$$\boldsymbol{\Sigma} = \mathbf{D}\boldsymbol{\Sigma}^{(p)}\mathbf{D}^T, \tag{3}$$

where $\mathbf{D}$ is an orthonormal $64\times 64$ DCT matrix, and $\boldsymbol{\mu} \in \mathbb{R}^{64}, \boldsymbol{\Sigma} \in \mathbb{R}^{64\times 64}$. The precover DCT coefficients thus follow $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$.

Figure 4 shows a two-dimensional version of this MVG in the DCT domain together with a specific realization (precover) $\mathbf{x}$, its quantization errors $e_i q_i$, and the cover JPEG $\mathbf{c}$ shown with a black circle. We note that except for the largest JPEG qualities, the quantization noise dominates the statistical spread of the MVG. This also means that one cannot adopt fine-quantization assumption for the modeling. Instead, we focus on the fact that SI-UNIWARD modulates the costs of modifications that take the cover (black) to any of the green points by $1-2|e_i|$, while the costs to the red dots are unmodified. In contrast, any embedding method with symmetric costs moves the cover to a green or red point with equal probability. Considering the coarse quantization case we face, this means that the resulting stego image will on average have a much smaller likelihood under the precover MVG than a stego image produced by SI-UNIWARD, which prefers changing the cover to the green points. For the dimensionality of an $8\times 8$ JPEG block, the green dots start occupying increasingly smaller part of the set of all possible stego images ($2^{64}$ out of a total of $3^{64}$ images). While the deblockers are not very accurate in predicting the directions to the green dots (see Figure 2), they do create the right bias on average, preventing combinations of embedding changes with small cover likelihood. This rough reasoning, however, does not explain why SITp is less secure than SIEp with a deblocker.

To properly assess the impact of embedding, one would need to consider the distribution of stego images w.r.t. the cover distribution. Since modeling the directions predicted by a deblocker is rather complex, we adopt two simplifying assumptions:

1. We only consider the impact of embedding on cover likelihood. An embedding scheme that decreases the cover likelihood more will likely be more detectable than one that preserves it.
2. The model adopted in this section only considers interactions of embedding changes within one $8\times 8$ block (intra-block) but does not consider inter-block relationships.

Instead of working with the likelihood itself, we work with the exponent of the MVG $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, which is the squared Mahalanobis distance

$$d^2_{\boldsymbol{\Sigma}}(\mathbf{x},\boldsymbol{\mu}) = (\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \tag{4}$$

with $\boldsymbol{\mu},\mathbf{x}$, and $\boldsymbol{\Sigma}$ restricted to one on JPEG $8\times 8$ block.

Table 3 shows the detection error of SRNet on BB1 and BB1/2 for J-UNIWARD, SI-UNIWARD, SIEp(*DnCNN*) and SIEp(*SSRQC*), and SITp. The embedding algorithms are ordered in the table in increasing order by their empirical security. We observe the same order in terms of $P_{\mathrm{E}}$ as for BB dataset. Note that SITp always performs worse than SIEp(*DnCNN*). For BB1, SIEp(*DnCNN*) and SIEp(*SSRQC*) offer similar security but SIEp(*SSRQC*) performs much worse for the less noisy dataset BB1/2. For this dataset, SIEp(*DnCNN*) also performs very close to SI-JUNI. We note that the DnCNN deblocker was trained on the artificial dataset to have the deblocker match the cover source.

To analyze how well various distance measures capture detectability, we use the BB-MS dataset to reduce the impact of acquisition noise within a scene. Figure 5 shows confusion matrices for four different stego algorithms and four quality factors for images from BB-MS constructed by the following procedure. For a fixed cover $\mathbf{c}$ in BB-MS, we generate a stego image $\mathbf{s}^S$ for each stego scheme $S$ in Table 3 except SIEp(*SSRQC*). The four stego images are ranked based on ordering distance measurements, e. g., $d_{\boldsymbol{\Sigma}}(\mathbf{s}^S,\boldsymbol{\mu}) - d_{\boldsymbol{\Sigma}}(\mathbf{c},\boldsymbol{\mu})$ in a non-decreasing fashion using $\boldsymbol{\mu}$ that corresponds to the scene from which $\mathbf{c}$ was taken. The rank of $\mathbf{s}^S$ w.r.t. distance $d$ is denoted $R_d(S,\mathbf{c})$ . Thus,

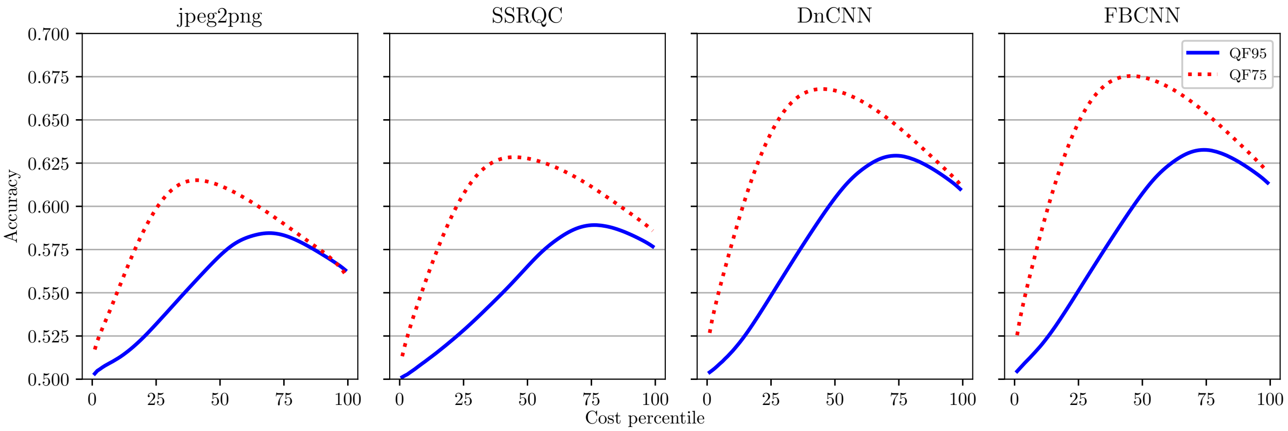


***Figure 2.*** *Accuracy of estimating the sign of quantization errors from JPEG images as a function of J-UNIWARD cost percentile. BB with JPEG quality 95.*

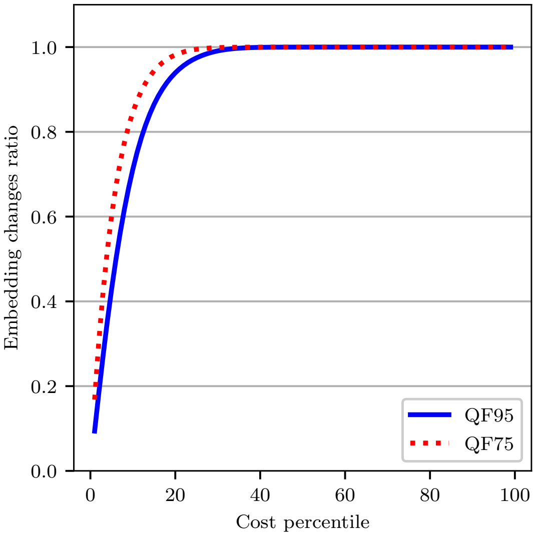


***Figure 3.*** *Fraction of embedding changes of J-UNIWARD at 0.4 bpnzac at 75 and 95 JPEG qualities as a function of the ratio of pixels with the smallest costs.*

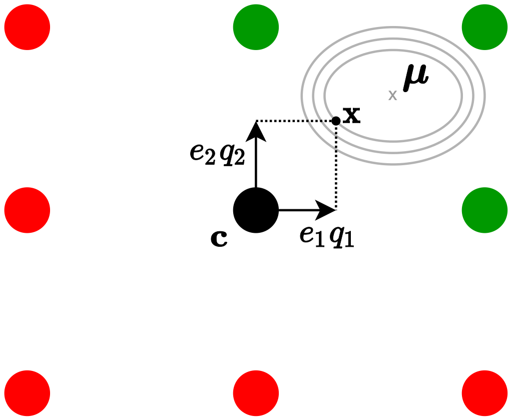


***Figure 4.*** *Example of a MVG in the DCT domain with quantization bin centers as colored dots. The symbols $\boldsymbol{\mu}$, $\mathbf{x}$, and $\mathbf{c}$, are the MVG mean, one realization of the MVG (precover), and the cover DCTs rounded to multiples of quantization steps. The cover quantization errors are $e_1q_1$ and $e_2q_2$ ($q$ being quantizations steps). Green dots mark $2^2$ directions preferred by SI-UNIWARD while the remaining $3^2 - 2^2$ dost are depicted in red.*

the bin of confusion matrix $\mathbf{A}$ at row $S$ and column $j$ is given by

$$A_{S,j} = \frac{|\{\mathbf{c} : R_d(S,\mathbf{c}) = j\}|}{|\text{BB-MS}|}, \tag{5}$$

that is, $A_{S,j}$ is the relative number of samples for which the rank of scheme $S$ in terms of distance measurement is $j$. The columns in Figure 5 are labelled by the the stegos schemes according to the empirical security ranking to aid in the comparison between distance measure and detectability.

Note that the ranking reported in Table 3 remains the same for BB-MS since it is a smaller version of BB1 but with multiple samples for the same scene. The ranks based on the Mahalanobis distance match the ordering of the stego algorithms in terms of empirical security. In other words, the cover likelihood does capture the security as evaluated by real life detectors. In contrast, measuring with the L2 distance does not capture detectability correctly, which can be seen for QF95. In particular, it does not correctly predict the security of SIEp(*DnCNN*) and SI-UNI for QF95 and QF98.

In general, the Mahalanobis distance tends to be less certain where L2 totally fails. For example, L2 distance completely misranks for QF98, while the Mahalanobis distance just becomes less certain (less clear staircase pattern). For QF95, the Mahalanobis distance clearly yields better match than L2. This indirectly confirms our claim that one needs to consider the embedding changes in their entirety w.r.t. the model of content complexity of natural images to explain the mechanism behind the benefit of using deblocked images for cost polarization.

### Dequantizing with Wiener filter

In this section, we investigate another deblocker for SIEp in the artificial datasets which is based entirely on the known cover model instead of being data driven. Adopting a uniform model for the quantization noise in the DCT domain, we can dequantize using a Wiener filter in the spatial domain since for the artificial dataset we do have a known cover model. Curiously, this purely model based deblocker gives very close performance as the data driven deep learning deblockers.

Keeping the same notation as above, given a cover JPEG $\mathbf{c}$,[5] we wish to estimate the spatial representation of the precover $\mathbf{x}^{(p)}$, $\mathbf{x}^{(p)} = \mathbf{D}^T\mathbf{x}$, where $\mathbf{D}^T \in \mathbb{R}^{64\times 64}$ is the matrix facilitating the inverse DCT and $\mathbf{x}$ is the precover of $\mathbf{c}$, which we write as

$$\mathbf{x} = \mathbf{c} + \mathbf{f}, \tag{6}$$

where $f_i = q_i e_i \in (-q_i/2, q_i/2]$ is the quantization error. Applying the inverse DCT

$$\mathbf{x}^{(p)} = \mathbf{D}^T\mathbf{x} = \mathbf{D}^T\mathbf{c} + \mathbf{D}^T\mathbf{f} \tag{7}$$

with $\mathbf{D}^T\mathbf{f}$ being the JPEG quantization error represented in the pixel domain. Since for the artificial dataset $\mathbf{x}^{(p)}$ follows a known model $\mathbf{x}^{(p)} \sim \mathcal{N}(\boldsymbol{\mu}^{(p)}, \boldsymbol{\Sigma}^{(p)})$, we can use a Wiener filter to estimate the spatial representation of the precover $\mathbf{x}^{(p)}$ provided the covariance of $\mathbf{D}^T\mathbf{f}$ is available. To this end, we assume the JPEG quantization error $e_i \sim \mathcal{U}(-q_i/2, q_i/2)$ with $e_i$ mutually independent with covariance matrix $\boldsymbol{\Xi} = \frac{1}{12}\text{diag}(q_1^2, \ldots, q_{64}^2)$. The covariance of $\mathbf{D}^T\mathbf{f}$ is thus $\boldsymbol{\Xi}^{(p)} = \mathbf{D}^T\boldsymbol{\Xi}\mathbf{D}$.

The Wiener filter estimate of $\mathbf{x}^{(p)}$ is

$$\hat{\mathbf{x}}^{(p)} = \mathbf{W}(\mathbf{D}^T\mathbf{c} - \boldsymbol{\mu}^{(p)}), \tag{8}$$

where $\mathbf{W} = \boldsymbol{\Sigma}^{(p)}(\boldsymbol{\Sigma}^{(p)} + \boldsymbol{\Xi}^{(p)})^{-1}$. Note that prior to applying the Wiener filter, we had to subtract the mean $\boldsymbol{\mu}^{(p)}$ from $\mathbf{D}^T\mathbf{c}$ to make it zero mean.

Table 4 shows the detection error of SRNet for SIEp(*Wie*) with SIEp(*DnCNN*) and three deblockers all based on the Wiener filter for the artificial datasets, J-UNIWARD, and payload 0.4 bpnzac. Wie, WieE, and WieER correspond to implementations of the Wiener filter with the exact parameters $\boldsymbol{\mu}^{(p)}, \boldsymbol{\Sigma}^{(p)}$, estimated from the decompressed JPEG from a local $3\times 3$ window, and estimated from the uncompressed image from a local $3\times 3$ window. As expected, the more accurate the model of the signal that is being dequantized, the better the performance of SIEp. In particular, with perfect modeling knowledge (Wie) for quality 95 and the noisier dataset BB1, the deblocker SIEp(*Wie*) achieves slightly better performance than SIEp(*DnCNN*), which is unaware of the model but requires training on a large dataset.

The deblocker based on the Wiener filter is very different from a deep learning deblocker in the sense that the former cannot consider relationships across $8\times 8$ blocks in contrast to the DnCNN. We take a look at this more closely in the next section.

Encouraged by the success of the Wiener dequantizer, we next inspect its performance on the real BB dataset. The results are summarized in Table 5. Since for real images the exact precover model is not known, only SIEp(*WieE*) and SIEp(*WieER*) can be evaluated. In general, SIEp with the WieE dequantizer is very close to SSRQC in terms of empirical security (which is also the case on artificial datasets). Interestingly, having access to uncompressed images for model estimation yields only a slight improvement. We also observe the same behavior for the smaller payload 0.2 bpnzac with SIEp(*WieER*) being slighter better than SIEp(*WieE*).

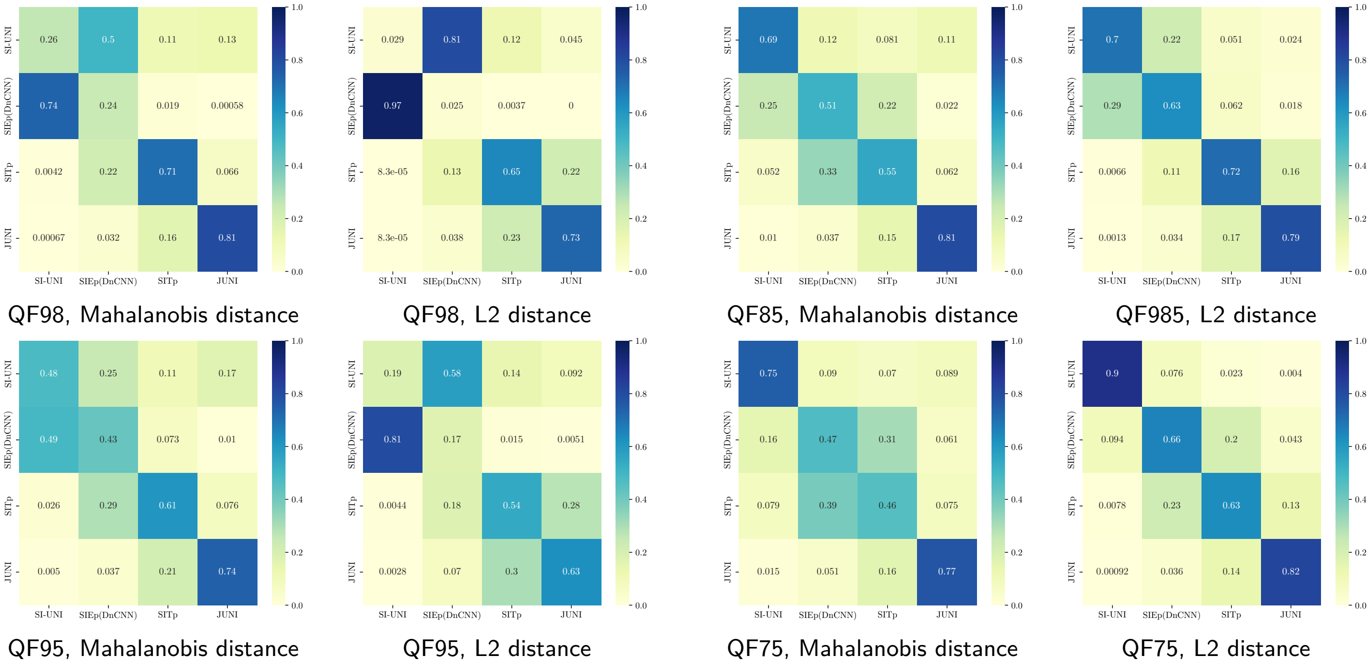
## Interblock Relationships

The cost modulation in SIEp (1) itself provides some insight into why the asymmetric costs improve security. First, the deblocked image is smoother than the cover image as it should be by the nature of what a deblocker does. The cost modulation then exerts pressure for the embedding to make changes that make the resulting stego image smoother. We quantify this using the so-called blockiness. Formally, for a grayscale $N_1 \times N_2$ image $\mathbf{z}$ represented in pixel domain with pixel values $z_{ij} \in \{0, 1, \ldots, 255\}$, $N_1, N_2$ multiples of 8 and pixel indexing starting from 1, the blockiness is

[5] We remind the reader that $\mathbf{c}$ are the dequantized DCT coefficients, or equivalently, $c_i$ is an integer multiple of $q_i$.

| Dataset | QF | J-UNI | SITp | SIEp(*SSRQC*) | SIEp(*DnCNN*) | SI-UNI |
|---|---|---|---|---|---|---|
| BB1 | 95 | 0.2737 | 0.3184 | 0.3862 | 0.3894 | 0.4202 |
| | 75 | 0.0838 | 0.1288 | 0.2195 | 0.2764 | 0.3337 |
| BB1/2 | 95 | 0.1963 | 0.2423 | 0.2986 | 0.3622 | 0.3772 |
| | 75 | 0.0363 | 0.0652 | 0.0936 | 0.1825 | 0.2713 |

**Table 3. Detection error $P_{\mathrm{E}}$ of SRNet on various steganographic schemes for payload 0.4 bpnzac on BB1 and BB1/2 for two quality factors.**



QF98, Mahalanobis distance

QF98, L2 distance

QF85, Mahalanobis distance

QF985, L2 distance

QF95, Mahalanobis distance

QF95, L2 distance

QF75, Mahalanobis distance

QF75, L2 distance

***Figure 5.*** *Confusion matrices for multiple quality factors. The $x$-axis shows the order of the embedding algorithms in terms of security (left is the most secure, right is the least). For the Mahalanobis distance, $y$-axis shows the average order in terms of $d_{\mathbf{\Sigma}}(\mathbf{s},\boldsymbol{\mu}) - d_{\mathbf{\Sigma}}(\mathbf{c},\boldsymbol{\mu})$. For L2 distance, $y$-axis shows the average order in terms of $||\mathbf{s}-\boldsymbol{\mu}||_2 - ||\mathbf{c}-\boldsymbol{\mu}||_2$. The number in each cell is the fraction of cases when the order of this cell matches the true order. The ranking is averaged across all images from BB-MS.*

| Dataset | QF | SIEp(*DnCNN*) | SIEp(*Wie*) | SIEp(*WieE*) | SIEp(*WieER*) |
|---|---|---|---|---|---|
| BB1 | 95 | 0.3894 | 0.3977 | 0.3256 | 0.3476 |
| | 75 | 0.2764 | 0.2139 | 0.1756 | 0.1985 |
| BB1/2 | 95 | 0.3622 | 0.3397 | 0.2979 | 0.2982 |
| | 75 | 0.1825 | 0.0971 | 0.0839 | 0.0888 |

**Table 4. Detection error $P_{\mathrm{E}}$ with SRNet for the Wiener family of SIEp algorithms in artificial datasets, J-UNIWARD, 0.4 bpnzac. Wie, WieE, and WieER correspond to deblockers implemented with the exact knowledge of the mean and variance $\boldsymbol{\mu}^{(p)}, \mathbf{\Sigma}^{(p)}$, estimated from the decompressed JPEG, and from the uncompressed cover.**

| QF | J-UNI | SI-JUNI | SITp | SIEp(*SSRQC*) | SIEp(*DnCNN*) | SIEp(*FBCNN*) | SIEp(*WieE*) | SIEp(*WieER*) |
|---|---|---|---|---|---|---|---|---|
| 98 | 0.1468 | **0.3118** | 0.1855 | 0.2265 | 0.2237 | 0.2569 | 0.2103 | 0.2013 |
| 95 | 0.1672 | **0.3234** | 0.2022 | 0.2409 | 0.2775 | 0.2886 | 0.2291 | 0.2374 |
| 85 | 0.0783 | **0.2561** | 0.1051 | 0.1348 | 0.17 | 0.1719 | 0.1138 | 0.1301 |
| 75 | 0.0557 | **0.2481** | 0.0783 | 0.1083 | 0.1469 | 0.1439 | 0.0973 | 0.1023 |

**Table 5. Detectability of various versions of SIEp in terms of $P_{\mathrm{E}}$ for SRNet on BB, J-UNIWARD, 0.4 bpnzac.**

$$B(\mathbf{z}) = \frac{1}{255}\left(\sum_{i=1}^{N_1}\sum_{l=1}^{N_2/8-1}|z_{i,8l} - z_{i,8l+1}|+ \right.$$
$$\left. + \sum_{k=1}^{N_1/8}\sum_{j=1}^{N_2}|z_{8k,j} - z_{8k+1,j}|\right) \quad (9)$$

Figure 6 proves this statement by showing the histogram of differences between the blockiness of the stego and the blockiness of the corresponding cover image over a dataset. Note that SIEp increases the blockiness less than J-UNIWARD. Also, for SIEp(*DnCNN*), the difference between the stego and cover blockiness is the least biased (approximately centered around 0). While J-UNIWARD stego images are always blockier than for the side-informed counterparts, making them too smooth (the case of SIEp(*jpeg2png*)) in the end makes the embedding more detectable.

Next, we analyze the dependencies among embedding changes between adjacent blocks. Figure 7 shows the covariance matrix between the sign of the estimated rounding error (2) of all 64 DCT coefficients (scanned by rows) and the sign of the estimated error of 64 coefficients from a horizontally and vertically adjacent blocks. Notice that signs of errors of DCT modes with even horizontal frequencies exhibit a positive correlation with errors from horizontal frequencies from a horizontally neighboring blocks while a negative correlation exists between odd horizontal frequencies. This confirms the tendency of the embedding to preserve smoothness between blocks and decreases the blockiness that would otherwise be increased in a scheme with symmetric costs. A similar (complementary) observation can be made about vertical frequencies between vertically adjacent blocks.

## Deblockers

This section describes the deblockers used in this paper together with all information needed for their implementation. We also contrast their performance in terms of the PSNR between the uncompressed image and the deblocked image and in terms of the accuracy of predicting the signs of quantization errors. We note that all deblockers are given a decompressed JPEG image, which is not rounded to integers. They also output non-rounded images.

For faster convergence, the two deep learning deblockers (DnCNN and FBCNN) were forced to output within the dynamic range [0, 255]. DnCNN was also trained with L2 and L1 loss, and in both the spatial and DCT domains to see if there is a benefit for security (which was not observed). FBCNN was trained only with L1 loss as suggested in the original paper. The batch size is set to 16. The learning rate is controlled with Cosine Annealing scheduler varying it from $10^{-3}$ to $10^{-8}$ for DnCNN and from $10^{-4}$ to $1.25 \times 10^{-5}$ for FBCNN. FBCNN is also seeded with weights given in `https://github.com/jiaxi-jiang/FBCNN` to exploit pre-trained attention layer. Both DnCNN and FBCNN are trained for each quality factor separately on two NVIDIA TITAN RTX GPUs. Each training of deblocker takes approximately three hours to complete.

SSRQC deblocker was downloaded from `https://github.com/coolbay/Image-deblocking-SSRQC` and used without modifications. The jpeg2png deblocker was downloaded from `https://github.com/victorvde/jpeg2png` and modified to support grayscale images.

The deblocking performance is reported in Table 6 with PSNR and sign accuracy computed over the TST set. The state-of-the-art FBCNN deblocker yields the best results for a wide range of quality factors. We also give a reference number in column “Cover” for the PSNR between uncompressed and cover JPEG images decompressed to the spatial domain without rounding. Both Wiener $3 \times 3$ and jpeg2png deblockers smooth the images too harshly, which results in a smaller PSNR.

## BACKPACK

In this section, we provide more details on how BACKPACK was implemented and used in this paper. To decrease the computational cost, it was used with only two global attack iterations. The detector the attack is performed against is J-XuNet [40]. The J-XuNet is re-trained after each iteration on covers and composition of stegos, where the best stego for each cover is selected from the past iterations according to the “minmax” strategy. During the attack, the algorithm samples multiple stego images with modified costs to reduce the noise in the gradients. We set the number of stego samples to 20 to fit within the GPU memory. For each image, the number of attacking iterations was set to 500 to ensure a higher chance of success. The initial temperature used for soft stego generation is set to 5.

It is also worth mentioning that the attack is quite computationally demanding. To complete two iterations for BB QF95, it takes 8 hours with 8 NVIDIA TITAN RTX GPUs, not counting the time needed to train the J-XuNet. For BB QF75, the attack time is 10 hours with 10 threads utilizing 2 NVIDIA TITAN RTX GPUs for the first iteration and 11 hours with 22 threads utilizing 3 NVIDIA TITAN RTX GPUs and 3 NVIDIA GeForce RTX 2080 Ti GPUs for the second iteration.

## Conclusions

Simple ideas that are powerful are always significant even if they are heuristic because they show the way and point out existence of new phenomena and connections. The SIEp method for polarizing costs with a dequantized cover is a prime example of this. In this paper, we shed some light on the inner workings of this approach. Our explanation starts with a critique. While it is true that better deblockers (in terms of PSNR) lead to more secure steganography, even the best deep learning deblockers are only slightly better in predicting the polarities of true rounding errors where it matters – for the smallest costs where the vast majority of embedding changes are executed. Instead, we argue that the key aspect is the fact that the polarities are determined from a real dequantized image. After all, from previous research we know that

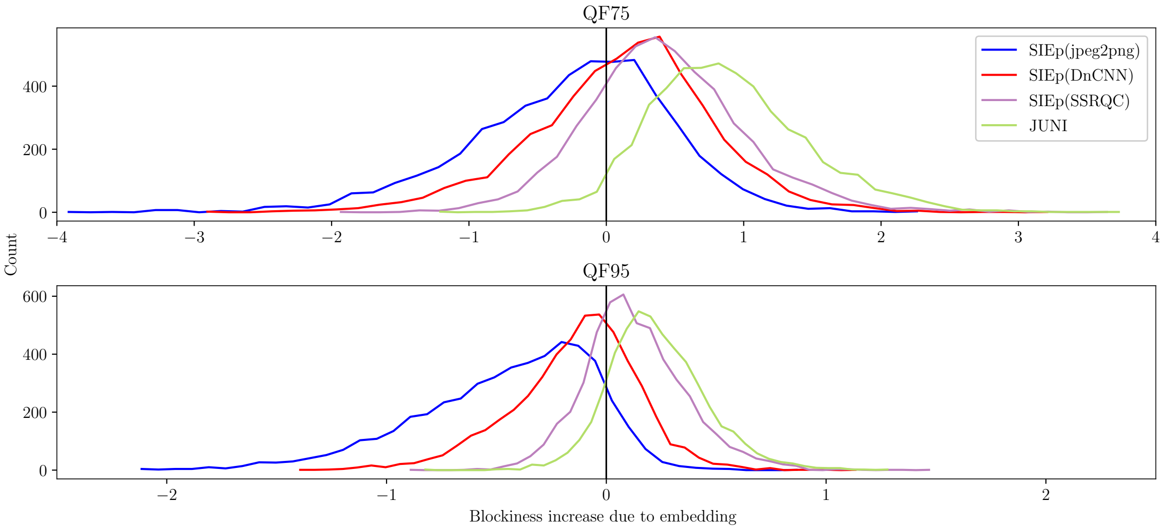
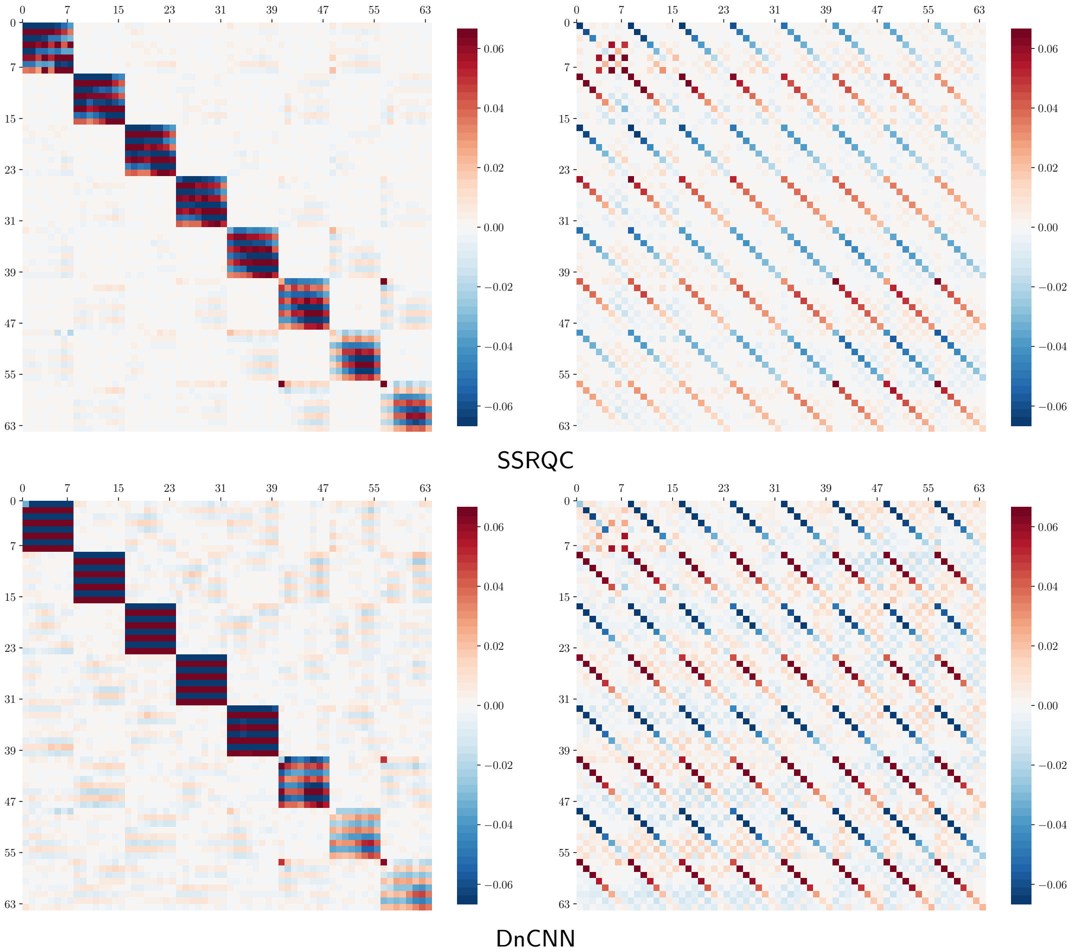


***Figure 6.*** *Histogram of blockiness differences between stego images and their cover counterparts from BB TST set for JPEG quality 95 and 75. Top: QF 75; Bottom QF 95. The vertical line at $x = 0$ is added for better visualization.*



SSRQC

DnCNN

***Figure 7.*** *Covariance between the sign of the estimated rounding error $\hat{e}_i$ from two neighboring $8 \times 8$ blocks adjacent horizontally (left) and vertically (right) for the SSRQC deblocker (top) and DnCNN (bottom). JPEG quality 95. The covariance is computed over $2 \times 2$ DCT block window ($16 \times 16$ DCT coefficients) across all images in BB TST. The covariance matrix is thresholded for a better contrast.*

| Metric | QF | Cover | Wiener $3\times3$ | jpeg2png | SSRQC | DnCNN | FBCNN |
|---|---|---|---|---|---|---|---|
| PSNR, dB | 98 | 51.963 | 34.214 | 47.444 | 52.085 | **52.95** | 52.937 |
| | 95 | 45.561 | 34.182 | 43.023 | 46.062 | 47.097 | **47.165** |
| | 85 | 38.925 | 33.917 | 38.047 | 39.551 | 40.824 | **40.914** |
| | 75 | 36.43 | 33.531 | 36.005 | 37.109 | 38.335 | **38.453** |
| Sign accuracy | 98 | N/A | 0.5518 | 0.547 | 0.569 | **0.5893** | 0.588 |
| | 95 | N/A | 0.5617 | 0.5597 | 0.5734 | 0.6057 | **0.6093** |
| | 85 | N/A | 0.5678 | 0.5651 | 0.5843 | 0.6137 | **0.6196** |
| | 75 | N/A | 0.5642 | 0.5581 | 0.5841 | 0.6108 | **0.6171** |

**Table 6. Performance of various JPEG deblockers in terms of their PSNR w.r.t. the uncompressed image and in terms of accuracy of predicting the signs of quantization errors.**

when another JPEG exposure of the same scene is available to the sender, the polarities learned from it provide a significant security boost [10]. The final piece of evidence (and critique) is the fact that a deblocker that predicts the rounding error polarities perfectly does not perform well with the constant cost modulation factor of SIEp.

To gain insight and prove our point, we move from a real dataset to an artificial dataset by introducing a model of content complexity in the pixel domain. Porting it into the DCT domain, we use the log cover likelihood (the Mahalanobis distance) as a scalar measure of performance. This measure considers the covariance structure of the cover model in the DCT domain, and it closely correlates with empirical detectability as established by machine-learning detectors. In particular, it predicts that SIEp with true rounding error polarities will not perform well. For some quality factors on the artificial dataset a simple Wiener filter achieves equal detection performance as data-driven deblockers implemented as deep convolutional neural networks. Additionally, we inspect the impact of cost polarization from a deblocked image in terms of blockiness and inter-block correlations of embedding changes in specific DCT modes. The behavior we see is consistent with the recent findings determined from models [29, 37].

## Acknowledgments

Special thanks go to Jan Butora for helping start this research before he left Binghamton University. The work on this paper was supported by NSF grant No. 2028119.

## References

[1] P. Bas. Steganography via cover-source switching. In *Proceedings IEEE Workshop on Information Forensics and Security (WIFS)*, Abu Dhabi, UAE, December 4-7, 2016.

[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] S. Bernard, P. Bas, T. Pevný, and J. Klein. Optimizing additive approximations of non-additive distortion functions. In D. Borghys and P. Bas, editors, *The 9th ACM Workshop on Information Hiding and Multimedia Security*, pages 105–112, Brussels, Belgium, June 22–25, 2021.

[5] 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.

[6] 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.

[7] M. Boroumand, J. Fridrich, and R. Cogranne. Are we there yet? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2019*, San Francisco, CA, January 14–17, 2019.

[8] J. Butora, Y. Yousfi, and J. Fridrich. How to pretrain for steganalysis. In D. Borghys and P. Bas, editors, *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, 2021. ACM Press.

[9] T. Denemark and J. Fridrich. Improving steganographic security by synchronizing the selection channel. In J. Fridrich, P. Comesana, and A. Alattar, editors, *3rd ACM IH&MMSec. Workshop*, Portland, Oregon, June 17–19, 2015.

[10] T. Denemark and J. Fridrich. Steganography with multiple JPEG images of the same scene. *IEEE Transactions on Information Forensics and Security*, 12(10):2308–2319, 2017.

[11] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 20–25, 2009.

[12] T. Filler and J. Fridrich. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4):705–720, 2010.

[13] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, September 2011.

[14] J. Fridrich. Feature-based steganalysis for JPEG im-

ages and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.

[15] Q. Giboulot, P. Bas, and R. Cogranne. Multivariate side-informed Gaussian embedding minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 17:1841–1854, 2022.

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

[17] L. Guo, J. Ni, and Y.-Q. Shi. An efficient JPEG steganographic scheme using uniform embedding. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[18] 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.

[19] V. Holub. *Content Adaptive Steganography – Design and Detection.* PhD thesis, Binghamton University, May 2014.

[20] 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.

[21] 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.

[22] X. Hu, J. Ni, and Y.Q. Shi. Efficient JPEG steganography using domain transformation of embedding entropy. *IEEE Signal Processing Letters*, 25(6):773–777, 2018.

[23] 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.

[24] J. Jiang, K. Zhang, and R. Timofte. Towards flexible blind JPEG artifacts removal. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4977–4986, 2021.

[25] A. D. Ker and T. Pevnyý. A mishmash of methods for mitigating the model mismatch mess. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 1601–1615, San Francisco, CA, February 3–5, 2014.

[26] 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.

[27] 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.

[28] W. Li, K. Chen, W. Zhang, H. Zhou, Y. Wang, and N. Yu. JPEG steganography with estimated side-information. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):2288–2294, 2020.

[29] W. Li, W. Zhang, K. Chen, W. Zhou, and N. Yu. Defining joint distortion for JPEG steganography. In R. Böhme and C. Pasquini, editors, *The 6th ACM Workshop on Information Hiding and Multimedia Security*, Innsbruck, Austria, June 20–22, 2018. ACM Press.

[30] I. Lubenko and A. D. Ker. Steganalysis with mismatched covers: Do simple classifiers help? In J. Dittmann, S. Katzenbeisser, and S. Craver, editors, *Proc. 13th ACM Workshop on Multimedia and Security*, pages 11–18, Coventry, UK, September 6–7, 2012.

[31] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, December 1999.

[32] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme and R. Safavi-Naini, editors, *Information Hiding, 12th International Conference*, volume 6387 of Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.

[33] T. Pevný and A. D. Ker. Exploring non-additive distortion in steganography. In R. Böhme and C. Pasquini, editors, *The 6th ACM Workshop on Information Hiding and Multimedia Security*, Innsbruck, Austria, June 20–22, 2018. ACM Press.

[34] 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.

[35] M. Sharifzadeh, M. Aloraini, and D. Schonfeld. Adaptive batch size image merging steganography and quantized Gaussian image steganography. *IEEE Transactions on Information Forensics and Security*, 15:867–879, 2020.

[36] W. Su, J. Ni, X. Li, and Y.Q. Shi. A new distortion function design for JPEG steganography using the generalized uniform embedding strategy. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(12):3545–3549, 2018.

[37] T. Taburet, P. Bas, W. Sawaya, and R. Cogranne. JPEG steganography and synchronization of DCT coefficients for a given development pipeline. In C. Riess, editor, *The 8th ACM Workshop on Information Hiding and Multimedia Security*, Denver, June 22–25, 2020. ACM Press.

[38] T. Taburet, P. Bas, W. Sawaya, and J. Fridrich. A natural steganography embedding scheme dedicated to color sensors in the JPEG domain. In A. Alattar

and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2019*, San Francisco, CA, January 14–17, 2019.

[39] Z. Wang, Z. Yin, and X. Zhang. Asymmetric distortion function for JPEG steganography using block artifact compensation. *International Journal of Digital Crime Forensics*, 11(1):90–99, January 2019.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.

[46] C. Zhao, J. Zhang, S. Ma, X. Fan, Y. Zhang, and W. Gao. Reducing image compression artifacts by structural sparse representation and quantization constraint prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(10):2057–2071, 2017.

## Author Biography

*Edgar Kaziakhmedov received M.S. degree in Applied Mathematics and Physics from Moscow Institute of Physics and Technology in 2020. He is currently pursuing PhD degree in electrical engineering at Binghamton University. His research areas lie within digital image steganalysis and steganography, neural network based image processing and digital media forensics.*

*Eli Dworetzky is currently pursuing his PhD in electrical and computer engineering at Binghamton University. His research currently focuses on image steganography and steganalysis. He received an MS in computer engineering from Binghamton University in 2021.*

*Yassine Yousfi received his PhD from the Electrical and Computer Engineering at Binghamton University in 2022. His research areas are digital multimedia security and forensics, particularly steganography and steganalysis of digital images. He also received an MS in Machine Learning from Ecole Centrale de Lille in France in 2018.*

*Jessica Fridrich is Distinguished Professor of Electrical and Computer Engineering at Binghamton University. She received her PhD in Systems Science from Binghamton University in 1995 and MS in Applied Mathematics from Czech Technical University in Prague in 1987. Her main interests are in steganography, steganalysis, and digital image forensics. Since 1995, she has received 22 research grants totaling over $13 mil that lead to more than 220 papers and 7 US patents.*