Matrix Embedding With Pseudorandom Coefficient Selection and Error Correction for Robust and Secure Steganography

Anindya Sarkar, *Member, IEEE*, Upamanyu Madhow, *Fellow, IEEE*, and B. S. Manjunath, *Fellow, IEEE*

***Abstract*—In matrix embedding (ME)-based steganography, the host coefficients are minimally perturbed such that the transmitted bits fall in a coset of a linear code, with the syndrome conveying the hidden bits. The corresponding embedding distortion and vulnerability to steganalysis are significantly less than that of conventional quantization index modulation (QIM)-based hiding. However, ME is less robust to attacks, with a single host bit error leading to multiple decoding errors for the hidden bits. In this paper, we employ the ME-RA scheme, a combination of ME-based hiding with powerful repeat accumulate (RA) codes for error correction, to address this problem. A key contribution of this paper is to compute log likelihood ratios for RA decoding, taking into account the many-to-one mapping between the host coefficients and an encoded bit, for ME. To reduce detectability, we hide in randomized blocks, as in the recently proposed Yet Another Steganographic Scheme (YASS), replacing the QIM-based embedding in YASS by the proposed ME-RA scheme. We also show that the embedding performance can be improved by employing punctured RA codes. Through experiments based on a couple of thousand images, we show that for the same embedded data rate and a moderate attack level, the proposed ME-based method results in a lower detection rate than that obtained for QIM-based YASS.**

***Index Terms*—Log likelihood ratio (LLR), matrix embedding (ME), repeat-accumulate coding, steganalysis, steganography.**

I. Introduction

STEGANOGRAPHY is the art of secure communication where the existence of the communication itself cannot be detected while steganalysis is the art of detecting the secret communication. The requirements for a "good" steganographic scheme are a high embedding capacity, while remaining statistically secure. When there is an active adversary in the transmission channel, the hiding scheme is considered to be an example of "active steganography." Various examples of the active warden scenario (active steganography) are in [4], [8], [10], [16], and [21]. In our problem formulation, we assume that the "active warden" may modify the transmitted signal which may (or may not) contain the embedded message—the signal being the stego (or cover) image. When the stego image is modified by the "active warden," a secure stego scheme is effective only if the embedded message can be recovered.

Matrix embedding (ME) [7], an embedding method with a high embedding efficiency (*number of bits embedded per unit embedding distortion*), is generally used for passive steganography as the lower embedding distortions (as compared to embedding methods like quantization index modulation (QIM) [5], while both methods have the same embedded data rate) make the hiding less detectable. Here, we use ME for active steganography by combining it with powerful error correction codes. We review ME using an example.

ME Example: Consider (7,3) ME, in which 3 data bits are embedded in 7 host bits. The idea is to perturb the host bits minimally so that they fall in the coset of a linear code, whose syndrome equals the data bits to be hidden. In particular, we consider the (7,4) Hamming code with parity check matrix $\mathbf{H}$ which is expressed as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{1}$$

For a host sequence $\mathbf{a} = (1,0,0,1,0,0,1)$, the syndrome $\mathbf{b}'$ is obtained as $(\mathbf{b}')^T = \mathbf{H}(\mathbf{a})^T = (0,1,0)^T$, where the operations are performed over the binary field. If the data bits to be embedded are (0,1,0), we can send the host bits without perturbation. However, suppose that we wish to embed (0,0,0). The aim is to find $\mathbf{\Delta a}$, the perturbation vector for $\mathbf{a}$, with the lowest Hamming weight. Then, $\mathbf{H}(\mathbf{a})^T + \mathbf{H}(\mathbf{\Delta a})^T = (0,0,0)^T$. Therefore, $\mathbf{H}(\mathbf{\Delta a})^T = (0,1,0)^T$. If only the ith element in $\mathbf{\Delta a}$ is 1, then $\mathbf{H}(\mathbf{\Delta a})^T$ equals the ith column in $\mathbf{H}$. The second column in $\mathbf{H} = (0,1,0)^T$. Therefore, $\mathbf{\Delta a} = (0,1,0,0,0,0,0)$. Similarly, to embed the data bits (1,1,1), the perturbation $\mathbf{\Delta a}$ is such that $\mathbf{H}(\mathbf{a})^T + \mathbf{H}(\mathbf{\Delta a})^T = (1,1,1)^T \Rightarrow \mathbf{H}(\mathbf{\Delta a})^T = (1,0,1)^T$. Since the fifth column of $\mathbf{H} = (1,0,1)^T$, $\mathbf{\Delta a} = (0,0,0,0,1,0,0)$. Similarly, for embedding any three-tuple, we need to change at most one host bit (Hamming weight of $\mathbf{\Delta a} \leq 1$), which explains why ME is so powerful for passive warden steganography.

Comparison With QIM: Given the popularity of QIM embedding [5], [27], it is the natural benchmark against which to compare new steganographic methods. Scalar QIM in the context of JPEG stego schemes corresponds to rounding a discrete cosine transform (DCT) coefficient to the nearest even or odd quantization level, depending on the embedded bit. Thus, the DCT

Manuscript received July 28, 2009; revised February 02, 2010; accepted February 07, 2010. Date of publication March 22, 2010; date of current version May 14, 2010. The work of A. Sarkar was supported by ONR Grant #N00014-05-1-0816 and Grant #N00014-10-1-0141. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jessica J. Fridrich.

The authors are with the Vision Research Laboratory, Department of Electrical and Computer Engineering, University of California Santa Barbara, Santa Barbara, CA, 93106 USA (e-mail: anindya@ece.ucsb.edu; madhow@ece.ucsb.edu; manj@ece.ucsb.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2046218

coefficient is changed with probability half, relative to rounding to the nearest quantizer level as done during JPEG compression. Therefore, on average, the embedding efficiency is 2 bits per changed coefficient. For (7,3) ME, we embed 3 bits by modifying one of 7 coefficients resulting in a higher embedding efficiency of 3. Given a set of 7 embedding coefficients, QIM and ME schemes can embed 7 and 3 bits, respectively, so that QIM achieves a higher hiding rate for the same number of hiding locations. Also, a single error affects the decoding of only a single bit for QIM. For (7,3) ME, changing a single coefficient (e.g., flipping the seventh bit of $\mathbf{a}$ in the above example) can lead to three bit errors, so that ME is more fragile than QIM for active steganography. Thus the regime in which ME can outperform QIM is when the required hiding rate is low enough so that 1) the lower hiding rate of ME is not an issue; 2) also, the channel attacks should be of moderate strength so that 3) the errors can be rectified through powerful error correction codes. This is the regime explored in this paper.

Contributions: Our main contributions are outlined below.

1) We impart noise robustness to the ME-based active steganographic framework by performing error correction coding (ECC) on the data bits to generate the code bits embedded using ME. For maintaining perceptual transparency, hiding in coefficients of small magnitudes is avoided. This is interpreted as erasures at the encoder, as in our prior work on QIM-based hiding [27]. For ECC, we have used repeat accumulate (RA) codes [9], which are well suited for such high erasure channels [27]. We also show that *punctured* [2], [3] RA codes perform better than the original RA codes.
2) For iterative decoding, the decoder needs to be initialized with proper confidence values (log likelihood ratios or LLRs) at the embedding locations. Even if one out of 7 host coefficients is erased for (7,3) ME, it can affect the decoding at 3 bit locations—the RA decoder needs to consider the various erasure-related cases before computing the soft weights to initialize the iterative decoder. *Thus, a key contribution of the paper is to work through the combinatorics required to determine the soft decisions.*
3) To reduce detectability, we use Yet Another Steganographic Scheme (YASS), the hiding framework proposed in [28]. Here, the hiding blocks are pseudorandomly chosen to desynchronize the self-calibration scheme, which assumes that the hiding is done in regular 8 × 8 blocks of DCT coefficients, used in standard steganalysis methods [22], [23]. ME-based YASS is shown to be less detectable than QIM-based YASS for the same steganalysis schemes for the same (low) data rate.

To summarize the roles of the various modules, YASS suggests "where" to embed (randomized block locations), ME shows "how" to embed (embedding method), and the RA-based ECC framework determines "what" gets embedded (it generates the code bits, given the information bits)—this is illustrated in Fig. 1.

Related Work: Since the concept of ME was first introduced by Crandall [7], there have been a number of papers describing the properties and improvements of ME *for passive steganography* [11], [15], [17], [29]. The F5 algorithm [29] is a secure JPEG steganographic scheme based on ME. F5 uses $(1,n,k)$ coding based on binary Hamming codes, where $n=2^k-1$, and k message bits are to be embedded in n image DCT coefficients, by changing the least significant bit (LSB) of (at most) one of the n quantized DCT coefficients after rounding. This method was further generalized by Kim *et al.* [17]—(t,n,k) coding methodology is suggested where while embedding k-bits into n host bits $(n=2^k-1)$, at most t $(t \geq 1)$ embedding changes can be made. It is useful when the single coefficient to be perturbed in $(1,n,k)$ scheme cannot be modified due to various constraints (e.g., due to erasures). Approaches to increase the embedding rate for ME based on random linear codes of small dimension, and simplex codes, have been presented by Fridrich *et al.* [15]. An interesting approach for minimal distortion steganography is perturbed quantization (PQ) [12], which is highly secure from modern detectors [14]. Another contribution of the PQ scheme is the use of wet paper codes (WPC) which allows sending the same number of bits, on an average, as would have been possible if the actual set of hiding locations were known. In [11], the embedding impact profile of all the pixels is estimated and an optimal trade-off is studied between the number of embedding changes and their amplitudes, while using ME.

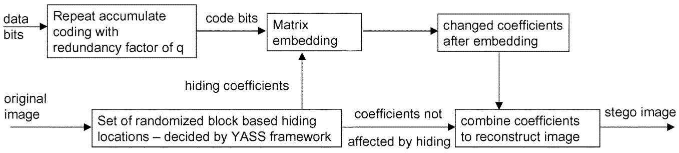
data bits
Repeat accumulate coding with redundancy factor of q
code bits
Matrix embedding
changed coefficients after embedding
hiding coefficients
original image
Set of randomized block based hiding locations – decided by YASS framework
coefficients not affected by hiding
combine coefficients to reconstruct image
stego image

Fig. 1. Entire hiding framework using RA coding, ME, and YASS-based coefficient selection.

While the ME-based methods generally focus on passive steganography, our QIM-based hiding scheme YASS [28] achieved secure (using randomized block locations) and robust (using RA-codes) steganography. Due to the lower embedding efficiency, QIM schemes are more detectable than ME-based schemes like F5 [29], nonshrinkage F5 (nsF5) [14], and modified ME (MMx) [17]; *but all these ME-based methods are passive stego schemes*. Here, we combine the low detectability of ME with the robustness resulting from powerful ECC to obtain an active stego scheme that works for real-world channels such as JPEG attacks. We use data sets for which QIM-based YASS is detectable even at low embedding rates, and attempt to determine when ME outperforms QIM (we can only hope to do better than QIM at low enough embedding rates and moderate enough attacks).

Prior work that combines ECC with ME-based hiding includes [31]. However, the combination codes in [31] are used over individual blocks of image coefficients, and are, therefore, much less powerful than the RA code framework employed here, where the codeword spans the entire set of embeddable coefficients. Erasures resulting from coefficient-adaptive hiding, as well as the effect of real-world channels such as JPEG compression, are also not considered in [31].

Outline of the Paper: The YASS framework and the overall ME-based hiding system are described in Section II. We consider the (7,3) ME case. The embedding method for the **ME-RA**

TABLE I
GLOSSARY OF NOTATIONS

Notation	Definition
$\mathbf{y} = (y_1, y_2, \ldots, y_7)$	set of 7 AC DCT coefficients that are considered together to embed 3 code bits for (7,3) ME
$\mathbf{a} = (a_1, a_2, \ldots, a_7)$	set of binary symbols obtained from $\mathbf{y}$ where $a_i = \text{mod}(\text{round}(y_i), 2),\ 1 \le i \le 7$
$\mathbf{b} = (b_1, b_2, b_3)$	set of 3 bits that are embedded in $\mathbf{y} = (y_1, y_2, \ldots, y_7)$
$\mathbf{b}' = (b'_1, b'_2, b'_3)$	syndrome obtained from LSBs of $\mathbf{y}$, i.e. from $\mathbf{a}$, thus, $(\mathbf{b}')^T = \mathbf{H}(\mathbf{a})^T$ where $\mathbf{H}$ is introduced in (1).
$\mathbf{y}^e = (y^e_1, y^e_2, \ldots, y^e_7)$	set of 7 AC DCT coefficients to which $\mathbf{y}$ is changed after embedding
$\mathbf{a}^e = (a^e_1, a^e_2, \ldots, a^e_7)$	set of binary symbols to which $\mathbf{a}$ is changed after embedding, $a^e_i = \text{mod}(\text{round}(y^e_i), 2),\ 1 \le i \le 7$
$\mathbf{y}^r = (y^r_1, y^r_2, \ldots, y^r_7)$	set of received DCT elements at the decoder side corresponding to $\mathbf{y}^e$
$LLR_{final}(\mathbf{b}\|\mathbf{y}^r)$	the set of log-likelihood ratio (LLR) values for the bit locations $\mathbf{b} = (b_1, b_2, b_3)$ based on $\mathbf{y}^r$
QF_h	the design quality factor selected for hiding, which is used to generate the quantized AC DCT coefficients
QF_a	the output quality factor at which the stego image, after randomized block based hiding, is JPEG compressed
B, N_B	B is the big-block size used for YASS ($B > 8$) while N_B is the number of $B \times B$ blocks in the image
λ	the number of top AC DCT coefficients, encountered during zigzag scan, used for hiding per 8×8 block
$T \in \mathbb{R}^{3\times 3}$	the transition probability matrix between $\mathbf{y}^e$ and $\mathbf{y}^r$, where each term is mapped to one of $\{0, 1, e\}$; e denotes an erasure
bpnc	ratio of the number of data bits successfully embedded in an image to the number of non-zero (after rounding) block-based AC DCT image coefficients
QIM-RA: n terms	the QIM-RA (QIM-based YASS where RA-coding is used as ECC) scheme where the first n AC DCT elements encountered during zigzag scan per 8×8 block are used for embedding, i.e. $\lambda = n$
q_{opt}	the minimum RA code redundancy factor which allows proper decoding and recovery of all the data bits

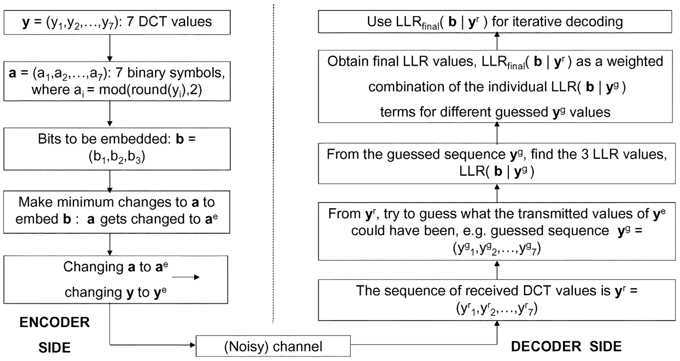
y = (y1,y2,...,y7): 7 DCT values
a = (a1,a2,...,a7): 7 binary symbols, where ai = mod(round(yi),2)
Bits to be embedded: b = (b1,b2,b3)
Make minimum changes to a to embed b : a gets changed to ae
Changing a to ae changing y to ye
ENCODER SIDE
(Noisy) channel
DECODER SIDE
The sequence of received DCT values is yr = (yr1,yr2,...,yr7)
From yr, try to guess what the transmitted values of ye could have been, e.g. guessed sequence yg = (yg1,yg2,...,yg7)
From the guessed sequence yg, find the 3 LLR values, LLR(b | yg)
Obtain final LLR values, LLRfinal(b | yr) as a weighted combination of the individual LLR(b | yg) terms for different guessed yg values
Use LLRfinal(b | yr) for iterative decoding

Fig. 2. Encoder embeds $\mathbf{b}$, a 3-bit sequence, in $\mathbf{y}$, a sequence of 7 DCT coefficients, by changing the minimum number of coefficients in $\mathbf{y}$, thus modifying it to $\mathbf{y}^e$ which is transmitted through the channel and $\mathbf{y}^r$ is the corresponding received sequence. The decoder uses $\mathbf{y}^r$ to estimate the LLR values for the 3-bit locations. The sequence $\mathbf{a}^r$, which is derived from $\mathbf{y}^r$ ($a^r_i = \text{mod}(\text{round}(y^r_i), 2),\ \forall i$), is not explicitly shown at the decoder.

scheme (using ME to embed the RA-encoded bits) is explained in Section III. The various approaches used for the initial LLR computation at the decoder are explained in Section IV. The **ME-RA-puncture** scheme, which is a refinement of the ME-RA method and produces a higher effective embedded data rate than ME-RA through "puncturing" [2], [3] (*deletion of a suitable number of encoded bits at appropriate bit locations*), is explained in Section V. Section VI compares the data rate obtained using various methods for the initial LLR computation. It also compares the performance of **ME-RA-puncture** to that of the **nonshrinkage** (which avoids the erasure induced shrinkage problem [14]) method. Comparison of the detectability against similar steganalysis methods and effective hiding rate under different noise channels, for **QIM-RA** (using QIM to embed the RA-encoded bits) and **ME-RA-puncture** methods, are presented in Section VII.

Table I introduces some notations which are frequently used in the paper. A sequence of seven terms $(y_1, y_2, \ldots, y_7)$ is referred to as $\mathbf{y}$. The complement of a bit a_i is referred to as $\overline{a_i}$.

II. Introduction to YASS and System Overview

YASS: The security of YASS [28] can be attributed to the randomized choice of hiding locations. The input image is decompressed if it is in JPEG format and then divided into blocks of size $B \times B$ ($B > 8$), where B is called the big-block size. For each big-block, an 8×8 sub-block is pseudorandomly chosen to hide data. *The encoder and decoder share the same key by which they can access the same set of* 8×8 *blocks*. For every sub-block, its 2-D DCT is computed and then divided by a JPEG quantization matrix at a *design* quality factor (QF), denoted by QF$_h$. A band of λ AC DCT coefficients lying in the low and mid-frequency range is used for hiding. After data embedding, the resultant image is JPEG compressed at a QF of QF$_a$. The embedding rate decreases, as compared to using regular 8×8 blocks, because a lower fraction $(((8 \times 8)/(B \times B)) < 1)$ of the DCT coefficients is now considered for embedding. To increase the embedding rate, we also consider multiple disjoint 8×8 blocks within a $B \times B$ block, for $B > 15$; e.g., choosing nine 8×8 blocks within a 25×25 big-block yields a higher embedding rate than choosing one 8×8 block within a 9×9 big-block. The effective big-block size B_{eff}, which accommodates one 8×8 block, is now reduced from 9 to 25/3.

System Overview: The system flow is explained in Fig. 2. Let N_B denote the number of $B \times B$ blocks obtained from the image, with λ elements used for hiding from the DCT matrix for the 8×8 pixel block pseudorandomly chosen from every $B \times B$ block. For $9 \le B \le 15$, the total number of DCT coefficients available for hiding equals $N = \lambda N_B$. When $B_{\text{eff}} < 9$ (e.g., when $B = 17, 25$, or 49), $N = N_B$(no. of 8×8 blocks in a $B \times B$ big-block)$\lambda = N_B(\lfloor B/8 \rfloor)^2\lambda$. Using (7,3) ME, the coefficients are partitioned into $\lfloor N/7 \rfloor$ sets, each having 7 terms—e.g., $\mathbf{y} = (y_1, y_2, \ldots, y_7)$ is such a set where 3 bits (b_1, b_2, b_3) are embedded. The total number of code bits equals $3\lfloor N/7 \rfloor$, and for an RA code of redundancy factor q, $\lfloor (3\lfloor N/7 \rfloor)/q \rfloor$ data bits can be accommodated. *The encoder aims to embed the code bits using ME while minimally changing the DCT coefficients* (e.g., from $\mathbf{y}$ to $\mathbf{y}^e$). The redundancy factor q is set to q_{opt} (defined in Table I) such that the hiding rate is maximized for a given set of hiding parameters and attack

TABLE II

FOR (7,3) ME, THERE ARE EIGHT POSSIBLE RELATIONS BETWEEN $\mathbf{b}$, THE 3-TUPLE OF BITS TO BE EMBEDDED, AND $\mathbf{b}'$, THE SYNDROME OBTAINED FROM $\mathbf{y}$. THERE IS ALWAYS A COEFFICIENT y_i, WHOSE MODIFICATION TO y_i^e (ENSURING $a_i^e = \overline{a_i}$, WHERE $a_i = \text{mod}(\text{round}(y_i), 2)$ AND $a_i^e = \text{mod}(\text{round}(y_i^e), 2)$) ENSURES PROPER EMBEDDING. IF $|y_i| \leq 0.5$, THE CANDIDATE 2-TUPLES/3-TUPLES FOR EMBEDDING ARE MENTIONED

Relation between $\mathbf{b}'$ and $\mathbf{b}$	Condition	1-tuple	Possible 2-tuples	Possible 3-tuples
$b_1' = b_1,\ b_2' = b_2,\ b_3' = b_3$	0	none	none	none
$b_1' = \overline{b_1},\ b_2' = b_2,\ b_3' = b_3$	1	y_1	$(y_2, y_3)(y_4, y_5)(y_6, y_7)$	$(y_2, y_4, y_7)(y_2, y_5, y_6)(y_3, y_5, y_7)(y_3, y_6, y_4)$
$b_1' = b_1,\ b_2' = \overline{b_2},\ b_3' = b_3$	2	y_2	$(y_1, y_3)(y_4, y_6)(y_5, y_7)$	$(y_1, y_4, y_7)(y_1, y_5, y_6)(y_3, y_4, y_5)(y_3, y_6, y_7)$
$b_1' = \overline{b_1},\ b_2' = \overline{b_2},\ b_3' = b_3$	3	y_3	$(y_1, y_2)(y_4, y_7)(y_5, y_6)$	$(y_1, y_5, y_7)(y_1, y_4, y_6)(y_2, y_4, y_5)(y_2, y_6, y_7)$
$b_1' = b_1,\ b_2' = b_2,\ b_3' = \overline{b_3}$	4	y_4	$(y_1, y_5)(y_2, y_6)(y_3, y_7)$	$(y_1, y_2, y_7)(y_1, y_3, y_6)(y_2, y_3, y_5)(y_5, y_6, y_7)$
$b_1' = \overline{b_1},\ b_2' = b_2,\ b_3' = \overline{b_3}$	5	y_5	$(y_1, y_4)(y_2, y_7)(y_3, y_6)$	$(y_1, y_3, y_7)(y_1, y_2, y_6)(y_2, y_3, y_4)(y_4, y_6, y_7)$
$b_1' = b_1,\ b_2' = \overline{b_2},\ b_3' = \overline{b_3}$	6	y_6	$(y_1, y_7)(y_2, y_4)(y_3, y_5)$	$(y_1, y_2, y_5)(y_1, y_3, y_4)(y_2, y_3, y_7)(y_4, y_5, y_7)$
$b_1' = \overline{b_1},\ b_2' = \overline{b_2},\ b_3' = \overline{b_3}$	7	y_7	$(y_1, y_6)(y_2, y_5)(y_3, y_4)$	$(y_1, y_2, y_4)(y_1, y_3, y_5)(y_2, y_3, y_6)(y_4, y_5, y_6)$

channel. The decoder can recover q from the RA-encoded sequence without any side information, as explained in [1], [25]. We do not consider attacks that can desynchronize the encoder and decoder (e.g., warping or cropping), so that the decoder has access to the same set of N hiding locations as the encoder. A proper initialization of the soft weights (log likelihood ratios or LLRs, explained in Section IV) for codeword locations allows faster convergence (at a lower redundancy factor) and increases the hiding rate. Various methods to compute the soft weights $(\text{LLR}_{\text{final}}(\mathbf{b}|\mathbf{y}^r))$ for 3 bit locations $(\mathbf{b} = (b_1, b_2, b_3))$ given 7 DCT terms $(\mathbf{y}^r = (y_1^r, y_2^r, \ldots, y_7^r)$, which is the noisy version of $\mathbf{y}^e)$ are studied in Section IV.

III. ME-RA EMBEDDING

The encoder embedding logic for (7,3) ME, introduced in Section I, is explained in detail here. The operations at the encoder side are outlined in Fig. 2. The sequence $\mathbf{b} = (b_1, b_2, b_3)$ is to be embedded in $\mathbf{y}$, a set of 7 locations. After rounding to the nearest integer and modulo 2 operation, the coefficients return the bit sequence $\mathbf{a} = (a_1, a_2, \ldots, a_7)$, where $a_i = \text{mod}(\text{round}(y_i), 2)$. Using the (3×7) mapping matrix $\mathbf{H}$, introduced in (1), on $\mathbf{a}$, the following syndrome $\mathbf{b}'$ is obtained: $(b_1', b_2', b_3')^T = \mathbf{H}(\mathbf{a})^T$. The aim is to make minimum changes to $\mathbf{a}$ to obtain $\mathbf{a}^e = (a_1^e, \ldots, a_7^e)$, such that $\mathbf{b}^T = \mathbf{H}(\mathbf{a}^e)^T$.

From Table II, for every condition (except condition 0), if y_i is the only element being changed (for the other elements, y_j^e is the rounded version of y_j, $j \neq i$), $a_i^e = \overline{a_i}$ and $a_j^e = a_j$, $j \neq i$. To ensure $a_i^e = \overline{a_i}$, y_i is changed to the nearest odd/even integer y_i^e, if y_i gets converted to an even/odd integer, respectively, after rounding off. It may happen that the absolute value of the coefficient to be modified $(|y_i|)$ is less than 0.5—an erasure occurs at that location. This erasure-induced problem can be solved by changing two or more of the remaining coefficients in $\mathbf{y}$ (using solution proposed in [17]). Based on the embeddable bits and the candidate coefficients, five cases (**A-E**) are possible, as shown in Fig. 3.

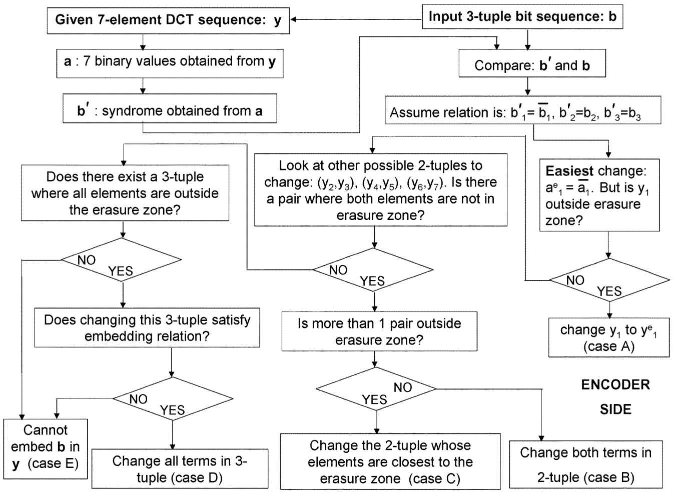
Given 7-element DCT sequence: y
Input 3-tuple bit sequence: b
a : 7 binary values obtained from y
Compare: b′ and b
b′ : syndrome obtained from a
Assume relation is: b′1= b̄1, b′2=b2, b′3=b3
Does there exist a 3-tuple where all elements are outside the erasure zone?
Look at other possible 2-tuples to change: (y2,y3), (y4,y5), (y6,y7). Is there a pair where both elements are not in erasure zone?
Easiest change: ae1 = ā1. But is y1 outside erasure zone?
NO
YES
NO
YES
NO
YES
Does changing this 3-tuple satisfy embedding relation?
Is more than 1 pair outside erasure zone?
change y1 to ye1 (case A)
NO
YES
NO
YES
ENCODER SIDE
Cannot embed b in y (case E)
Change all terms in 3-tuple (case D)
Change the 2-tuple whose elements are closest to the erasure zone (case C)
Change both terms in 2-tuple (case B)

Fig. 3. Embedding logic used at the encoder for a (7,3) ME example: cases (A)–(D) correspond to cases where embedding is possible, while case (E) is where embedding fails.

Embedding Method Involving Minimum Perturbations:

- We use Table II to find the single coefficient y_i to change to the nearest odd or even integer y_i^e, if y_i rounds off to an even or odd integer, respectively. If that element (e.g., y_1 in Fig. 3) can indeed be modified, it corresponds to **Case A**.
- If that single coefficient (y_i) lies in the erasure zone (−0.5, 0.5), we look for the pairs of 2-tuples that can be perturbed for proper embedding. If there is just one suitable 2-tuple, it corresponds to **Case B**. If there are more than one suitable 2-tuple, we select that tuple whose perturbation introduces the least embedding distortion (**Case C**). For example, among 2 pairs (0.6, 0.7) and (0.90, 0.95), (the elements in both pairs are changed to (0, 0) after embedding), the total perturbation equals $0.6 + 0.7 = 1.3$ and $0.90 + 0.95 = 1.85$, respectively. Hence, the first pair is used for embedding—in general, among two pairs with elements in (0.5, 1), the one with elements closer to 0.5 should be selected for embedding.
- The process is repeated for 3-tuples, if we do not get a suitable 2-tuple for embedding. **Case D** corresponds to the use of a 3-tuple for embedding. If a suitable 1/2/3-tuple cannot be found for embedding, then a higher order tuple cannot also be found for embedding using (7,3) ME (**Case E**). Thus, data embedding is possible for **Cases A–D**; while for **Case E**, the desired 3 bits cannot be embedded due to too many erasures.

IV. ME-RA DECODING

We now discuss the decoder operations depicted in Fig. 2. One of the main challenges involved in using ME along with RA coding is the computation of the initial LLR values provided to the decoder. It is instructive to review the LLR computations used in prior work on QIM-based embedding [27], [28] before discussing the more complicated computations for ME-based embedding.

Definition of LLR: Let a certain image coefficient be equal to y and the corresponding embedded bit be b. The LLR value $\mathrm{LLR}(b|y)$ denotes the logarithm of the ratio of the likelihood that a 0 was transmitted through that coefficient $(P(b=0|y))$ to the likelihood that a 1 was transmitted $(P(b=1|y))$

$$\mathrm{LLR}(b|y)=\log\left(\frac{P(b=0|y)}{P(b=1|y)}\right). \tag{2}$$

Hence, $\mathrm{LLR}(b|y)$ is positive when 0 is the more likely bit and vice versa. $\mathrm{LLR}(b|y)$ equals zero for an erasure. In our QIM-based methods [27], [28], an image coefficient was changed to the nearest even or odd integer to embed 0 or 1, respectively. If the absolute value of y were less than 0.5, $\mathrm{LLR}(b|y)=0$ (erasure); else, depending on whether the received coefficient is close to an even or odd integer, $\mathrm{LLR}(b|y)$ is set to α or $-\alpha$, where α reflects the decoder's confidence in the accuracy of the decoded bit values.

LLR Computation for ME: For (7,3) ME, the 7-element sequence of received DCT coefficients, $\mathbf{y}^r$, decides the log likelihood for $\mathbf{b}$, a 3-tuple bit sequence. We denote it as $\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r)$ in (3)—it is a sequence of 3 terms, each denoting the LLR for an individual bit location. From (2) and (3), it is evident that while an LLR value $(\mathrm{LLR}(b|y))$ for QIM-RA depends on only one coefficient value (y), there is a $7:1$ mapping between the elements in $\mathbf{y}^r$ and an LLR value ($\mathrm{LLR}(b_j|\mathbf{y}^r)$, $1\le j\le 3$) for ME-RA

$$\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r)=\left(\log\left(\frac{P(b_1=0|\mathbf{y}^r)}{P(b_1=1|\mathbf{y}^r)}\right),\log\left(\frac{P(b_2=0|\mathbf{y}^r)}{P(b_2=1|\mathbf{y}^r)}\right),\right.$$
$$\left.\log\left(\frac{P(b_3=0|\mathbf{y}^r)}{P(b_3=1|\mathbf{y}^r)}\right)\right). \tag{3}$$

If we assume an ideal channel between the transmitted sequence $\mathbf{y}^e$ and the received sequence $\mathbf{y}^r$ (i.e., $\mathbf{y}^e=\mathbf{y}^r$), the 3-tuple of decoded bits, $\mathbf{b}^r$, obtained as the syndrome of $\mathbf{a}^r$, can be directly used for LLR estimation—here, $(\mathbf{b}^r)^T=\mathbf{H}(\mathbf{a}^r)^T$, where $a_i^r=\mathrm{mod}(\mathrm{round}(y_i^r),2)$, $\forall i$. In practice, the LLR values should be modified to account for those conditions where embedding of $\mathbf{b}$ in $\mathbf{y}$ was not possible because of erasures, or to account for errors, when $a_i^e\neq a_i^r$.

Motivation Behind Proposed LLR Computation Methods: In the absence of an exact mathematical model for the statistics of the hiding channel (which varies across images), we investigate three methods (Methods **M1**, **M2**, and **M3**) for LLR computation and empirically evaluate their performance in terms of the hiding rate (Table IX in Section VI). Our performance metric in comparing the LLR computation methods is the overall hiding rate; thus, a better LLR computation method would require a smaller redundancy factor for the RA decoding to converge. Both M1 and M2 compute the LLR $(\mathrm{LLR}(b_j|\mathbf{y}^r))$ as a summation of the individual LLRs, for each of the eight possible mappings between $\mathbf{b}^r$ (syndrome obtained at the decoder) and $\mathbf{b}'$ (syndrome at the encoder side, unknown to the decoder). For each individual LLR computation, M1 assigns an LLR value of $\{\alpha,-\alpha,0\}$, depending on whether the bit more likely to be embedded is $\{0,1,e\}$ (e = erasure), respectively—this is similar to the LLR allocation for QIM-RA [28]. M2 uses a more detailed analysis of the various erasure scenarios for LLR allocation. M3 computes the LLR as a ratio of probabilities, as per the definition (3), while considering all the eight conditions, along with the different erasure scenarios (like M2). Computing the LLR as a ratio instead of a summation of individual LLRs eliminates the need for choosing α explicitly.

We initially ignore the channel effects (errors and erasures associated with the transition probability matrix between $\mathbf{y}^e$ and $\mathbf{y}^r$) in our LLR computation methods (we assume that $a_i^e=a_i^r$, $\forall i$) described in Sections IV-A and IV-B, and modify them to incorporate the channel effects in Section IV-C.

A. LLR Computation-Method 1 (M1) and Method 2 (M2)

The final LLR values $\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r)$ are computed by summing over the individual LLRs for all the eight possible conditions (possible mappings between $\mathbf{b}^r$ and $\mathbf{b}'$ as shown in Table II)

$$\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r)=\sum_{j=0}^{7}P(\text{condition } j)\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r,\text{condition } j). \tag{4}$$

Here, condition j is the condition where proper embedding can occur by changing only y_j (or suitable 2–3 coefficients if y_j lies in the erasure zone), and for condition 0, no term in $\mathbf{y}$ needs to be modified. The prior probability of condition j is denoted by $P(\text{condition } j)$ (4) and equals 1/8 since all the $2^3=8$ conditions are equally likely. The problem now becomes one of computing the individual LLR terms $(\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r,\text{condition } j))$. If there is a coefficient (or a set of 2–3 coefficients) which lies outside the erasure zone and allows proper embedding, the condition is classified as "**Not in Erasure Band**" (**NEB**). If all the relevant coefficients lie in the erasure band, the condition is denoted as "**Erasure Band**" (**EB**). For NEB, the LLR is computed similarly by M1 and M2 while for EB, the LLR is computed differently.

1) NEB and EB Conditions: Assuming a certain condition (say, condition i) to be true, if y_i is the coefficient that needs to be modified for proper embedding and $|y_i^r|>1/2$, then the decoder is sure that proper embedding has occurred by modifying y_i (cases (A)–(D) in Fig. 3). When y_i^r rounds off to zero, it was either modified from ±1 (rounded value of y_i) to zero or it was not modified at all, if y_i were in the erasure zone; e.g., if $y_i\in(0.5,1]$ or $y_i\in[-1,-0.5)$ and the embedding logic demands that $a_i^e=\overline{a_i}$, then it is converted to $y_i^e=0$, after embedding. So, *a received y_i^r coefficient that rounds to zero leaves open the possibility that embedding could not be carried out in that location as happens for case* **E**, *assuming that condition i is true*—this is the "shrinkage" problem (this has been countered by nonshrinkage F5 (nsF5) [14], an improvement on the F5 scheme). If $|y_i^r|\le 0.5$, relevant 2–3 tuples are considered to check if a tuple with all the elements outside the erasure zone can be found—if yes, the condition is NEB; else it is termed as EB.

Example of NEB and EB: For a certain combination of $\mathbf{b}$ and $\mathbf{b}'$, let $b'_1=\overline{b_1}$, $b'_2=b_2$, and $b'_3=b_3$. Thus, proper embedding needs that only y_1 be changed suitably (condition 1 in Table II). If y_1^r equals zero after rounding, we test for suitable 2-tuples out of $\{(y_2^r,y_3^r),(y_4^r,y_5^r),(y_6^r,y_7^r)\}$, where the corresponding y_i terms could have been modified at the encoder so as to achieve the same effect as modifying y_1. Again, we are sure we have a suitable 2-tuple if both the coefficients have an absolute magnitude ≥ 1 (after rounding). If we do not find a suitable 2-tuple, we look for a suitable 3-tuple (Table II). Finally, depending on whether (or not) we find a suitable 1/2/3-tuple, the combination is considered to be an example of NEB (or EB).

LLR Computation for NEB: For the NEB condition, if the coefficient which we assumed to have been changed for embedding were indeed the one that was modified, the 3-tuple $\mathbf{b}^r$ computed from $\mathbf{y}^r$ would equal the sequence $\mathbf{b}$ that was supposed to be embedded. The LLR values are then computed identically, for both M1 and M2, based on $\mathbf{b}^r$, using (5)

$$\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r, \text{condition } j) = \alpha\left((-1)^{b_1^r}, (-1)^{b_2^r}, (-1)^{b_3^r}\right), \quad \text{if condition } j \text{ is NEB (for M1, M2).} \tag{5}$$

2) LLR Computation for EB: M1 and M2, compute the individual LLR values $\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r, \text{condition } j)$ differently.

- **Method M1**: We assign zeros, corresponding to erasures, to all the 3 bit locations for that condition.
- **Method M2**: If a certain condition is classified as EB, we can still guess the actual embeddable bits; e.g., consider the EB condition (condition 1 in Table II) where y_1 or corresponding 2–3 terms could not be modified. Then, $b_1' = \overline{b_1}$, $b_2' = b_2$, and $b_3' = b_3$ (Table II) is the relation between $\mathbf{b}$ (bits to embed) and $\mathbf{b}'$ (default syndrome) at the embedder side. Therefore, bit b_1 will be wrongly decoded. Thus, if the EB condition indeed occurred due to y_1, or corresponding 2–3 terms, lying in the erasure zone, ($\overline{b_1^r}$, b_2^r, and b_3^r) would be the actual embeddable bits. The LLR for the 3 bit locations, assuming that condition 1 is true, can then be expressed in terms of $\overline{b_1^r}$, b_2^r, and b_3^r, as shown later in (9).

For an EB condition, embedding may still have been possible. Thus, we have *less confidence* in the embeddable bit values suggested by an EB condition than by an NEB condition. M2 uses a weighting factor w_{EB} $(0 < w_{\mathrm{EB}} \leq 1)$ for the EB condition LLRs, used below in (9). We demonstrate proper embedding under an EB condition using an example. Say, for condition 1, we observe that $(y_1^r, y_2^r, y_4^r, y_7^r)$ are all in the erasure zone, while the remaining terms are outside the erasure zone. Then, if (y_1, y_2, y_4, y_7) were all originally in the erasure zone, embedding under condition 1 using 1/2/3-tuples would not be possible. However, it may have been that one/more of these y_i coefficients were in [−1,−0.5) or (0.5,1] and they got modified to values in the erasure zone after embedding. Thus, proper embedding can occur if $y_1 \in [-1, -0.5)$ or (0.5,1], or if $y_1 \in [-0.5, 0.5]$ and at least one element from (y_2, y_4, y_7) is outside the erasure zone. The probability of proper embedding is expressed as p_{embed} (6) and the weighting factor w_{EB} (7) corresponds to the probability that embedding could not be performed, given an EB condition

$$\begin{aligned} p_{\mathrm{embed}} &= P\left(y_1 \in \{[-1, -0.5) \cup (0.5, 1]\}\right) \\ &\quad + \left(P\left(y_1 \in [-0.5, 0.5]\right)\right) \\ &\quad \times \left(1 - \left(P\left(y_i \in [-0.5, 0.5]\right)\right)^3\right) \end{aligned} \tag{6}$$

$$w_{\mathrm{EB}} = 1 - p_{\mathrm{embed}}. \tag{7}$$

TABLE III

EXPLANATION OF LLR ALLOCATION FOR EXAMPLE-1: CONDITIONS {0,3,4,7} ARE THE NEB CONDITIONS. HERE, THE SYNDROME BASED ON $\mathbf{y}^r$, $(b_1^r, b_2^r, b_3^r) = (0, 0, 1)$. IF CONDITION j IS NEB, THE ACTUAL BIT SEQUENCE THAT WAS EMBEDDED IS ASSUMED TO BE (b_1^r, b_2^r, b_3^r), WHILE FOR AN EB CONDITION, THE EMBEDDABLE BIT SEQUENCE IS ASSUMED TO BE $(f_j(b_1^r), f_j(b_2^r), f_j(b_3^r))$ FOR M2 (FUNCTIONS f_j ARE OBTAINED FROM TABLE II). FOR A CERTAIN CONDITION, "BITS" REFERS TO THE ORIGINAL SEQUENCE THAT IS EMBEDDED (FOR AN NEB CONDITION) OR THE SEQUENCE THAT WAS SUPPOSED TO BE EMBEDDED (FOR AN EB CONDITION). WE ASSUME $w_{\mathrm{NEB}} = 1/2$

NEB or EB	Computing Total LLR values			Individual	Individual
Condition	Condition	Bits	Sequence	LLR (M1)	LLR (M2)
NEB	0	(b_1^r, b_2^r, b_3^r)	$(0, 0, 1)$	$\alpha(1, 1, -1)$	$\alpha(1, 1, -1)$
EB	1	$(\overline{b_1^r}, b_2^r, b_3^r)$	$(1, 0, 1)$	$(0, 0, 0)$	$\frac{\alpha}{2}(-1, 1, -1)$
EB	2	$(b_1^r, \overline{b_2^r}, b_3^r)$	$(0, 1, 1)$	$(0, 0, 0)$	$\frac{\alpha}{2}(1, -1, -1)$
NEB	3	(b_1^r, b_2^r, b_3^r)	$(0, 0, 1)$	$\alpha(1, 1, -1)$	$\alpha(1, 1, -1)$
NEB	4	(b_1^r, b_2^r, b_3^r)	$(0, 0, 1)$	$\alpha(1, 1, -1)$	$\alpha(1, 1, -1)$
EB	5	$(\overline{b_1^r}, b_2^r, \overline{b_3^r})$	$(1, 0, 0)$	$(0, 0, 0)$	$\frac{\alpha}{2}(-1, 1, 1)$
EB	6	$(b_1^r, \overline{b_2^r}, \overline{b_3^r})$	$(0, 1, 0)$	$(0, 0, 0)$	$\frac{\alpha}{2}(1, -1, 1)$
NEB	7	(b_1^r, b_2^r, b_3^r)	$(0, 0, 1)$	$\alpha(1, 1, -1)$	$\alpha(1, 1, -1)$
Final LLR value $LLR(\mathbf{b}\|\mathbf{y}^r)$				$\alpha(\frac{4}{8}, \frac{4}{8}, -\frac{4}{8})$	$\alpha(\frac{4}{8}, \frac{4}{8}, -\frac{4}{8})$

Though w_{EB} varies per image as it depends on the distribution of the quantized DCT coefficients, we empirically observe that $w_{\mathrm{EB}} = 0.5$ is a good choice across a variety of design QFs

$$\begin{aligned} &\mathrm{LLR}(\mathbf{b}|\mathbf{y}^r, \text{condition } j) \\ &\quad = (0, 0, 0) \\ &\qquad \text{if condition } j \text{ is EB (M1),} \end{aligned} \tag{8}$$

$$\begin{aligned} &\quad = \alpha w_{\mathrm{EB}}\left((-1)^{f_j(b_1^r)}, (-1)^{f_j(b_2^r)}, (-1)^{f_j(b_3^r)}\right) \\ &\qquad \text{if condition } j \text{ is EB (M2).} \end{aligned} \tag{9}$$

In (9), the functions $f_j(\cdot)$ are given by the jth condition. For an EB condition, f_j maps the computed (b_1^r, b_2^r, b_3^r) values to the actual bit sequence which we assumed was embedded; e.g., when proper embedding is not possible for condition 1, the mapping between $\mathbf{b}$ (bits to embed) and $\mathbf{b}^r$ (output syndrome) is $b_1^r = \overline{b_1}$, $b_2^r = b_2$, and $b_3^r = b_3$. Hence, $f_1(b_1^r) = \overline{b_1^r}$, $f_1(b_2^r) = b_2^r$, and $f_1(b_3^r) = b_3^r$. The LLR computation for M1 and M2 is explained through an example (Table III).

Example 1: In this example, $\mathbf{y}^r = (0.4, 0.2, 0.9, 1.5, 0.3, 0.1, 1.2)$, $\mathbf{a}^r = (0, 0, 1, 0, 0, 0, 1)$, and $\mathbf{b}^r = \mathbf{H}(\mathbf{a}^r)^T = (0, 0, 1)$. Conditions {0,3,4,7} are the NEB conditions. The LLR values for the NEB conditions are computed using (5) (for M1 and M2) and the EB condition LLRs are computed using (8) and (9), for M1 and M2, respectively. To show why condition 1 is classified as EB, consider y_1^r which lies in the erasure zone. Looking for higher order tuples that satisfy condition 1 (from Table II), we find that $(y_2^r, y_3^r, y_5^r, y_6^r)$ all lie in the erasure zone and hence, 2–3 tuples which are suitable for embedding cannot be found. For M2, if condition 1 is true and we assume that proper embedding has not occurred, the actual bit sequence that should have been embedded is $(\overline{b_1^r}, b_2^r, b_3^r) = (1, 0, 1)$; the resultant LLR equals $(\alpha/2)((-1)^1, (-1)^0, (-1)^1)$ using (9).

B. LLR Computation For ME-RA—Method 3 (M3)

For M1 and M2, when bit b is assumed to be embedded, the LLR value is given by $\alpha(-1)^b$, as in (5), where α denotes the

TABLE IV
LLR COMPUTATION FOR METHOD 3 BASED ON (10)—EXPLAINED USING EXAMPLES 2–3, AND USING $w_{\mathrm{EB}} = 0.5$, $w_{\mathrm{NEB}} = 1$

Example	$\mathbf{y}^r = (0.4, 0.2, 0.9, 1.5, 0.3, 0.1, 1.2)$	Example	$\mathbf{y}^r = (1.2, 0.2, 0.9, 1.5, 1.9, 0.1, 0.2)$
2	NEB conditions are $\{0, 3, 4, 7\}$	3	NEB conditions are $\{0, 1, 2, 3, 4, 5, 6, 7\}$
$LLR(b_1\|\mathbf{y}^r) =$	$\log\left(\frac{1+4w_{NEB}+2w_{EB}}{1+2w_{EB}}\right)$ = 1.0986	$LLR(b_1\|\mathbf{y}^r) =$	$\log\left(\frac{1+8w_{NEB}}{1+0}\right)$ = 2.1972
$LLR(b_2\|\mathbf{y}^r) =$	$\log\left(\frac{1+4w_{NEB}+2w_{EB}}{1+2w_{EB}}\right)$ = 1.0986	$LLR(b_2\|\mathbf{y}^r) =$	$\log\left(\frac{1+0}{1+8w_{NEB}}\right)$ = -2.1972
$LLR(b_3\|\mathbf{y}^r) =$	$\log\left(\frac{1+2w_{EB}}{1+4w_{NEB}+2w_{EB}}\right)$ = -1.0986	$LLR(b_3\|\mathbf{y}^r) =$	$\log\left(\frac{1+8w_{NEB}}{1+0}\right)$ = 2.1972

decoder's confidence level. The proper choice of α varies with the hiding parameters (QF_h, λ). *Here, we propose an LLR computation method which is independent of* α. By definition (3), an LLR term is a ratio of two probability terms—each term (e.g., $P(b_1 = 0|\mathbf{y}^r)$) can be replaced by the frequency of occurrence of that event, considering all eight conditions. We express $\mathrm{LLR}(b_1|\mathbf{y}^r)$ as a ratio, as shown below in (10). Of the eight possible values for $\mathbf{b}$, we determine which conditions correspond to $b_1 = 0$ and $b_1 = 1$, respectively, and also weight each condition differently depending on whether it corresponds to NEB or EB. The jth condition can result in $b_1 = 0$ either because 1) it is an instance of NEB and b_1^r equals 0, or 2) it is an EB condition and $f_j(b_1^r)$, as used in (9), equals 0. As explained before, the EB conditions are weighted less [by $w_{\mathrm{EB}} = 0.5$ in (9)] than the NEB conditions (weighted by $w_{\mathrm{NEB}} = 1$). We denote the number of NEB and EB conditions where $b_1 = b$ by $N_{\mathrm{NEB},b_1=b}$ and $N_{\mathrm{EB},b_1=b}$, respectively

$$\begin{aligned}
&\mathrm{LLR}(b_1|\mathbf{y}^r)\\
&= \log\left(\frac{P(b_1=0|\mathbf{y}^r)}{P(b_1=1|\mathbf{y}^r)}\right)\\
&= \log\left(\frac{\text{weighted frequency of occurrence of } b_1=0}{\text{weighted frequency of occurrence of } b_1=1}\right)\\
&= \log\left(\frac{N_{\mathrm{NEB},b_1=0}w_{\mathrm{NEB}}+N_{\mathrm{EB},b_1=0}w_{\mathrm{EB}}}{N_{\mathrm{NEB},b_1=1}w_{\mathrm{NEB}}+N_{\mathrm{EB},b_1=1}w_{\mathrm{EB}}}\right),\ \text{where}\\
&\quad N_{\mathrm{NEB},b_1=b} = |\{j : \text{condition } j \text{ is NEB and } b_1^r = b\}|,\\
&\quad N_{\mathrm{EB},b_1=b} = |\{j : \text{condition } j \text{ is EB and } f_j\left(b_1^r\right) = b\}|,\\
&\quad \text{where } |\{\cdot\}| \text{ denotes the cardinality of the set} \{\cdot\},\\
&\quad \text{and } w_{\mathrm{NEB}}\ (\text{or } w_{\mathrm{EB}}) = \text{weight assigned to a}\\
&\quad \text{NEB (or EB) condition} = 1\ (\text{or } 0.5).
\end{aligned}$$

To avoid the numerator or denominator in $\mathrm{LLR}(b_1|\mathbf{y}^r)$ becoming zero, we add 1 to both of them

$$\begin{aligned}
&\text{LLR computed using M3 :}\quad \mathrm{LLR}(b_1|\mathbf{y}^r)\\
&= \log\left(\frac{1+N_{\mathrm{NEB},b_1=0}w_{\mathrm{NEB}}+N_{\mathrm{EB},b_1=0}w_{\mathrm{EB}}}{1+N_{\mathrm{NEB},b_1=1}w_{\mathrm{NEB}}+N_{\mathrm{EB},b_1=1}w_{\mathrm{EB}}}\right). \quad (10)
\end{aligned}$$

For QIM-RA and the M1 and M2 schemes, the LLR is expressed in terms of the scaling factor α. For a given set of hiding conditions (QF_h, λ), the best possible α (α_{opt}) is that value which results in the highest data rate. We experimentally observe that the α_{opt} values used for M1 and M2 (for M2, $\alpha_{\mathrm{opt}} = 3, 3, 2, 2$ for $\mathrm{QF}_h = 40, 50, 60, 70$) result in similar LLR values for M3. Therefore, in (10), we add 1 to both the numerator and denominator to avoid the LLR becoming $\pm\infty$.

We provide two examples (Table IV) to explain the steps involved in LLR computation using M3.

Example 2: The value of $\mathbf{y}^r$ is the same as used in Table III. Focussing on b_1, $b_1^r = 0$ at the NEB conditions {0,3,4,7}, therefore, $N_{\mathrm{NEB},b_1=0} = 4$. Of the EB conditions, $f_j(b_1^r)$ equals 0 for conditions {2,6} and 1 for conditions {1,5}, respectively—hence, $N_{\mathrm{EB},b_1=0} = 2$ and $N_{\mathrm{EB},b_1=1} = 2$. The resultant $\mathrm{LLR}(b_1|\mathbf{y}^r)$ equals $\log((1 + 4w_{\mathrm{NEB}} + 2w_{\mathrm{EB}})/(1 + 2w_{\mathrm{EB}})) = 1.0986$.

Example 3: Let $\mathbf{y}^r$ be (1.2,0.2,0.9,1.5,1.9,0.1,0.2). Then, $\mathbf{a}^r = (1, 0, 1, 0, 0, 0, 0)$ and $\mathbf{b}^r = \mathbf{H}(\mathbf{a}^r)^T = (0, 1, 0)$. Embedding is seen to be possible for all eight cases (all NEB conditions). For b_1 and b_3, $\mathrm{LLR}(b_i|\mathbf{y}^r)$ equals $\log((1 + 8w_{\mathrm{NEB}})/(1 + 0)) = 2.1972$. $\mathrm{LLR}(b_2|\mathbf{y}^r)$ equals $\log((1 + 0)/(1 + 8w_{\mathrm{NEB}})) = -2.1972$.

C. LLR Computation Considering Channel Effects

Let us consider the flow $\mathbf{y} \rightarrow \mathbf{a} \rightarrow \mathbf{a}^e \rightarrow \mathbf{y}^e \rightarrow \mathbf{y}^r \rightarrow \mathbf{a}^r$, as shown in Fig. 2. For the LLR computation methods described in Sections IV-A and IV-B, we assume that there are no errors or erasures in the channel between $\mathbf{y}^e$ and $\mathbf{y}^r$. We now refine the LLR computation method, accounting also for channel effects.

For a received sequence $\mathbf{y}^r$, we can compute the LLR value for the 3 bit locations corresponding to these 7 coefficients using M1/M2/M3. However, considering channel effects, the received sequence $\mathbf{y}^r$ need not be the same as the transmitted sequence $\mathbf{y}^e$. Here, we guess the value of the transmitted sequence and refer to it as $\mathbf{y}^g$. For each guessed sequence $\mathbf{y}^g$, we compute the probability that the transmitted sequence is $\mathbf{y}^g$, given that the received sequence is $\mathbf{y}^r$. The final LLR value for $\mathbf{b}$, given the sequence $\mathbf{y}^r$ and considering the channel effects for all possible $\mathbf{y}^g$ sequences, $\mathrm{LLR}_{\mathrm{final}}(\mathbf{b}|\mathbf{y}^r)$ is computed as shown in (11)

$$\begin{aligned}
&\text{LLR expression considering channel effects :}\\
&\mathrm{LLR}_{\mathrm{final}}(\mathbf{b}|\mathbf{y}^r) = \sum_{\mathbf{y}^g} \mathrm{LLR}(\mathbf{b}|\mathbf{y}^g)P(\mathbf{y}^g|\mathbf{y}^r),\ \text{where} \quad (11)\\
&P(\mathbf{y}^g|\mathbf{y}^r) = \mathrm{Prob}(\mathbf{y}^g \text{ is the transmitted sequence,}\\
&\qquad\qquad \text{i.e., } \mathbf{y}^e = \mathbf{y}^g \text{ given that } \mathbf{y}^r \text{ is the}\\
&\qquad\qquad \text{received sequence).} \quad (12)
\end{aligned}$$

We express $P(\mathbf{y}^g|\mathbf{y}^r)$ (12) in terms of the parameters in the 3×3 transition probability matrix T, for the channel between

TABLE V
VARIATION OF THE HIDING RATE (bpnc) WITH DIFFERENT LLR COMPUTATION METHODS FOR (7,3) ME-RA, WITH $B = 9$, QF$_a = 75$, AND USING THE FIRST 19 AC DCT TERMS FOR HIDING ($\lambda = 19$), AND M3 TO COMPUTE INDIVIDUAL LLRs

LLR Model QF_h	without using T	using p_0 computed using T	using p_0 and p_1 computed using T
60	0.0731	0.0741	**0.0745**
70	0.0436	0.0493	**0.0498**
75	0.0265	0.0323	**0.0329**

$\mathbf{y}^g$ and $\mathbf{y}^r$, where each coefficient can be mapped to one of $\{0, 1, e\}$ (e = erasure)

$$T = \begin{bmatrix} t_{00} & t_{01} & t_{0e} \\ t_{10} & t_{11} & t_{1e} \\ t_{e0} & t_{e1} & t_{ee} \end{bmatrix},$$

$$\text{where } t_{ij} = P\left(m\left(y_k^r\right) = s_j | m\left(y_k^g\right) = s_i\right)$$

$$\text{and symbol set } s = \{0, 1, e\}, \tag{13}$$

$$\text{and the mapping function } m(\cdot) \text{ is such that}$$

$$m\left(y_k^r\right) = \begin{cases} 0/1 & \text{if } |y_k^r| > 0.5 \\ & \quad \& \bmod\left(\text{round}\left(y_k^r\right), 2\right) = 0/1 \\ e & \text{if } |y_k^r| \le 0.5. \end{cases} \tag{14}$$

Representing the sequences in terms of the ternary symbols, we consider only those guessed sequences $\mathbf{y}^g$, where 0 or 1 symbols are changed to obtain the received sequence $\mathbf{y}^r$ from $\mathbf{y}^g$, to compute LLR$_{\text{final}}$ (15)—we assume that the channel error and erasure probabilities are small enough so that $P(\mathbf{y}^g|\mathbf{y}^r) \approx 0$ when more symbols are changed

$$\text{LLR expression using 0/1 changes} : \text{LLR}_{\text{final}}(\mathbf{b}|\mathbf{y}^r) =$$

$$\sum_{i=0}^{1} \left\{ \sum_{\mathbf{y}^g \in B_i(\mathbf{y}^r)} \text{LLR}(\mathbf{b}|\mathbf{y}^g) P(\mathbf{y}^g|\mathbf{y}^r) \right\}$$

$$\text{where } B_i(\mathbf{y}^r) = \{\mathbf{y}^g : \mathbf{y}^g \text{ is obtained from } \mathbf{y}^r \text{ by}$$

$$\text{changing any } i \text{ elements in } \mathbf{y}^r\}. \tag{15}$$

The elements in T *depend on the JPEG compression channel and not on the distribution of the image DCT coefficients*. For a given set of hiding parameters (QF$_h$, QF$_a$, and λ), we compute T for each image, from a set of 500 images, and the average is used as an estimate of T.

We compute the probability terms $P(\mathbf{y}^g|\mathbf{y}^r)$ assuming that the symbols are independent of each other. Using $m(\mathbf{y}^g)$ and $m(\mathbf{y}^r)$ to denote the ternary symbols, the probability $P(\mathbf{y}^g|\mathbf{y}^r)$ equals $\prod_{i=1}^{7} P(m(y_i^g)|m(y_i^r))$.

For the T matrix, we assume $t_{00} = t_{11}, t_{01} = t_{10}, t_{e0} = t_{e1}$, $t_{0e} = t_{1e}$ (this is also experimentally verified). Let p_n denote the value of $P(\mathbf{y}^g|\mathbf{y}^r)$, when n LSBs from $\mathbf{y}^g$ are changed (through errors/erasures) to generate $\mathbf{y}^r$. The probability terms p_0 and p_1 are computed as shown below in (16) and (17), respectively,

$$p_n|_{n=0} = p_0 = (t_{00})^{7-n_1} (t_{ee})^{n_1}, \tag{16}$$

$$\text{where } n_1 \text{ elements in } \mathbf{y}^r \text{ are in the erasure zone}$$

$$p_1 = \begin{cases} (t_{00})^{6-n_1} (t_{ee})^{n_1} t_{e0}, & \text{if } e \text{ in } \mathbf{y}^g \Rightarrow 0/1 \text{ in } \mathbf{y}^r \\ (t_{00})^{6-n_1} (t_{ee})^{n_1} t_{0e}, & \text{if a } 0/1 \text{ in } \mathbf{y}^g \Rightarrow e \text{ in } \mathbf{y}^r \\ (t_{00})^{6-n_1} (t_{ee})^{n_1} t_{01}, & \text{if a } 0/1 \text{ in } \mathbf{y}^g \Rightarrow \text{a } 1/0 \text{ in } \mathbf{y}^r. \end{cases} \tag{17}$$

Experimental Setup: An output quality factor QF$_a$ of 75 is used for all the experiments here. While accounting for channel effects, we systematically increase the values of the design quality factor QF$_h$, as shown below in Table V. As QF$_h$ increases, the DCT coefficients are divided by a finer quantization matrix (the elements in the JPEG quantization matrix get smaller)—the perturbation introduced by a fixed JPEG channel (QF$_a$ $= 75$) can cause more errors/erasures if the original elements undergo finer quantization. We show results for QF$_h$ $= 60, 70$, and 75—the error and erasure probabilities are much smaller for lower QF$_h$. For QF$_h$ of 60, 70, and 75, QF$_a$ being fixed at 75, and using an embedding band of $\lambda = 19$ elements, T (averaged over 500 images) equals $\begin{bmatrix} 0.9589 & 0.0333 & 0.0078 \\ 0.0332 & 0.9592 & 0.0076 \\ 0.0043 & 0.0043 & 0.9914 \end{bmatrix}$, $\begin{bmatrix} 0.9108 & 0.0733 & 0.0160 \\ 0.0737 & 0.9102 & 0.0161 \\ 0.0126 & 0.0125 & 0.9749 \end{bmatrix}$, $\begin{bmatrix} 0.8776 & 0.1022 & 0.0201 \\ 0.1031 & 0.8761 & 0.0208 \\ 0.0192 & 0.0191 & 0.9617 \end{bmatrix}$, respectively. The average hiding rate is computed in terms of *bpnc* (explained in Table I). "Hiding rate" refers to the bpnc averaged over 250 images (the image dataset is explained in Section VII-A), while using the minimum RA-code redundancy (q_{opt}) that ensures perfect data recovery for each image. Since considering the transition probability matrix improves the bpnc for more noisy channels, we use (15) for LLR computation for QF$_h$ of 60 and 70 in further experiments. For lower QF$_h$ values, the LLR is computed using the erasure-only model (10).

Experimental Results: We show the usefulness of the assumed model (individual LLR terms LLR($\mathbf{b}|\mathbf{y}^g$) are computed using M3) in Table V. The bpnc using both p_0 and p_1, as in (15), for LLR computation is slightly higher than that computed using only p_0, while both are significantly higher than the erasure-only model as in (10), especially for channels with a higher error rate (QF$_h$ $= 70$ and 75).

V. PERFORMANCE IMPROVEMENT WITH PUNCTURED RA CODES

RA codes are near-optimal for our application because of the high proportion of erasures, but the available rates are limited to $1/q$, where q is an integer. We address this shortcoming by the use of punctured RA codes. Puncturing [2], [3] is a technique where the data rate is increased by deletion of some bits in the encoder output. The bits are deleted according to some puncturing matrix. We explain how puncturing operates using an example. Assume that there are 200 hiding locations—for an RA codeword of 200 bits, let the value of q_{opt} be 4 and hence, we can embed $200/4 = 50$ data bits. Now, we increase the effective codeword length using "puncturing." Let the new codeword length be 300—we assume that the extra $300 - 200 = 100$ bits are deleted (these deletions are regarded as erasures at the decoder output). As the effective channel is the same, the error and erasure rate for the 200 code-bits is unchanged while there are 100 additional erasures. We obtain a higher data rate if the new value of $q_{\text{opt}} \leq 5$, as $\lfloor 300/5 \rfloor > 50$. *The design choices for puncturing here are 1) the number of additional erasures and 2) their locations in an RA codeword.* By suitably puncturing the RA codeword which is embedded using ME-RA, we obtain a higher bpnc—this new approach is called "ME-RA-puncture." We first explain the algorithm and then describe why/how it results in an improved performance.

Algorithm Description:

- The embedding band remains the same for ME-RA and ME-RA-puncture schemes. We use the top 19 AC DCT coefficients per 8 × 8 block ($\lambda = 19$) for hiding. Let the number of $B \times B$ blocks pseudorandomly chosen by YASS be N_B. The total number of hiding coefficients $N = 19N_B$ (assuming $B \leq 15$).
- To create a longer codeword than ME-RA (which has $3\lfloor N/7 \rfloor$ code bits), we assume that η ($\eta \geq 3\lceil 19/7 \rceil$, i.e., $\eta \geq 9$) bits are embedded per 8 × 8 block. Hence, the codeword has ηN_B bits. The problem becomes one of distributing the $(\eta N_B - 3\lfloor N/7 \rfloor)$ erasures among the ηN_B code bits.
- One can spread the erasures pseudorandomly or the erasures can occur at consecutive locations (*bursty erasures*). Let L denote the set of locations which correspond to the $3\lfloor N/7 \rfloor$ code bits which are embedded out of ηN_B code bits.

  The four methods that we explore to obtain L are mentioned below:

  1) **Locally Random Erasures** (**LREs**): We assume that out of a set of η consecutive code bits, $(\eta - 9)$ bits are erased. Let the 9 pseudorandomly selected bit locations, decided based on a key shared with the decoder, used for embedding out of η locations be $\{\ell_i\}_{i=1}^{9}$, where $\ell_i \in \{1, 2, \ldots, \eta\}$. Thus, the set of the bit locations (out of ηN_B locations) which correspond to the RA-code bits which are actually embedded is $L = \cup_{i=1,k=0}^{i=9,k=N_B-1} \{\ell_i + \eta k\}$. As $3\lfloor 19N_B/7 \rfloor < 9N_B$, we consider the first $3\lfloor 19N_B/7 \rfloor$ of the $9N_B$ locations in L to obtain the embeddable code bits.
  2) **Locally Bursty Erasures** (**LBEs**): We assume that out of η consecutive code bits, locations $\{1, 2, \ldots, 9\}$ are used for embedding and $\{10, \ldots, \eta\}$ are erasure locations.
  3) **Globally Random Erasures** (**GREs**): Out of ηN_B locations, $3\lfloor N/7 \rfloor$ locations are pseudorandomly selected as the embedding locations. For LRE and GRE, the pseudorandom locations are decided based on a shared key, so that the decoder knows the additional erasure locations.
  4) **Globally Bursty Erasures** (**GBEs**): We assume that out of ηN_B locations, locations $\{1, 2, \ldots, 3\lfloor N/7 \rfloor\}$ are used for embedding while the remaining are erased.

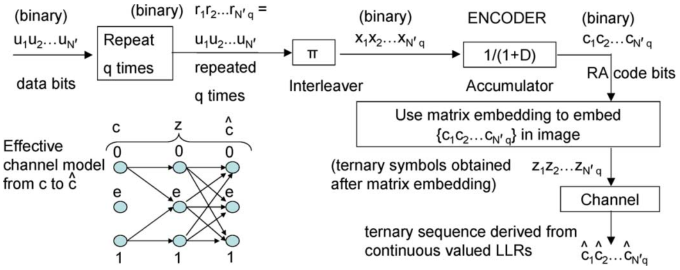

Fig. 4. Mapping between the binary RA code bits to the ternary sequence obtained from the continuous valued LLRs at decoder output is shown here for the ME framework—"channel" refers to the JPEG compression that introduces errors and erasures while mapping $\mathbf{z}$ to $\hat{\mathbf{c}}$. The additional erasure locations for the ME-RA-puncture scheme are selected in the final RA encoded sequence ($\mathbf{c}$) and not in the intermediate sequences ($\mathbf{r}$ or $\mathbf{x}$). For RA decoding, continuous valued LLRs are used—the ternary sequence ($\hat{\mathbf{c}}$) is used only to represent the channel as a 2 × 3 transition probability matrix.

- The LLR values for the locations where code bits are actually embedded (locations specified by L) are computed using M1, M2, or M3. The LLR values at the additional erasure locations are set to zero.

Understanding How "ME-RA-Puncture" Works: When can increasing the codeword length, while "actually embedding" the same number of bits, increase the effective hiding rate? Suppose that the RA codeword lengths for two different choices of "number of additional erasures" are $\eta_1 N_B$ and $\eta_2 N_B$ (we assume that both $\eta_1, \eta_2 \geq 9$ and $\eta_2 > \eta_1$). Let the value of q_{opt} for the two cases be $q_{\text{opt},1}$ and $q_{\text{opt},2}$, respectively. The number of data bits embedded is $\lfloor \eta_1 N_B / q_{\text{opt},1} \rfloor$ and $\lfloor \eta_2 N_B / q_{\text{opt},2} \rfloor$, respectively. As $\eta_2 > \eta_1$, the number of erasures introduced in the second case is higher (the channel becomes more noisy) and hence, the minimum redundancy needed $q_{\text{opt},2}$ may be equal to or higher than $q_{\text{opt},1}$. Thus, to have a higher data rate, the rate of increase in the redundancy factor q_{opt} should be less than the fractional increase in the code length, i.e., $(q_{\text{opt},2}/q_{\text{opt},1}) < (\eta_2/\eta_1)$.

To understand how the redundancy varies after inserting additional erasures, we study the effective hiding channel (the channel shown in Fig. 4 applies to both ME-RA and ME-RA-puncture), and observe how the channel transition probability matrix changes with η and how it affects q_{opt}. Since the LLR values are continuous valued for the ME-RA scheme, we empirically obtain a suitable threshold (δ) to map the LLR values to ternary symbols. The continuous LLR values belonging to $[-\infty, -\delta)$, $[-\delta, \delta]$ and $(\delta, \infty]$ are mapped to the three discrete values $-\alpha$, 0 and α, respectively. The 2 × 3 mapping from the binary RA code bits $\mathbf{c}$ to the ternary symbols $\mathbf{z}$ and then the 3 × 3 mapping between the ternary symbols ($\mathbf{z}$ and $\hat{\mathbf{c}}$) owing to the JPEG-based compression channel are shown in Fig. 4. The effective 2 × 3 mapping from $\mathbf{c}$ to $\hat{\mathbf{c}}$ is used to compute the effective channel capacity $\mathcal{C}$, which is obtained by maximizing the mutual information $I(\mathbf{c}, \hat{\mathbf{c}})$ (18)—a discrete memoryless channel is assumed here

$$\mathcal{C} = \max_{p(c)} I(\mathbf{c}, \hat{\mathbf{c}}) = \max_{p(c)} \sum_{c} \sum_{\hat{c}} p(c, \hat{c}) \log \left(\frac{p(c|\hat{c})}{p(c)} \right). \quad (18)$$

TABLE VI
FOR EACH IMAGE, THE bpnc IS COMPUTED USING ME-RA-PUNCTURE, USING QF$_h$ $= 60$ AND $B = 9$. THE bpnc INCREASES FOR A SUITABLE RANGE OF η (RATE OF ADDITIONAL ERASURES). HERE, LRE IS USED FOR ERASURE DISTRIBUTION

Image 1	$\eta=9$	$\eta=12$	$\eta=15$	$\eta=17$	$\eta=19$	$\eta=21$	$\eta=23$	**Image 2**	$\eta=9$	$\eta=12$	$\eta=15$	$\eta=17$	$\eta=19$	$\eta=21$	$\eta=23$
$\mathcal{C}$	0.0976	0.0746	0.0572	0.0510	0.0450	0.0407	0.0386	$\mathcal{C}$	0.2610	0.1933	0.1589	0.1421	0.1266	0.1162	0.1029
$\lceil\frac{1}{\mathcal{C}}\rceil$	11	14	18	20	23	25	26	$\lceil\frac{1}{\mathcal{C}}\rceil$	4	6	7	8	8	9	10
q_{opt}	17	18	21	22	25	29	31	q_{opt}	6	7	8	9	10	12	14
bpnc	0.0588	0.0740	0.0793	**0.0858**	0.0844	0.0804	0.0824	*bpnc*	0.0801	0.0915	0.1001	0.1008	0.**1014**	0.0934	0.0877
Image 3	$\eta=9$	$\eta=12$	$\eta=15$	$\eta=17$	$\eta=19$	$\eta=21$	$\eta=23$	**Image 4**	$\eta=9$	$\eta=12$	$\eta=15$	$\eta=17$	$\eta=19$	$\eta=21$	$\eta=23$
$\mathcal{C}$	0.1128	0.0850	0.0694	0.0592	0.0532	0.0499	0.0448	$\mathcal{C}$	0.1301	0.0965	0.0774	0.0717	0.0630	0.0558	0.0501
$\lceil\frac{1}{\mathcal{C}}\rceil$	9	12	15	17	19	21	23	$\lceil\frac{1}{\mathcal{C}}\rceil$	8	11	13	14	16	18	20
q_{opt}	11	15	17	19	20	24	27	q_{opt}	10	14	16	18	20	21	23
bpnc	0.0816	0.0798	0.0880	0.0892	**0.0948**	0.0873	0.0850	*bpnc*	0.0736	0.0701	0.0767	0.0772	0.0777	**0.0818**	**0.0818**

TABLE VII
THE bpnc VALUES ARE COMPUTED USING ME-RA-PUNCTURE FOR DIFFERENT ERASURE DISTRIBUTION METHODS, FOR VARYING QF$_h$ AND η, AND $B = 9$. THE CHANNEL EFFECTS ARE CONSIDERED FOR LLR COMPUTATION FOR MORE NOISY CHANNELS (QF$_h$ OF 60 AND 70) USING (15), WHILE AN IDEAL CHANNEL IS ASSUMED FOR QF$_h$ OF 50

Method	$QF_h = 50$			$QF_h = 60$			$QF_h = 70$		
used	$\eta{=}12$	$\eta{=}15$	$\eta{=}19$	$\eta{=}12$	$\eta{=}15$	$\eta{=}19$	$\eta{=}12$	$\eta{=}15$	$\eta{=}19$
LRE	0.0864	**0.0935**	0.0960	0.0801	0.0867	0.0872	0.0519	0.0529	0.0488
LBE	0.0830	0.0836	0.0794	0.0758	0.0738	0.0698	0.0475	0.0440	0.0413
GRE	**0.0876**	0.0930	**0.0975**	**0.0811**	**0.0874**	**0.0916**	**0.0527**	**0.0536**	0.0485
GBE	0.0739	0.0731	0.0727	0.0728	0.0723	0.0716	0.0518	0.0520	**0.0537**

The inverse of the capacity ($\lceil 1/\mathcal{C} \rceil$) provides the minimum redundancy factor needed for proper decoding for an ideal channel code—the RA code is expected to be close to the ideal channel code for channels with high erasure rates [27]. The minimum q needed for RA decoding should be equal to or slightly higher than this redundancy factor. We empirically observe that using $\delta = 0.30$ provides a 2×3 transition probability matrix between $\mathbf{c}$ and $\hat{\mathbf{c}}$ that results in q_{opt} values close to $\lceil 1/\mathcal{C} \rceil$.

Say the overall transition probability matrix between $\mathbf{c}$ and $\hat{\mathbf{c}}$, for $\eta = 9$, is expressed as $\mathcal{P} = \begin{bmatrix} \rho_{0,0} & \rho_{0,1} & \rho_{0,e} \\ \rho_{1,0} & \rho_{1,1} & \rho_{1,e} \end{bmatrix}$. For $\eta = \eta'$, where $\eta' > 9$, out of every η' bit locations, 9 bits obey the mapping specified by $\mathcal{P}$, while the remaining $(\eta' - 9)$ bits always get erased. The modified transition probability matrix $\mathcal{P}'$ (for $\eta = \eta'$) is related with $\mathcal{P}$ as follows:

$$\mathcal{P}' = \begin{bmatrix} \rho'_{0,0} & \rho'_{0,1} & \rho'_{0,e} \\ \rho'_{1,0} & \rho'_{1,1} & \rho'_{1,e} \end{bmatrix} = \left(\frac{9}{\eta'}\right)\mathcal{P} + \left(\frac{\eta' - 9}{\eta'}\right)\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{9}{\eta'}\rho_{0,0} & \frac{9}{\eta'}\rho_{0,1} & \frac{9}{\eta'}\rho_{0,e} + \left(\frac{\eta'-9}{\eta'}\right) \\ \frac{9}{\eta'}\rho_{1,0} & \frac{9}{\eta'}\rho_{1,1} & \frac{9}{\eta'}\rho_{1,e} + \left(\frac{\eta'-9}{\eta'}\right) \end{bmatrix}.$$

Let the channel capacity based on $\mathcal{P}$ and $\mathcal{P}'$ be denoted by $\mathcal{C}$ and $\mathcal{C}'$, respectively; we empirically observe that in general, $\mathcal{C}/\mathcal{C}' \approx \eta'/9$. Let $\mathcal{C}(\eta')$ denote the channel capacity using $\eta = \eta'$. The average values of $\mathcal{C}(9)/\mathcal{C}(12)$, $\mathcal{C}(9)/\mathcal{C}(15)$, $\mathcal{C}(9)/\mathcal{C}(17)$, $\mathcal{C}(9)/\mathcal{C}(19)$, $\mathcal{C}(9)/\mathcal{C}(21)$, and $\mathcal{C}(9)/\mathcal{C}(23)$ are 1.33, 1.66, 1.87, 2.11, 2.31, and 2.55 for QF$_h = 60$; for comparison, η'/η equals 1.33, 1.67, 1.89, 2.11, 2.33, and 2.56 for η' of 12, 15, 17, 19, 21, and 23, respectively, where $\eta = 9$. *For the RA code framework, the redundancy q is constrained to be an integer*. Hence, it is seen for certain cases (numerical examples from Table VI) that even on inserting extra erasures, the RA code redundancy remains the same or increases at a rate lower than $\eta'/9$ leading to increased bpnc.

TABLE VIII
THE bpnc VALUES ARE COMPUTED USING ME-RA-PUNCTURE FOR DIFFERENT η AND QF$_h$ AND LRE FOR ERASURE DISTRIBUTION; WE SET $B = 9$ AND USE M3 FOR INDIVIDUAL LLR COMPUTATION

QF_h	$\eta=9$	$\eta=12$	$\eta=15$	$\eta=17$	$\eta=19$	$\eta=21$	$\eta=25$
30	0.0594	0.0666	0.0756	**0.0777**	0.0776	0.0755	0.0750
40	0.0708	0.0785	0.0886	**0.0930**	0.0929	0.0895	0.0883
50	0.0766	0.0864	0.0935	0.0954	**0.0960**	0.0944	0.0923
60	0.0746	0.0801	0.0867	0.0870	**0.0872**	0.0863	0.0854
70	0.0499	0.0519	**0.0529**	0.0516	0.0488	0.0479	0.0456

Numerical Examples: For four sample images, η is varied from 9 to 23 and we observe how bpnc, $\mathcal{C}$ and q_{opt} vary, for QF$_h$ of 60, as shown in Table VI. The LRE method is used to determine the embeddable bit locations. We set $B = 9$ and use M3 for individual LLR computation.

Comparing Erasure Distribution Methods: In general, bursty erasures result in a lower bpnc as compared to when erasures are pseudorandomly distributed (Table VII). For less noisy channels (QF$_h = 50, 60$), LRE and GRE perform much better than LBE and GBE. For more noisy channels (QF$_h = 70$), erasures located in globally consecutive positions (GBE) perform similar to/better than LRE and GRE schemes. We set $B = 9$, QF$_a = 75$, and use M3 for individual LLR computation.

Experimental Results: Our experiments, performed on 250 images, show that as η is increased from 9, the bpnc increases significantly initially while it flattens out for η in the range 17–19, for "ME-RA-puncture" (Table VIII). We use (15) for LLR computation for QF$_h$ of 60 and 70 and use an erasure-only model for QF$_h$ of 30, 40, and 50 (the same setup is again used in Table IX and also in Section VII-B). We use LRE for choosing the embeddable code bits. We use M3 to compute the individual LLR values.

Utility of ME-RA-Puncture Scheme: The same number of RA-encoded bits gets embedded for the ME-RA and ME-RA-puncture schemes; hence, the effective embedding distortion

TABLE IX
Table Comparing the Hiding Rate (bpnc) Obtained After Using M1, M2, and M3, for LLR Computation, Using $B = 9$ and QF$_a$ $= 75$. For M1 and M2, the Optimum α (That Results in Highest bpnc for a Set of Hiding Parameters) Value for LLR Scaling is Empirically Determined and the Reported bpnc Corresponds to the Optimum α. For ME-RA and ME-RA-Nonshrinkage, We Use $\lambda = 19$, While for ME-RA-Puncture, We Use $\eta = 19$. The Effective bpnc Obtained Using M3 is Higher, in General, Than That Using M1 and M2

Hiding Method	Decoding Method	QF_h=30	QF_h=40	QF_h=50	QF_h=60	QF_h=70
ME-RA	M1	0.0566	0.0652	0.0695	0.0690	0.0463
	M2	0.0578	0.0672	0.0728	0.0715	0.0490
	M3	**0.0592**	**0.0706**	**0.0766**	**0.0745**	**0.0498**
ME-RA-puncture (using LRE for QF_h of 30-60 and GBE for QF_h of 70)	M1	0.0744	0.0875	0.0917	0.0841	0.0492
	M2	0.0760	0.0889	0.0948	0.0847	0.0519
	M3	**0.0776**	**0.0929**	**0.0960**	**0.0872**	**0.0537**
ME-RA-non-shrinkage	M3	0.0662	0.0760	0.0789	0.0755	0.0433

TABLE X
Comparing Detection Performance (P_{detect}) and Embedding Rate (bpnc) Using QIM-RA and "ME-RA-Puncture" Schemes—for $B = 9$ and 10, QF$_h$ $= 50$, QF$_a$ $= 75$. The bpnc for "ME-RA-Puncture (7,3)" [or ME-RA-Puncture (3,2)] is Higher Than That of "QIM-RA: 4 Terms" (or QIM-RA: 6 Terms) While the Latter is More Detectable, for the Self-Calibration-Based Features

Hiding	big-block size B=9					big-block size B=10				
Scheme	PF-219	PF-324	PF-274	Chen-486	bpnc	PF-219	PF-324	PF-274	Chen-486	bpnc
QIM-RA: 2 terms	69.45	65.52	67.73	52.39	0.0493	68.83	65.28	67.85	52.52	0.0382
QIM-RA: 4 terms	80.00	77.18	79.39	56.20	0.0864	78.53	74.97	77.91	55.09	0.0700
QIM-RA: 6 terms	81.84	77.67	84.05	57.55	0.1138	78.90	79.26	79.39	55.95	0.0923
ME-RA-puncture (7,3)	64.79	65.40	69.45	55.95	**0.0975**	63.31	68.83	67.61	54.36	**0.0805**
ME-RA-puncture (3,2)	74.97	72.27	78.65	61.60	**0.1200**	73.87	78.77	78.77	59.02	**0.0998**

and the detectability against steganalysis are identical for these methods. *Thus, for a proper choice of η, we obtain a higher data rate for ME-RA-puncture as compared to ME-RA even though both have the same detectability.*

VI. Comparison of LLR Computation Methods and Effects of Avoiding Shrinkage

In Section IV, we introduced three methods for LLR computation (M1, M2, and M3). Here, we compare the hiding rate (in terms of bpnc) obtained using these methods in Table IX. M2 and M3 are more complicated than M1 in the way the different erasure scenarios are analyzed. Performance-wise, in general, $\mathbf{M1} < \mathbf{M2} < \mathbf{M3}$ (in terms of bpnc achieved using these methods). It is also seen that the performance benefits of ME-RA-puncture over ME-RA (in terms of increased bpnc) are higher for lower QF$_h$ values, where the effect of erasures is more dominant. In Section VII-B, while reporting the steganalysis results using ME-RA-puncture, we also report its bpnc—for that, we use those parameters (erasure distribution method and η) which maximize the bpnc. We use GRE for erasure distribution for QF$_h$ of 50 and 60 and GBE for QF$_h$ of 70, and use $\eta = 19$ (based on Tables VII and VIII).

Avoiding Shrinkage: The MMx algorithm [17] avoids shrinkage as follows—when y_i lies in a nonerasure zone ([0.5,1] or [−1,−0.5]), but gets converted to zero after embedding, y_i is converted to 2 (or −2) depending on whether it is $\geq$0 (or <0). Thus, shrinkage is avoided as a zero-valued coefficient can arise only due to erasure and not due to embedding; however, embedding distortion is also higher for the nonshrinkage case which leads to higher detectability. For LLR computation, we now use $w_{\text{EB}} = 1$ for M3 (10) as there is no ambiguity between an erasure and an embedding.

While comparing the performance of nonshrinkage ME-RA with ME-RA, we observe that the nonshrinkage version results in a higher bpnc only when the shrinkage problem is dominant, which happens when the erasure rate is high enough, i.e., at lower QF$_h$ (of 30–60 in Table IX). The gain in bpnc (for ME-RA-nonshrinkage as compared to ME-RA) decreases as QF$_h$ increases (erasure rate decreases) from 30 to 60. The embedding distortion of the nonshrinkage version is always higher than ME-RA, which in turn has the same embedding distortion as ME-RA-puncture. Hence, the nonshrinkage scheme is expected to be more detectable than ME-RA-puncture for the same steganalysis features. In Table IX, it is seen that the bpnc for ME-RA-puncture is higher than the nonshrinkage version across different QF$_h$.

VII. Experiments and Results

We first compare the detectability of both the QIM-RA and the ME-RA-puncture schemes against steganalysis, at similar hiding rates (shown later in Tables X and XI). The hiding rates are adjusted by varying B and the number of coefficients used for hiding (λ). We also investigate the level of noise attacks up to which ME performs better than QIM, as shown later in Table XII. We also present the steganalysis results using some recently proposed features, most of which were designed specifically to detect YASS (Table XIII).

A. Setup for Steganalysis Experiments

The experiments are done on a set of 1630 high-quality JPEG images taken with a Canon S2-IS Powershot camera; the images were originally at a QF of 95 and they were JPEG compressed

TABLE XI
COMPARING P_{detect} AND bpnc FOR QIM-RA AND "ME-RA-PUNCTURE"—FOR $B = 25$ AND 49, $\text{QF}_h = 50$, $\text{QF}_a = 75$

Hiding	big-block size $B = 25$					big-block size $B = 49$				
Scheme	PF-219	PF-324	PF-274	Chen-486	bpnc	PF-219	PF-324	PF-274	Chen-486	bpnc
QIM-RA: 2 terms	71.53	66.38	71.17	53.74	0.0588	71.17	68.96	71.78	54.23	0.0606
QIM-RA: 4 terms	82.70	79.26	82.45	57.55	0.1018	82.33	80.00	83.56	59.14	0.1048
QIM-RA: 6 terms	84.29	80.61	86.26	59.51	0.1336	87.98	84.54	88.34	61.47	0.1379
ME-RA-puncture (7,3)	72.15	73.99	75.95	59.14	**0.1106**	69.08	67.61	71.04	56.69	**0.1136**
ME-RA-puncture (3,2)	77.30	82.82	81.35	62.54	**0.1382**	82.94	84.91	84.91	63.80	**0.1421**

TABLE XII
COMPARING BPNC UNDER VARIOUS ATTACKS—"QIM-n" REFERS TO THE "QIM-RA: n TERMS" METHOD, WHILE $(\mathbf{p}, \mathbf{q})$ REFERS TO THE "ME-RA-PUNCTURE $(\mathbf{p}, \mathbf{q})$" METHOD. FOR HIDING, WE USE $\text{QF}_h = 50$, $B = 9$, AND AFTER THE ATTACK, THE IMAGES ARE JPEG COMPRESSED USING $\text{QF}_a = 75$. HERE, THE bpnc FOR "ME-RA-PUNCTURE(7,3)" AND "ME-RA-PUNCTURE(3,2)" ARE COMPARED WITH THAT OF QIM-4 AND QIM-6, RESPECTIVELY

Gamma correction: $\gamma < 1$					Gamma correction: $\gamma > 1$					AWGN attack: SNR (dB)				
γ	QIM-4	QIM-6	(7,3)	(3,2)	γ	QIM-4	QIM-6	(7,3)	(3,2)	SNR	QIM-4	QIM-6	(7,3)	(3,2)
0.99	0.0851	0.1125	0.0953	0.1191	**1.01**	0.0854	0.1125	0.0959	0.1194	**50**	0.0860	0.1133	0.0952	0.1190
0.98	0.0839	0.1105	0.0929	0.1171	**1.02**	0.0843	0.1114	0.0935	0.1171	**45**	0.0859	0.1125	0.0940	0.1179
0.95	0.0783	0.1013	0.0849	0.1069	**1.05**	0.0799	0.1026	0.0863	0.1095	**40**	0.0840	0.1096	0.0885	0.1126
0.90	0.0608	0.0799	0.0655	0.0845	**1.10**	0.0631	0.0827	0.0685	0.0893	**35**	0.0728	0.0912	0.0659	0.0889
0.80	0.0307	0.0401	0.0200	0.0250	**1.20**	0.0366	0.0465	0.0260	0.0400	**30**	0.0393	0.0485	0.0200	0.0260

TABLE XIII
COMPARING P_{detect} FOR A VARIETY OF RECENTLY PROPOSED FEATURES TO DETECT YASS, USING $\text{QF}_h = 50$. WE USE $B = 9$ FOR THE QIM SCHEMES. THE ACRONYMS USED FOR THE VARIOUS METHODS ARE THE SAME AS USED IN TABLE XII

Feature	QIM-2	QIM-4	QIM-6	QIM-12	QIM-15	QIM-19	(7,3), B=9	(3,2), B=9	(3,2), B=10
KF-548	68.45	79.48	83.82	89.20	90.44	92.03	69.61	80.15	78.97
Li-14	54.01	55.27	56.75	53.74	58.70	67.65	52.88	59.93	55.76
Li-2	64.43	69.49	77.51	81.19	95.71	96.08	68.83	76.05	71.08
YB-243	54.64	55.70	56.75	58.12	64.01	69.89	54.23	59.68	56.12

at a QF of 75 for the experiments.[1] The advertised QF (QF_a) is, therefore, kept at 75, so that both the cover and stego images, considered for steganalysis, are at the same JPEG QF.

Steganalysis Performance Measures: The steganalysis results are expressed in terms of the detection probability P_{detect} (19) while the embedding rates are expressed in terms of the bpnc. We train a support vector machine (SVM) on a set of known stego and cover images. The SVM classifier has to distinguish between cover (class "0") and stego (class "1") image classes. Let X_0 and X_1 denote the events that the image being observed belongs to classes "0" and "1", respectively. On the detection side, let Y_0 and Y_1 denote the events that the observed image is classified as belonging to classes "0" and "1", respectively. The probability of detection P_{detect} is defined as follows:

$$\begin{aligned} P_{\text{error}} &= P(X_0)P(Y_1|X_0) + P(X_1)P(Y_0|X_1) \\ &= \frac{1}{2}P_{\text{FA}} + \frac{1}{2}P_{\text{miss}}, \text{ for } P(X_0) = P(X_1) = \frac{1}{2} \\ P_{\text{detect}} &= 1 - P_{\text{error}} \end{aligned} \quad (19)$$

where $P_{\text{FA}} = P(Y_1|X_0)$ and $P_{\text{miss}} = P(Y_0|X_1)$ denote the probability of false alarm and missed detection, respectively. Note that the above equation assumes an equal number of cover and stego images in the dataset ($P(X_0) = P(X_1) = 1/2$). An uninformed detector can classify all the test images as stego (or cover) and get an accuracy of 0.5. Thus, P_{detect} being close to 0.5 implies nearly undetectable hiding, and as the detectability improves, P_{detect} should increase towards 1. For the steganalysis results, we report P_{detect} as a percentage, at a precision of two significant digits after the decimal point.

Steganalysis Features: The following features are used as these have generally been reported as having the best detection performance among modern JPEG steganalyzers.

1) **PF-219/324/274**: Pevný and Fridrich's 274-dim feature vector (**PF-274**) is based on the self-calibration method [23] and it merges Markov and DCT features. The extended DCT feature set and Markov features are 193-dim (**PF-193**) and 81-dim, respectively. The logic behind the fusion is that while Markov features capture the intrablock dependency among DCT coefficients of similar spatial frequencies, the DCT features capture the interblock dependencies. For the extended DCT features [18], [23], the authors have a 219-dim implementation (**PF-219**).[2] The Markov features (**PF-324**) are obtained based on the 324-dim intrablock correlation-based feature set (**Shi-324**) proposed by Shi *et al.* [26]—the only difference being that the features are "calibrated" in [23].
2) **Chen-486**: Another steganalysis scheme that accounts for both intra- and interblock correlation among JPEG DCT

[1] We have experimentally observed that the detectability is higher using high-quality JPEG images than images taken with the same camera, but at poorer quality, i.e., JPEG compressed with lower QF. Hence, we use high-quality images for our experimental setup to show that ME-based YASS is more undetectable as compared to QIM-based YASS.

[2] **PF-219** differs from **PF-193** in the following ways: 1) In **PF-219**, there are 25 co-occurrence features for both the horizontal and vertical directions—these are averaged to give 25 features in **PF-193**. 2) Instead of one variation feature in **PF-193**, there are two variation features (for horizontal and vertical directions, separately) in **PF-219**.

TABLE XIV
THE bpnc VALUES ARE COMPARED FOR ME-RA AND QIM-RA METHODS, BEFORE AND AFTER PUNCTURING, AT QF$_h$ = 50

Hiding	$B = 9$		$B = 10$		$B = 25$		$B = 49$	
Scheme	before	after	before	after	before	after	before	after
QIM-RA: 2 terms	0.0493	0.0572	0.0382	0.0438	0.0588	0.0670	0.0606	0.0683
QIM-RA: 4 terms	0.0864	0.0965	0.0700	0.0784	0.1018	0.1110	0.1048	0.1134
QIM-RA: 6 terms	0.1138	0.1206	0.0923	0.0999	0.1336	0.1392	0.1379	0.1427
ME-RA (7,3)	0.0766	0.0975	0.0634	0.0805	0.1000	0.1106	0.1050	0.1136
ME-RA (3,2)	0.1100	0.1200	0.0900	0.0998	0.1300	0.1382	0.1350	0.1421

coefficients is the 486-dim feature vector, proposed by Chen *et al.* [6]. It improves upon the 324-dim intrablock correlation-based feature [26].

B. Discussion of Experimental Results

Comparison After Varying Big-Block Size B: The detection performance, in terms of P_{detect} (19), and the embedding rate, in terms of bpnc, are compared for QIM-RA and "ME-RA-puncture," using $B = 9$ and 10 (Table X), and 25 and 49 (Table XI). The ME-based method has been experimented with for both the (7,3) and (3,2) encoding schemes. The "QIM-RA: n terms" scheme has been defined in Table I.

From these tables, it is seen that P_{detect} is comparable for "QIM-RA: 2 terms" and "ME-RA-puncture (7,3)" while the latter has a higher embedding rate. The bpnc for "ME-RA-puncture (7,3)" [or ME-RA-puncture (3,2)] is higher than that of "QIM-RA: 4 terms" [or QIM-RA: 6 terms] while the latter is more detectable, for the self-calibration-based features. It is seen that YASS is more detectable using the self-calibration-based features than using Chen-486. Hence, the performance improvement of ME over QIM (lower P_{detect} at similar bpnc values) is more significant for PF-219/324/274 features.

Depending on the bpnc requirements for a certain stego scheme, one can decide whether to use (3,2) or (7,3) ME—the former allows for higher bpnc while the latter is more undetectable. Using (15,4) code for ME results in very low hiding rates and hence has not been considered.

Robustness Comparison for Various Noise Attacks: We now study how the bpnc is affected by additional noise attacks for these schemes. The YASS framework can be made robust against various global (*and not local*) attacks by adjusting the RA-code redundancy factor. We consider a wider range of attacks—gamma variation and additive white Gaussian noise (AWGN) attacks, which are followed by JPEG compression at QF$_a = 75$. It is seen that for higher noise levels ($|\gamma - 1| > 0.10$, for gamma variation, or SNR $\leq$ 35 dB, for AWGN), the bpnc is significantly lower for the ME-based method, as compared to QIM-RA, for similar detection rates (Table XII).

Using Recent Steganalysis Features More Tuned to Detect YASS: We explain the following features and then show their steganalysis performance in Table XIII:

1) **KF-548**: To improve upon the **PF-274** feature, Kodovský and Fridrich [19] proposed the use of a 548-dimensional feature set which accounts for both calibrated and uncalibrated features. Here, the reference feature is used as an additional feature instead of being subtracted from the original feature.
2) **Li-14** and **Li-2**: In [20], Li *et al.* propose the use of the frequency of requantized DCT coefficients in the candidate embedding band which round off to zero. The $(2i-1)$th and $(2i)$th features correspond to B of $(8+i)$, for $1 \leq i \leq 7$. Thus, if we are sure that $B = 9$, we use the first two dimensions of **Li-14**, i.e., **Li-2**; else when the exact value of B is not known, the 14-dim feature is used.
3) **YB-243**: In [30], Yu *et al.* propose a 243-dim feature based on transition probability matrices computed using the difference matrix computed in the pixel and DCT domains.

It is seen that in the lower embedding rate regime which is discussed in this paper, these newer features (**KF-548** and **Li-2**) provide similar levels of detectability as that provided by features already discussed, like **PF-274**. We have also experimented using a larger sized training set and P_{detect} increases marginally after increasing the size of the training set by a factor of more than 2 (for detailed results, see [1]).

Performance Comparison After Puncturing: We now use puncturing for QIM-RA and compare the bpnc results for ME-RA and QIM-RA, both with and without puncturing, in Table XIV. From Tables X and XI, ME-RA-puncture is less detectable than QIM-RA and also has higher bpnc. After using puncturing, we observe that the bpnc gain margin (of ME-RA-puncture over QIM-RA-puncture) decreases—however, in general, ME-RA-puncture is still less detectable (puncturing does not affect the detectability) than QIM-RA-puncture at similar bpnc values.

Fig. 5(a) illustrates how ME outperforms QIM in the "bpnc versus P_{detect}" trade-off. Considering points along the same vertical line (equal value of P_{detect}), the ME-points have higher y-values than the QIM-points, indicating higher bpnc. Fig. 5(b) corresponds to Table X—ME (7,3) [which corresponds to ME-RA-puncture (7,3)] is shown to be less detectable than QIM-2 (QIM-RA: 2 terms) and QIM-4 from ROC curves while Table X shows that ME (7,3) achieves higher bpnc than these QIM-based schemes. The variation in detectability with the hiding parameters (B, λ, (7,3) ME or (3,2) ME) for ME and QIM-based schemes is shown in Fig. 5(c) and (d), respectively.

To conclude, for hiding conditions where the embedding rate has to be low enough to ensure a certain level of undetectability, ME-based embedding with suitable puncturing generally results in higher bpnc than QIM, for similar robustness levels against steganalysis. However, this holds true only when the channel noise is low enough—for more severe noise, the LLR estimation for ME is erroneous enough to result in a lower hiding rate than QIM.

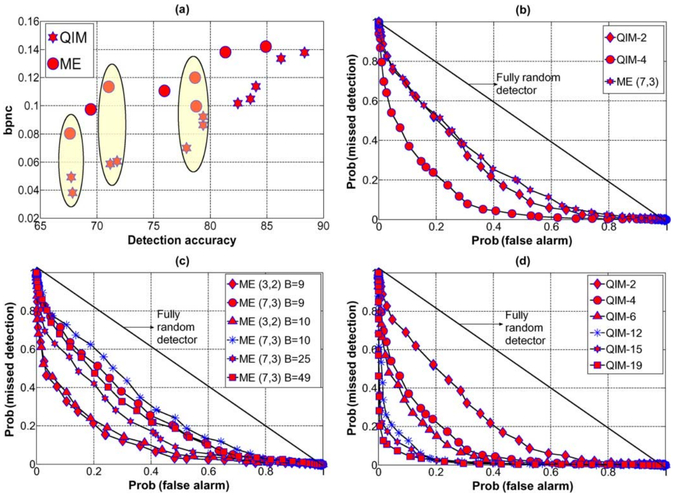

Fig. 5. **PF-274** is used for steganalysis in these plots—(a) trade-off of hiding rate versus detection accuracy is shown considering different parameter settings for ME and QIM-based hiding, as used in Tables X, XI, (b)–(d) comparison of ROCs is done for (b) ME versus QIM at comparable bpnc, (c) variants of ME, and (d) variants of QIM (varying λ). Here, $B = 9$ unless otherwise mentioned. The diagonal line corresponding to a fully random detector is kept as reference—the closer a ROC curve is to this line, more secure is the hiding method.

VIII. Conclusion

Randomized block-based hiding as in YASS [28] provides a powerful framework for secure hiding, especially against self-calibrating steganalysis. In this paper, we have shown that using ME instead of QIM within the YASS framework provides improved steganalysis performance in certain regimes, specifically when avoiding detection is a high priority (so that the hiding rate is small) and attacks are moderate. Technically, the key to our approach is to combine ME-based data hiding, which has a high embedding efficiency but is fragile against attacks, with a powerful channel code employing soft decisions. While we use RA codes as in our prior work, the LLR computation framework developed here, which depends only on the ME embedding logic, is applicable to any channel code whose decoder employs soft decisions (e.g., turbo codes or low density parity check codes). The performance is further improved by the use of punctured RA codes. While such codes have been used previously for obtaining good high-rate codes for classical communication channels [24], our results demonstrate their potential benefits for low-rate data hiding channels. It would be interesting to examine their utility in other stego schemes.

One approach to gain further performance improvements is to address the shrinkage problem in YASS: when given a zero coefficient, the decoder is confused as to whether the zero resulted from an embedding or due to an erasure. Fridrich *et al.* have used WPC [12], [13] to overcome this problem, as in the "non-shrinkage F5" method [14]. Combining WPC with the ME-RA framework might lead to further improvement in the embedding rate while maintaining the undetectability of the stego scheme. Another approach is to use more sophisticated "inner codes," possibly combining error correction with hiding as in [31], with RA or other turbo-like codes used as outer codes. However, the combinatorial complexity of computing soft decisions (at least in the direct fashion considered here) for such an inner code would be excessive for larger blocklengths and a larger number of data bits. An interesting topic for future research might be to explore techniques for overcoming this complexity bottleneck.

Acknowledgment

The authors would like to thank all the anonymous reviewers for their valuable comments.

References

[1] A. Sarkar, U. Madhow, and B. S. Manjunath, Estimating the Redundancy Factor for RA-encoded sequences and also Studying Steganalysis Performance of YASS Univ. of California, Santa Barbara, Dec. 2008 [Online]. Available: http://vision.ece.ucsb.edu/publications/Sarkar_tech_report_estimate_q.pdf

[2] A. Abbasfar*, Turbo-like Codes Design for High Speed Decoding*. New York: Springer Verlag, 2007.

[3] A. Abbasfar, D. Divsalar, K. Yao, R. Inc, and C. Los Altos, "Accumulate-repeat-accumulate codes," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 692–702, Apr. 2007.

[4] M. Backes and C. Cachin, "Public-key steganography with active attacks," in *Proc. Theory of Cryptography Conf.*, 2005, vol. 3378, pp. 210–226, Springer.

[5] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[6] C. Chen and Y. Q. Shi, "JPEG image steganalysis utilizing both intrablock and interblock correlations," in *Proc. Int. Symp. Circuits and Systems (ISCAS)*, May 2008, pp. 3029–3032.

[7] R. Crandall, Some Notes on Steganography, Posted on Steganography Mailing List 1998 [Online]. Available: http://os.inf.tu-dresden.de/~westfeld/crandall.pdf

[8] S. Craver, "On public-key steganography in the presence of an active warden," in *Proc. Second Int. Information Hiding Workshop*, pp. 355–368, Springer.

[9] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo-like codes," in *36th Allerton Conf. Communications, Control, and Computing*, Sep. 1998, pp. 201–210.

[10] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in Internet traffic with active wardens," in *Lecture Notes in Computer Science*, 2003, pp. 18–35.

[11] J. Fridrich, "Minimizing the embedding impact in steganography," in *Proc. 8th Workshop on Multimedia and Security (MM&Sec '06)*, New York, 2006, pp. 2–10, ACM.

[12] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed quantization steganography with wet paper codes," in *Proc. 2004 Workshop on Multimedia and Security (MM&Sec '04)*, New York, 2004, pp. 4–15, ACM.

[13] J. Fridrich, M. Goljan, and D. Soukal, "Wet paper codes with improved embedding efficiency," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 102–110, Mar. 2006.

[14] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in *Proc. ACM*, Sep. 2007, pp. 3–14.

[15] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 3, pp. 390–395, Sep. 2006.

[16] M. Kharrazi, H. Sencar, and N. Memon, "Image steganography: Concepts and practice," in *Lecture Note Series*. Singapore: Institute for Mathematical Sciences, National University of Singapore.

[17] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion," in *Lecture Notes in Computer Science: 8th Int. Workshop on Information Hiding*, Jul. 2006, pp. 314–327.

[18] J. Kodovský and J. Fridrich, "Influence of embedding strategies on security of steganographic methods in the JPEG domain," in *Proc. SPIE*, San Jose, CA, Jan. 2008, pp. 2 1–2 13.

[19] J. Kodovský and J. Fridrich, "Calibration revisited," in *Proc. 11th ACM Workshop on Multimedia and Security (MM & Sec '09)*, New York, 2009, pp. 63–74, ACM.

[20] B. Li, Y. Shi, and J. Huang, "Steganalysis of YASS," in *Proc. 10th ACM Workshop on Multimedia and Security*, New York, 2008, pp. 139–148, ACM.

[21] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Special Issue on Identification and Protection of Multimedia Information, Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[22] T. Pevný and J. Fridrich, "Multi-class blind steganalysis for JPEG images," in *Proc. SPIE*, San Jose, CA, 2006, pp. 257–269.

[23] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multiclass JPEG steganalysis," in *Proc. SPIE*, San Jose, CA, 2007, pp. 3 1–3 14.

[24] S. Planjery and T. Gulliver, "Design of rate-compatible punctured repeat-accumulate codes," in *IEEE Global Telecommunications Conf. 2007 (GLOBECOM'07)*, 2007, pp. 1482–1487.

[25] A. Sarkar, L. Nataraj, B. S. Manjunath, and U. Madhow, "Estimation of optimum coding redundancy and frequency domain analysis of attacks for YASS—A randomized block based hiding scheme," in *Proc. ICIP*, Oct. 2008, pp. 1292–1295.

[26] Y. Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Lecture Notes in Computer Science: 8th Int. Workshop on Information Hiding*, Jul. 2006, pp. 249–264.

[27] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Robust image-adaptive data hiding based on erasure and error correction," *IEEE Trans. Image Process.*, vol. 13, no. 12, pp. 1627–1639, Dec. 2004.

[28] K. Solanki, A. Sarkar, and B. S. Manjunath, "YASS: Yet another steganographic scheme that resists blind steganalysis," in *9th Int. Workshop on Information Hiding*, Jun. 2007, pp. 16–31.

[29] A. Westfeld, "High capacity despite better steganalysis (F5—A steganographic algorithm)," in *4th Int. Workshop on Information Hiding*, Apr. 2001, vol. 2137, pp. 289–302.

[30] X. Yu and N. Babaguchi, "Breaking the YASS algorithm via pixel and DCT coefficients analysis," in *19th Int. Conf. Pattern Recognition*, 2008, pp. 1–4.

[31] X. Zhang and S. Wang, "Stego-encoding with error correction capability," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, pp. 3663–3667, 2005.

Anindya Sarkar (S'06–M'09) received the B.E. degree from Jadavpur University, Kolkata, India, in 2003, in the Electronics and Telecommunication Engineering Department. He received the M.E. degree (with distinction) in signal processing from the Indian Institute of Science, Bangalore, in 2005, where his research was on speech segmentation. He received the Ph.D. degree from the Electrical and Computer Engineering Department, University of California, Santa Barbara, while working with Prof. Manjunath.

In the summer of 2008, he interned at Google Research in Mountain View, CA. His research interests include data hiding, steganography, image forensics, video fingerprinting, and efficient video search and retrieval.

Dr. Sarkar received the Alfred Hay Gold Medal for the best M.E. thesis in the Electrical Engineering Department, Indian Institute of Science.

Upamanyu Madhow (S'86–M'90–SM'96–F'05) received the bachelor's degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1985. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana-Champaign, in 1987 and 1990, respectively.

He is now a Professor in the Department of Electrical and Computer Engineering, University of California, Santa Barbara. His research interests are in communication systems and networking, with current emphasis on wireless communication, sensor networks and multimedia security.

Dr. Madhow is a recipient of the NSF CAREER award. He has served as Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION THEORY, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He is the author of the graduate textbook, *Fundamentals of Digital Communication* (Cambridge, U.K.: Cambridge Univ. Press).

B. S. Manjunath (S'88–M'91–SM'01–F'05) received the B.E. degree in electronics from the Bangalore University, in 1985, the M.E. degree in systems science and automation from the Indian Institute of Science, in 1987, and the Ph.D. degree in electrical engineering from the University of Southern California, in 1991.

He is now a Professor of Electrical Computer Engineering and Director of the Center for Bio-Image Informatics at the University of California, Santa Barbara. His current research interests include image processing, computer vision, image and video forensics, multimedia databases, and bio-image informatics.

Dr. Manjunath was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, and the *IEEE Signal Processing Magazine*.