LDGM Codes-Based Near-Optimal Coding for Adaptive Steganography

Qiyi Yao, Weiming Zhang, Kejiang Chen, *Member, IEEE*, and Nenghai Yu

***Abstract*— Steganographic coding is an essential part of adaptive steganography. There are only two practical near-optimal codes in the context of adaptive steganography so far: Syndrome-Trellis Codes (STCs) based on linear convolutional codes and Steganographic Polar Codes (SPCs) based on polar codes. It can be noticed that both STCs and SPCs are based on channel codes. Like the need for the variety of cryptographic algorithms, to make steganography practical and secure, it is important to devise more adaptive steganographic codes to create more choices for users. Moreover, we want to solve the long-lasting problem of whether lossy source codes-based near-optimal adaptive steganographic coding exists. In this paper, we consider using Low-Density Generator-Matrix (LDGM) codes in adaptive steganography where a new algorithm is proposed. First, we describe the framework of our LDGM codes-based steganographic coding algorithm and establish rigorous upper bounds on average embedding efficiency for individual LDGM steganographic codes with a given information bit degree distribution under the constant distortion profile. Then, we give a provably optimal method of distortion incorporation for adaptive steganography and provide the corresponding log-domain Belief Propagation Guided Decimation (log-BPGD) algorithm to minimize the additive distortion. The syndrome coding technique is applied to realize definitive encoding and decoding of the secret message. We report experiments for various distortion profiles, payload rates, and code lengths. The results verify the near-optimal performance of the proposed method, by which the possibility of designing near-optimal adaptive steganographic coding methods based on lossy source coding is confirmed.**

***Index Terms*— Adaptive steganography, LDGM codes, distortion incorporation, belief propagation, decimation.**

I. Introduction

INFORMATION hiding technology has gathered a lot of attention from researchers these years and is being widely used in many research fields and applications [1]. Steganography is a branch of information hiding that is mainly used in covert communication [2]. The most popular and effective steganographic scheme is called adaptive steganography [3] which consists of two phases: distortion calculation phase and steganographic coding phase. In the first phase, the distortion calculation phase, a distortion function defining the embedding loss introduced by a cover element modification is obtained which will be an input in the second phase. A distortion function is called *additive* when the modification of each element in cover media is considered to be independent, i.e., the total distortion is the sum of the modification distortion of each element. The aim of the second phase, the steganographic coding phase, is to complete the information embedding process while minimizing total distortion.

Throughout the development of steganographic coding, researchers have devised a lot of methods to minimize embedding distortion. The most commonly used methodology is *syndrome coding* [4] as the method usually uses a parity-check matrix of channel codes and embeds the secret message into the syndrome, therefore, the method is also known as *matrix embedding*. Matrix embedding was first introduced by Crandall [5] in 1998 and later implemented by other researchers based on many linear block codes such as Hamming codes [6], Golay codes [7] and BCH codes [8], [9]. These methods aim to minimize the number of elements changed by the embedding process, meaning that changing each element brings the same distortion. The model above where the modification distortion is constant is called the constant distortion model.

After the constant distortion model, the two-level distortion model, wet paper model [10], made a step forward. But very soon, the adaptive distortion model where the distortion of the elements in cover media could be any positive real number and could be different from each other appeared and predominated. Steganography on the adaptive distortion model is called adaptive steganography and the corresponding coding methods are called adaptive steganographic coding. Modified Matrix Embedding [12] was an algorithm that contains the thoughts of adaptive steganography but its performance is far from the theoretical bound [3]. Filler et al. [4] proposed the first and most widely used near-optimal adaptive steganographic coding algorithm: Syndrome-Trellis Coding (STC) which fits arbitrary additive distortion function using linear convolutional codes as well as the Viterbi decoding algorithm. STC had been the only choice for users for a long time after its appearance. Li et al. [13] proposed Steganographic Polar Coding (SPC) using polar codes [14] as well as its corresponding Successive Cancellation List (SCL) decoding algorithm [15]. SPC achieves comparable performance as STC for short covers and is superior for long covers.

Manuscript received 7 August 2022; revised 22 January 2023, 11 May 2023, and 11 September 2023; accepted 6 December 2023. Date of publication 13 December 2023; date of current version 18 April 2024. This work was supported in part by the Natural Science Foundation of China under Grant 62102386, U2336206, 62002334, 62072421, and 62121002, and by Anhui Initiative in Quantum Information Technologies under Grant AHY150400. The associate editor coordinating the review of this article and approving it for publication was M. Ji. *(Corresponding authors: Weiming Zhang; Kejiang Chen.)*

The authors are with the CAS Key Laboratory of Electro-Magnetic Space Information, University of Science and Technology of China, Hefei 230026, China (e-mail: zhangwm@ustc.edu.cn; chenkj@ustc.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2023.3342247.

Digital Object Identifier 10.1109/TCOMM.2023.3342247

STC and SPC are the only two practicable adaptive steganographic coding algorithms by far. Like the need for the variety of cryptographic algorithms, to make steganography practical and secure, it is important to devise more adaptive steganographic coding methods to create more choices for users. It can be seen that the two existing steganographic coding methods are all devised based on channel codes and the reason why STC and SPC show good performance is indeed that the performance of the underlying channel codes is close to the channel capacity [13]. As a result, it is natural to think of using Low-Density Parity-Check (LDPC) codes as well as the Belief Propagation (BP) decoding algorithm [16] to design a steganographic coding method, where one can expect a flexible, near-optimal, and fast algorithm due to the good attributes of the BP decoding algorithm on LDPC codes. The problem is that, to design a syndrome coding method, one needs to find a vector that is close to the given cover vector in the coset corresponding to the secret message to make sure that the secret message is successfully embedded into the cover. However, in most cases, the BP decoding algorithm is not able to find any member vector from the coset because the vectors in the coset and the cover vector are usually not close enough. Therefore, the steganographic encoding based on LDPC codes and the BP decoder fails in most cases [17], [35].

Although it is hard to use LDPC codes and the BP decoding algorithm to devise steganographic codes, the iterative thoughts can still be used in other ways. Inspired by the success of the Survey Propagation (SP) algorithm and Low-Density Generator-Matrix (LDGM) codes in lossy source coding for the Bernoulli symmetric source [20], [21], [22], Fridrich and Filler [17] designed a steganographic coding algorithm using LDGM codes and the SP algorithm in 2007. This algorithm is the first LDGM codes-based steganographic coding algorithm and can be applied to any LDGM code ensemble. However, only the constant distortion situation, where the modification distortion of each element is the same, was considered in [17]. Therefore, the algorithm is not able to minimize adaptive distortion as it only finds a stego vector close to the cover vector in Hamming distance, i.e., the algorithm is not suitable for practical adaptive steganography. Whether lossy source codes-based near-optimal adaptive steganographic coding exists has been an unsolved problem since then. Furthermore, the computational complexity of the SP algorithm is much higher compared to BP-based algorithms, which means that employing BP-based methods would significantly reduce the complexity of the original method in [17]. Aside from the practical coding methods evaluated experimentally, no theoretical analysis was conducted on LDGM steganographic code ensembles as well.

In this paper, we propose a new LDGM codes-based adaptive steganographic coding algorithm which is efficient and has near-optimal security performance. Rigorous upper bounds on average embedding efficiency for individual LDGM steganographic codes under the constant distortion profile are established. The essential problem of how to introduce the adaptive distortion, called *distortion incorporation*, is solved through a provably optimal approach which builds up a bridge between adaptive steganographic coding and LDGM codes-based lossy source coding. BP-based algorithms are employed, which contributes to the log-domain Belief Propagation Guided Decimation (log-BPGD) algorithm that minimizes the additive distortion. The proposed steganographic coding algorithm is very flexible which fits arbitrary distortion functions and can be applied to arbitrary cover lengths, meanwhile, users can choose different decimation methods according to their needs in time complexity or security performance. Experimental results show that the algorithm performs close to the rate-distortion bound of adaptive steganography as the existing steganographic coding algorithms STC and SPC while holding its superiority in flexibility and computational complexity for some decimation methods, e.g., soft decimation. The proposed algorithm performs the best in the large payload situation with relatively short cover lengths under which circumstances it outperforms STC and SPC.

The proposed method makes LDGM codes-based steganographic coding feasible in practical adaptive steganography and any new advances in LDGM codes-based lossy source coding can be applied to the algorithm. The main contributions of the paper are as follows:

1) We establish rigorous upper bounds on the average embedding efficiency for individual LDGM steganographic codes with a given information bit degree distribution under the constant distortion profile. The upper bounds show that the performance of LDGM codes-based steganographic coding methods is bounded away from the optimal theoretical rate-distortion bound, which then indicates that LDGM codes-based steganographic coding methods are not optimal.
2) We give a method for distortion incorporation with proof of its optimality which solves the essential problem of making the LDGM codes-based steganographic coding suitable for the adaptive distortion model. The method breaks down the barrier between adaptive steganographic coding and LDGM codes-based lossy source coding, and thus makes the techniques used in LDGM codes applicable in steganography.
3) We propose the log-domain Belief Propagation Guided Decimation (log-BPGD) algorithm as an instance of our scheme to minimize the adaptive distortion. The algorithm performs close to the theoretical rate-distortion bound of adaptive steganographic coding.
4) We employ the hard decimation and soft decimation methods designed for steganography in our algorithm. Various decimation strategies are provided in hard decimation, which allows users to balance between complexity and security performance. The soft decimation method lowers the computational complexity from $O(n^2)$ to $O(n)$ with only a slight loss in security performance compared with the hard decimation methods.

The remainder of this paper is organized as follows. In the next section, we briefly review the fundament of adaptive steganographic coding along with the code construction of STC and SPC, and introduce LDGM graph representation. In Section III, we demonstrate the adaptive steganographic

coding scheme, derive the upper bounds of individual LDGM codes under the constant distortion profile, and give our method of distortion incorporation. In Section IV, we give the detailed log-BPGD algorithm using hard decimation and soft decimation as an instance in our scheme. Simulations and analysis on parameters and complexity are presented in Section V. The paper is concluded in Section VI.

II. Preliminaries

In this section, we introduce the fundamentals of adaptive steganographic coding, code construction of STC and SPC, and LDGM graph representation. Random variables will be typeset in capital letters while their corresponding realizations will be in lowercase throughout the paper. Besides, all the vectors we use in this paper are column vectors.

A. Fundament of Adaptive Steganographic Coding

The steganographic coding procedure is to embed the secret message into the cover media and obtain the stego to be transmitted, while the receiver extracts the secret message from the stego. We will use calligraphic font for sets and boldface for vectors in this section. We will use $\mathcal{I}$ to represent the alphabet of each element in the cover vector. For example, $\mathcal{I} = \{0, 1, \ldots, 255\}$ for 8-bit grayscale images. We will use $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in \mathcal{X} = \{\mathcal{I}\}^n$ to represent the cover vector with length n. Similarly, $\mathcal{I}_i \subset \mathcal{I}$ represents the alphabet of the i-th element in the stego vector. If $|\mathcal{I}_i| = 2$, we say the embedding process is *binary*, while if $|\mathcal{I}_i| = 3$, we say the embedding process is *ternary* [4], etc. $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T \in \mathcal{Y} = \mathcal{I}_1 \times \mathcal{I}_2 \times \ldots \times \mathcal{I}_n$ will be used to represent the stego vector and $\boldsymbol{m}$ will be used to represent the secret message. Most contents in this subsection are the same as [4].

1) Adaptive Distortion Model: A distortion function D is used to measure the distortion brought by the embedding process. If the embedding payload rate is fixed, the encoder will try to minimize the distortion function D. We only consider the additive distortion scenario in this paper as it is the most widely used situation and the embedding of non-additive distortion functions can be converted into a sequence of embeddings with an additive distortion [3]. The additive distortion function D has the form

$$D(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \rho_i(y_i), \tag{1}$$

where $\rho_i : \mathcal{I}_i \rightarrow [0, \infty)$ are functions representing the cost of replacing the cover element x_i with y_i. Note that in steganography, the cover vector $\boldsymbol{x}$ is fixed in the very beginning [4]. Besides, in the additive model, the embedding change made at each element x_i is independent of changes made at other elements.

We use $\pi(\boldsymbol{y})$ to denote the probability distribution of the embedding changes, $\pi(\boldsymbol{y}) \triangleq P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{x})$. As we assume that the cover $\boldsymbol{x}$ is fixed in the very beginning, we simply write $D(\boldsymbol{y}) \triangleq D(\boldsymbol{x}, \boldsymbol{y})$. If the decoder knew $\boldsymbol{x}$, the sender could send at most

$$H(\pi) = -\sum_{\boldsymbol{y} \in \mathcal{Y}} \pi(\boldsymbol{y}) \log_2 \pi(\boldsymbol{y}) \tag{2}$$

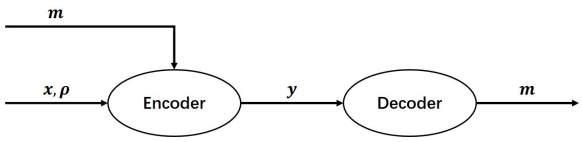


Fig. 1. Diagram of a common adaptive steganography system.

bits information on average. By generating the stego according to π, the average distortion

$$E_\pi(D) = \sum_{\boldsymbol{y} \in \mathcal{Y}} \pi(\boldsymbol{y}) D(\boldsymbol{y}) \tag{3}$$

would be introduced [3], [4]. With the help of the Gel'fand-Pinsker theorem [28], knowing $\boldsymbol{x}$ provides no fundamental advantage to the decoder, and the same performance can be achieved if $\boldsymbol{x}$ is only known to the encoder [4]. Indeed, the practical steganographic coding algorithms do not need the knowledge of $\boldsymbol{x}$ or $\boldsymbol{\rho}$ to extract the secret message.

The schematic of a common adaptive steganographic coding algorithm is shown in Fig. 1.

The main problem that we consider in the context of minimizing distortion while embedding the secret message under the adaptive distortion model is as follows.

a) Payload-limited sender (PLS): embed m bits *fixed average payload* while minimizing the average distortion,

$$\underset{\pi}{minimize}\ E_\pi(D) \quad subject\ to\ H(\pi) = m. \tag{4}$$

2) Performance Bounds: Following the maximum entropy principle [29, Th. 12.1.1], the optimal solution to the PLS problem has a form of a Gibbs distribution [3], [4]:

$$\pi(\boldsymbol{y}) = \prod_{i=1}^{n} \frac{exp(-\lambda \rho_i(y_i))}{Z_i(\lambda)}, \tag{5}$$

where the parameter $\lambda \in [0, \infty)$ is obtained by solving the algebraic equations corresponding to the constraints in (4), while in practice, a simple binary search is usually used because of the monotonicity of $H(\pi)$ and $E_\pi(D)$ w.r.t. λ; $Z(\lambda) = \sum_{\boldsymbol{y} \in \mathcal{Y}} exp(-\lambda D(\boldsymbol{y}))$, $Z_i(\lambda) = \sum_{y_i \in \mathcal{I}_i} exp(-\lambda \rho_i(y_i))$ are the corresponding normalization factors. By using the independence of the modifications on the cover elements/points, we have the definition

$$\pi_i(\boldsymbol{y}_i) \triangleq \frac{exp(-\lambda \rho_i(y_i))}{Z_i(\lambda)} \tag{6}$$

and the equation

$$\pi(\boldsymbol{y}) = \prod_{i=1}^{n} \pi_i(\boldsymbol{y}_i). \tag{7}$$

By modifying each cover element according to π_i in (6), one can simulate the optimal embedding which is very important in steganography. The security performance of an adaptive steganographic coding method is usually evaluated using *embedding efficiency* defined by $e = \frac{m}{D(\boldsymbol{y})}$ denoting message bits embedded per distortion incurred where m is the length of the secret message $\boldsymbol{m}$. By changing $D(\boldsymbol{y})$ into $E_\pi(D)$ where the distribution π follows (7), we have the optimal theoretical rate-distortion bound $e_\pi = \frac{m}{E_\pi(D)}$.

3) Binary Embedding: As the q-ary embedding can be implemented using multi-layered binary embedding [4], [30], [31] and if each binary embedding is optimal, the multi-layered embedding is optimal, the binary embedding problem is essentially the most important task. As a consequence, this paper mainly discusses the binary embedding problem, while using the same method in [4] and [30], one can easily extend the binary embedding algorithm to the q-ary situation.

In the binary embedding situation, we usually take the Least Significant Bit (LSB) of each cover point to make the modifications. We have $\mathcal{I}_i = \{x_i, \overline{x}_i\}$, $x_i \neq \overline{x}_i$. Then, the equation (6) would be in the form

$$\pi_i(\boldsymbol{y}_i) = \frac{exp\big(-\lambda\rho_i(y_i)\big)}{exp\big(-\lambda\rho_i(y_i = x_i)\big) + exp\big(-\lambda\rho_i(y_i = \overline{x}_i)\big)}. \tag{8}$$

Usually, if the cover point x_i is left unchanged during the embedding process, i.e., $y_i = x_i$, the distortion is 0. Then (8) can be rewritten into

$$\pi_i(\boldsymbol{y}_i) = \begin{cases} \dfrac{exp\big(-\lambda\rho_i(y_i = \overline{x}_i)\big)}{1 + exp\big(-\lambda\rho_i(y_i = \overline{x}_i)\big)} & y_i = \overline{x}_i \\ \dfrac{1}{1 + exp\big(-\lambda\rho_i(y_i = \overline{x}_i)\big)} & y_i = x_i \end{cases} \tag{9}$$

which is the most commonly used form in practice.

Here we give an example: under the constant distortion profile, we have that for each $i \in \{1, \ldots, n\}, \rho_i(y_i = x_i) = 0, \rho_i(y_i \neq x_i) = 1$. In this situation, by solving $H(\pi) = m$ using binary search, we can obtain parameter λ. Then, the modification probabilities in (9) are obtained.

B. STC and SPC

STC and SPC are the only two near-optimal adaptive steganographic coding methods in the literature. We briefly introduce the code construction of the two methods in this subsection. Readers can refer to [4] and [13] for more details about STC and SPC respectively.

1) STC: STC employs a parity-check matrix of the form shown in Fig. 2. It is obtained by placing submatrices $\hat{H}_{h\times w}$ along the diagonal of $H_{m\times n}$. The code construction of STC is to determine the submatrices $\hat{H}_{h\times w}$ which are found by exhaustive search for various parameters h and w. For a fixed submatrix $\hat{H}_{h\times w}$, the embedding payload rate $\alpha = \frac{1}{w}$. For each embedding, the encoder first chooses several submatrices and assembles them into the parity-check matrix fitting the code length and embedding payload rate. Then, it applies a steganographic Viterbi algorithm to finish the message embedding while minimizing the total distortion [4].

2) SPC: The code construction of SPC is by Arikan's heuristic method [37] modified for steganography. For each embedding, the algorithm assumes that the underlying channel is a Binary Erasure Channel (BEC) with erasure probability $\epsilon = \alpha$ where $\alpha = \frac{m}{n}$ denotes the embedding payload rate. With this assumption, the encoder uses Arikan's heuristic method to obtain the frozen indices and complete the code construction.

$$H_{m\times n} = \begin{pmatrix} \hat{h}_{1,1} & \cdots & \hat{h}_{1,w} & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \hat{h}_{1,1} & \cdots & \hat{h}_{1,w} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \hat{h}_{h,1} & \cdots & \hat{h}_{h,w} & \vdots & \ddots & \vdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \hat{h}_{h,1} & \cdots & \hat{h}_{h,w} & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \hat{h}_{1,1} & \cdots & \hat{h}_{1,w} & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \hat{h}_{2,1} & \cdots & \hat{h}_{2,w} & \hat{h}_{1,1} & \cdots & \hat{h}_{1,w} \end{pmatrix}$$

Fig. 2. The common form of an STC parity-check matrix.

$$\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

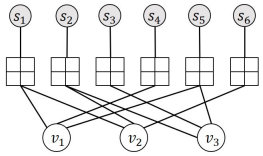


Fig. 3. An example of a generator matrix $\boldsymbol{G}_{3\times 6}$ and its corresponding factor graph in which the ⊞ symbols denote check nodes, the gray circles denote source bits and the white circles denote information bits.

The encoder then calculates the optimal modification probability distribution π_i for each cover element and employs the SCL encoder [13] to finish the embedding. The secret message is embedded in the frozen bits, so the decoder just needs to decode the stego and obtain the frozen bits.

C. LDGM Graph Representation

LDGM codes are duals of LDPC codes and thus the structure of a given generator matrix $\boldsymbol{G} \in \{0,1\}^{k\times n}$ can be captured by its factor graph [33] which is a bipartite graph defined as $\mathcal{G} = (V, C, E)$ with n *check nodes* $C = \{1, 2, \ldots, n\}$, k *information bits* $V = \{1, 2, \ldots, k\}$, n *source bits* [23] and the edge set $E \subseteq \{(a, i) | i \in V, a \in C\}$. We will use $\boldsymbol{s}$ to denote the source sequence in the following contents. Fig. 3 shows an example of a generator matrix $\boldsymbol{G}_{3\times 6}$ and its corresponding factor graph. The generator matrix $\boldsymbol{G}$ can be simply seen as the adjacency matrix of its factor graph where rows represent information bits and columns represent check nodes. We will use a to denote a certain check node and i to denote a certain information bit throughout the paper. Each check node a corresponds to a source bit s_a from which the check node gets information about the source. We define $V(a) = \{i \in V | (a, i) \in E\}, C(i) = \{a \in C | (a, i) \in E\}$ and $\overline{V}(a) = V(a) \cup \{s_a\}$.

LDGM codes can be divided into two classes: regular codes where the columns and rows in $\boldsymbol{G}$ share the same column weight and row weight respectively; irregular codes where the row and column weight are not both constant. In this paper, we mainly discuss irregular LDGM codes since regular codes can be recognized as special irregular codes. Like LDPC codes, an LDGM code ensemble can be represented by its length and degree distribution [34]. The degree distribution can be written as (ν, τ) from the edge perspective where $\nu(x) = \sum_{i=1}^{d_c} \nu_i x^{i-1}, \tau(x) = \sum_{i=1}^{d_v} \tau_i x^{i-1}$ with d_c denoting the max degree of check nodes and d_v denoting the max degree of information bits, and ν_i, τ_i denoting the portion of edges connected to check nodes and information bits with degree i, respectively. The degree of a check node a does not contain the corresponding source bit, i.e., only information bits are considered as neighbors here. We will use the degree distribution

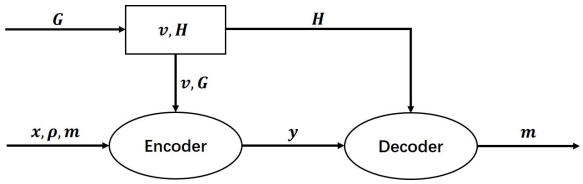


Fig. 4. The overall structure of the steganographic coding scheme.

of the information bits from the node perspective in the next section. The distribution is written as $L(x) = \sum_i L_i x^i$ where L_i represents the fraction of information bits of degree i. For regular LDGM codes, the polynomials $\nu(x), \tau(x)$, and $L(x)$ are indeed monomials. Consequently, they can be seen as special cases of irregular LDGM codes.

III. Coding Scheme, Upper Bounds, and Distortion Incorporation

In this section, we first give the overall structure of the adaptive steganographic coding scheme based on LDGM codes and then derive the upper bounds on average embedding efficiency of individual LDGM steganographic codes under the constant distortion profile. Finally, we demonstrate our method of distortion incorporation along with the proof of its optimality. As we are considering the binary embedding situation, we assume that all the vectors and matrices along with the arithmetic operations are all carried out in GF(2).

A. Structure of the Steganographic Coding Scheme

Assume that we have a sparse generator matrix $\boldsymbol{G} \in \{0,1\}^{k\times n}$ and its corresponding parity check matrix $\boldsymbol{H} \in \{0,1\}^{m\times n}$ where $m+k=n$. The core of syndrome coding is to embed the secret message $\boldsymbol{m}$ into the syndrome $\boldsymbol{Hy} = \boldsymbol{m}$ so that the decoder could simply extract the message by calculating $\boldsymbol{Hy}$. Let $\mathcal{C}(\boldsymbol{m}) = \{\boldsymbol{u} \in \{0,1\}^n | \boldsymbol{Hu} = \boldsymbol{m}\}$ be the aim coset, the embedding problem can be written as

$$\boldsymbol{y} \triangleq arg \min_{\boldsymbol{u}\in\mathcal{C}(\boldsymbol{m})} D(\boldsymbol{x}, \boldsymbol{u}). \tag{10}$$

For the given code, the optimization goal of the encoder is to find the $\boldsymbol{y}$ in (10).

Assume that we have an arbitrary $\boldsymbol{v} \in \mathcal{C}(\boldsymbol{m})$. Then, for each $\boldsymbol{u} \in \mathcal{C}(\boldsymbol{m})$, $\boldsymbol{u} - \boldsymbol{v} \in \mathcal{C}(\boldsymbol{0})$, thus, $\boldsymbol{y} - \boldsymbol{v} \in \mathcal{C}(\boldsymbol{0})$. Let $\boldsymbol{w} \in \mathcal{W}$ denote the information sequence where $\mathcal{W} = \{0,1\}^k$. Then, there is a one-to-one mapping $\phi : \mathcal{W} \mapsto \mathcal{C}(\boldsymbol{m})$ defined by:

$$\phi(\boldsymbol{w}) \triangleq \boldsymbol{G}^T\boldsymbol{w} + \boldsymbol{v}.$$

Here we can use LDGM codes-based lossy source coding algorithms for the Bernoulli symmetric source to find a $\boldsymbol{w}$ with the property that $\phi(\boldsymbol{w})$ (or $\boldsymbol{u}$) is close to $\boldsymbol{x}$.

The embedding process starts by calculating $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}$. Then the problem of finding a $\phi(\boldsymbol{w})$ (or $\boldsymbol{u}$) near $\boldsymbol{x}$ becomes the problem of finding a $\boldsymbol{w}$ to make $\boldsymbol{G}^T\boldsymbol{w}$ near $\boldsymbol{s}$. With the help of the lossy source coding process, the aim $\boldsymbol{w}$ is found and thus we will obtain the stego vector by calculating $\boldsymbol{y} = \phi(\boldsymbol{w})$. Fig. 4 shows the overall structure of the scheme.

In the steganographic coding scheme, an LDGM steganographic code is specified by the sparse generator matrix $\boldsymbol{G}$. Here we use G to denote the code.

In practice, it can be assumed that matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ are synchronized by the sender and receiver as the steganographic coding algorithm is shared. Here we concisely introduce the way we obtain $\boldsymbol{H}$ and $\boldsymbol{v}$ in the experiments of the paper. The method is to bring the original sparse matrix $\boldsymbol{G}$ into the *approximate lower triangular form* by a series of row and column exchanges [32]. Then, by using the new matrix $\boldsymbol{G}$, matrix $\boldsymbol{H}$ in systematic form can be obtained efficiently [32]. As a consequence, the special vector $\boldsymbol{v}$ would be $\boldsymbol{v} = (\boldsymbol{m}, \boldsymbol{0})$.

B. The Upper Bounds

For an LDGM steganographic code G with generator matrix $\boldsymbol{G} \in \{0,1\}^{(n-m)\times n}$, we define average embedding efficiency

$$\bar{e} \triangleq \sum_{\boldsymbol{m}} \frac{1}{2^m} \frac{m}{E[D(\boldsymbol{x}, \boldsymbol{y})]}, \tag{11}$$

where $\boldsymbol{y}$ is the real stego obtained by the encoder and $E[D(\boldsymbol{x}, \boldsymbol{y})] = \frac{1}{2^n}\sum_{\boldsymbol{x}} D(\boldsymbol{x}, \boldsymbol{y})$. Here we have the following result which states that the average embedding efficiency is independent of the secret message $\boldsymbol{m}$.

Lemma 1: Given LDGM steganographic code G with generator matrix $\boldsymbol{G} \in \{0,1\}^{(n-m)\times n}$, for any secret message $\boldsymbol{m}$ with length m, the embedding efficiency over all possible covers $\boldsymbol{x}$

$$e_{\boldsymbol{m}} \triangleq \frac{m}{E[D(\boldsymbol{x}, \boldsymbol{y})]} = \bar{e}. \tag{12}$$

Proof: Suppose that we have two different secret messages $\boldsymbol{m} \neq \boldsymbol{m}'$ whose lengths are m. Following the LDGM codes-based steganographic coding scheme, the two different messages correspond to two special vectors $\boldsymbol{v}$ and $\boldsymbol{v}'$, respectively. Then, for any cover vector $\boldsymbol{x}$, we have $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}$, $\boldsymbol{s}' = \boldsymbol{x} - \boldsymbol{v}'$. As $\boldsymbol{x}$ ranges over $\{0,1\}^n$, $\boldsymbol{s}$ and $\boldsymbol{s}'$ also range over $\{0,1\}^n$. Here we define a mapping $\psi : \{0,1\}^n \to \{0,1\}^n$ by

$$\psi(\boldsymbol{x}) = \boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{v} + \boldsymbol{v}'. \tag{13}$$

Since $\boldsymbol{v}$ and $\boldsymbol{v}'$ are fixed after the first step of the embedding process, the mapping is one-to-one.

Suppose that $\boldsymbol{x}$ is encoded into $\boldsymbol{y} = \boldsymbol{G}^T\boldsymbol{w} + \boldsymbol{v}$ when the secret message is $\boldsymbol{m}$. Similarly, $\boldsymbol{x}'$ is encoded into $\boldsymbol{y}' = \boldsymbol{G}^T\boldsymbol{w}' + \boldsymbol{v}'$ when the secret message is $\boldsymbol{m}'$. Since $\boldsymbol{x}' - \boldsymbol{v}' = \boldsymbol{x} - \boldsymbol{v} = \boldsymbol{s}$ by (13), we know that $D(\boldsymbol{x}, \boldsymbol{y}) = D(\boldsymbol{x}-\boldsymbol{v}, \boldsymbol{y}-\boldsymbol{v}) = D(\boldsymbol{s}, \boldsymbol{G}^T\boldsymbol{w}) = D(\boldsymbol{x}' - \boldsymbol{v}', \boldsymbol{y}' - \boldsymbol{v}') = D(\boldsymbol{x}', \boldsymbol{y}'), \boldsymbol{w}' = \boldsymbol{w}$. Then, $E[D(\boldsymbol{x}, \boldsymbol{y})] = E[D(\boldsymbol{x}' = \psi(\boldsymbol{x}), \boldsymbol{y}')]$. Therefore,

$$e_{\boldsymbol{m}} = \frac{m}{E[D(\boldsymbol{x}, \boldsymbol{y})]} = \frac{m}{E[D(\boldsymbol{x}' = \psi(\boldsymbol{x}), \boldsymbol{y}')]} = e_{\boldsymbol{m}'}. \tag{14}$$

Generally, the above result holds for any secret message pair $(\boldsymbol{m}, \boldsymbol{m}')$. Thus, we have that for each $\boldsymbol{m} \in \{0,1\}^m, e_{\boldsymbol{m}} = \bar{e}$, which concludes the proof. □

Here we note that the encoder we consider in the above proof is a definitive encoder. When a probabilistic encoder is considered, one just needs to change the distortion equation $D(\boldsymbol{x}, \boldsymbol{y}) = D(\boldsymbol{x}', \boldsymbol{y}')$ into an average form proportional to the probabilities. Then, the lemma still holds.

Now, we can give the main theorem of this subsection.

Theorem 1: Let n be the cover length and m be the length of the secret message $\boldsymbol{m}$. For any LDGM steganographic code

G with generator matrix $\boldsymbol{G} \in \{0,1\}^{(n-m)\times n}$ such that the degree distribution of the information bits is $L(x)$, the average embedding efficiency $\bar{e}$ under the constant distortion profile is bounded by

$$\bar{e} \leq \begin{cases} \dfrac{2m/n}{1 - RL'\left(1 - 2\left(\dfrac{x\left(\frac{1}{L'(1)}\right)}{1+x\left(\frac{1}{L'(1)}\right)} - \dfrac{a\left(x\left(\frac{1}{L'(1)}\right)\right)}{L'(1)}\right)\right)}, & \\ & R \in [0, \dfrac{1}{L'(1)}] \\ \dfrac{m/n}{\frac{x(R)}{1+x(R)} - a(x(R))R}, & R \in [\dfrac{1}{L'(1)}, 1], \end{cases} \tag{15}$$

where $R = 1 - \alpha = \frac{n-m}{n}$,

$$f(x) = \prod_{i=0}^{d_v} (1+x^i)^{L_i}, \quad a(x) = \sum_{i, L_i \neq 0} i L_i \frac{x^i}{1+x^i}, \tag{16}$$

and $x(r) \in [0,1]$ is the unique positive solution to equation

$$r = \frac{1 - H(\frac{x}{1+x})}{1 - \log_2 \frac{f(x)}{x^{a(x)}}}.$$

Proof: By Lemma 1, we have that the secret message $\boldsymbol{m}$ does not affect the average performance of code G. Therefore, without loss of generality, we assume that $\boldsymbol{m} = (0, \ldots, 0)$. In this situation, the coset $\mathcal{C}(\boldsymbol{m})$ is trivial and the special vector is simply $\boldsymbol{v} = (0, \ldots, 0)$. Then, for any cover vector $\boldsymbol{x} \in \{0,1\}^n$, $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v} = \boldsymbol{x}, \boldsymbol{G}^T\boldsymbol{w} + \boldsymbol{v} = \boldsymbol{G}^T\boldsymbol{w}$, where $\boldsymbol{G}^T\boldsymbol{w} + \boldsymbol{v}$ is the stego $\boldsymbol{y}$. Consequently, $D(\boldsymbol{x}, \boldsymbol{y}) = D(\boldsymbol{s}, \boldsymbol{G}^T\boldsymbol{w}) = D(\boldsymbol{x}, \boldsymbol{G}^T\boldsymbol{w})$, $E[D(\boldsymbol{x}, \boldsymbol{y})] = E[D(\boldsymbol{s}, \boldsymbol{G}^T\boldsymbol{w})]$.

By [39, Theorem 2], we have that for given rate $R = \frac{n-m}{n}$, the normalized average distortion $\frac{1}{n}E[D(\boldsymbol{s}, \boldsymbol{G}^T\boldsymbol{w})] = \sum_{\boldsymbol{s}} \frac{1}{2^n}\frac{1}{n} D(\boldsymbol{s}, \boldsymbol{G}^T\boldsymbol{w})$ is bounded by

$$\frac{1}{n}E[D(\boldsymbol{s}, \boldsymbol{G}^T\boldsymbol{w})] \geq \begin{cases} \dfrac{1}{2}\left(1 - RL'\left(1 - 2\left(\dfrac{x\left(\frac{1}{L'(1)}\right)}{1+x\left(\frac{1}{L'(1)}\right)} - \dfrac{a\left(x\left(\frac{1}{L'(1)}\right)\right)}{L'(1)}\right)\right)\right), & \\ & R \in [0, \dfrac{1}{L'(1)}], \\ \dfrac{x(R)}{1+x(R)} - a(x(R))R, & R \in [\dfrac{1}{L'(1)}, 1], \end{cases} \tag{17}$$

where functions $f(x), a(x)$ and $x(R)$ are defined in the theorem statement. Then, the average embedding efficiency $\bar{e} = \frac{m}{E[D(\boldsymbol{x},\boldsymbol{y})]} = \frac{m}{E[D(\boldsymbol{s},\boldsymbol{G}^T\boldsymbol{w})]}$ is bounded by

$$\bar{e} \leq \begin{cases} \dfrac{2m/n}{1 - RL'\left(1 - 2\left(\dfrac{x\left(\frac{1}{L'(1)}\right)}{1+x\left(\frac{1}{L'(1)}\right)} - \dfrac{a\left(x\left(\frac{1}{L'(1)}\right)\right)}{L'(1)}\right)\right)}, & \\ & R \in [0, \dfrac{1}{L'(1)}] \\ \dfrac{m/n}{\frac{x(R)}{1+x(R)} - a(x(R))R}, & R \in [\dfrac{1}{L'(1)}, 1], \end{cases} \tag{18}$$

which completes the proof. □

When the cover length tends to infinity, one just needs to replace m/n with α in (15) to extend the bounds in Theorem 1 to the limit situation.

Here we have the following corollary which says that our bounds are strictly bounded away from the optimal theoretical bound, which means that our bounds are better.

Corollary 1: For any given degree distribution of the information bits $L(x)$ with maximum degree d_v, under the same settings as Theorem 1, the bound (15) is strictly bounded away from the optimal theoretical bound in additive steganography under the constant distortion profile.

Proof: Applying [38, Lemma 2], we know that the optimal theoretical bound on average embedding efficiency for the constant distortion profile in steganography is given by $\frac{m}{nH^{-1}(\alpha)} = \frac{\alpha}{H^{-1}(\alpha)}$. Then, the optimal normalized average distortion is given by $H^{-1}(\alpha)$.

Let design distortion $\bar{D} = H^{-1}(\alpha)$. Then, the rate-distortion bound for lossy source coding suggests that the optimal rate $R = 1 - \alpha$ [29]. As a consequence, we have the optimal rate-distortion pair $(R, \bar{D})$ in lossy source coding which corresponds to the pair $(\alpha, H^{-1}(\alpha))$ in steganography. By (17) and [39, Theorem 2], we have that for any specific LDGM code, $\frac{1}{n}E[D]$ is strictly bounded away from the rate-distortion bound for lossy source coding. As the corresponding LDGM steganographic code shares the same normalized average distortion $\frac{1}{n}E[D]$, the bound on embedding efficiency $\bar{e} = \frac{m}{E[D]}$ in Theorem 1 is strictly bounded away from the optimal theoretical bound $\frac{m}{nH^{-1}(\alpha)}$. □

Since the performance of any individual LDGM steganographic code with bounded degree distribution $L(x)$ is strictly bounded away from the optimal theoretical bound in additive steganography under the constant distortion profile, LDGM steganographic codes are not optimal for adaptive steganography, i.e., no universally optimal code exists.

The bounds in Theorem 1 can be used to evaluate different degree distributions: by calculating the gap between the bound in Theorem 1 and the theoretical rate-distortion bound, one can distinguish between good distributions and bad distributions. Consequently, degree distributions that perform well at least under the constant distortion profile could be chosen. Moreover, our massive simulations show that codes optimized for the constant distortion profile are often good for other distortion profiles as well. Similar things also happen to STC and SPC [4], [13]. We will present some examples of how the theorem works in finding good codes in Section V-A.

C. Distortion Incorporation

Because we are considering the adaptive distortion situation, the common lossy source coding in Hamming distortion is not suitable. Here we use $< \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 >$ to denote the dot product of vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. We define the conditional probability distribution over LDGM codewords as

$$P(\boldsymbol{w}|\boldsymbol{s};\boldsymbol{\gamma}) = \frac{1}{Z} e^{-2<\boldsymbol{\gamma}\cdot(\boldsymbol{G}^T\boldsymbol{w}-\boldsymbol{s})>} \tag{19}$$

where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ is a vector related to the distortion profile $\boldsymbol{\rho} = \{\rho_i | i \in \{1, 2, \ldots, n\}\}$ and $Z = \sum_{\boldsymbol{w}} e^{-2<\boldsymbol{\gamma}\cdot(\boldsymbol{G}^T\boldsymbol{w}-\boldsymbol{s})>}$ is the normalization constant. Note that

the definition is the same as [23]. We give a definition of the process of getting $\boldsymbol{\gamma}$:

Definition 1 (Distortion Incorporation): Given distortion profile $\boldsymbol{\rho} = \{\rho_i | i \in \{1, 2, \ldots, n\}\}$, the mapping $f : \boldsymbol{\rho} \to \boldsymbol{\gamma}$ where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ is called *distortion incorporation*.

We use $\boldsymbol{w_s}$ to denote the most probable $\boldsymbol{w} \in \mathcal{W}$ subject to the source sequence $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}$. Then, the corresponding $\boldsymbol{G}^T \boldsymbol{w_s}$ is the optimal solution to the underlying lossy source coding problem for the given generator matrix $\boldsymbol{G}$ in the context of adaptive distortion.

Here come the two main problems to be solved in our steganographic coding scheme: distortion incorporation and lossy source coding in the context of adaptive distortion. In this subsection, we give a method to realize the essential problem of distortion incorporation and prove its optimality. The adaptive lossy source coding algorithm using log-BP combined with hard decimation and soft decimation will be demonstrated in Section IV.

There is a natural method to realize distortion incorporation used in [23] that one can perform the lossy source coding algorithm several times with a simple initialization of $\boldsymbol{\gamma}$ (e.g. vector $(1, \ldots, 1)$), obtain the sequence $\boldsymbol{G}^T \boldsymbol{w}$, estimate the probability distribution according to the embedding changes, and adjust $\boldsymbol{\gamma}$ w.r.t. the differences between the estimated distribution and the optimal distribution (8). Although this method shows good distortion performance (it has to), one needs to run the algorithm many times to finally find a good sequence $\boldsymbol{\gamma}$, which takes a lot of time. Besides, the method does not fully utilize the distortion profile $\boldsymbol{\rho}$ as well.

Here we give our method of distortion incorporation.

Theorem 2: Under the *independence assumption*,[1] for any given distortion profile $\boldsymbol{\rho} = \{\rho_a(y_a \neq x_a), \rho_a(y_a = x_a) | a \in C\}$ and embedding payload rate $\alpha = \frac{m}{n} \in (0, 1)$, the optimal distortion incorporation is given by

$$\boldsymbol{\gamma} = \{\gamma_a | \gamma_a = \frac{\lambda \rho_a(y_a \neq x_a) - \lambda \rho_a(y_a = x_a)}{2}, a \in C\} \quad (20)$$

where λ is the parameter in (8) and $C = \{1, \ldots, n\}$. Specially, when $\rho_a(y_a = x_a) = 0$, we have

$$\boldsymbol{\gamma} = \{\gamma_a | \gamma_a = \frac{\lambda \rho_a(y_a \neq x_a)}{2}, a \in C\}. \quad (21)$$

Proof: We first factorize the probability distribution (19):

$$\begin{aligned} P(\boldsymbol{w}|\boldsymbol{s}; \boldsymbol{\gamma}) &= \frac{1}{Z} e^{-2 < \boldsymbol{\gamma} \cdot (\boldsymbol{G}^T \boldsymbol{w} - \boldsymbol{s}) >} \\ &= \prod_{a \in C} \sigma_a(w_{V(a)}, s_a) \end{aligned} \quad (22)$$

where

$$\sigma_a(w_{V(a)}, s_a) = \begin{cases} \frac{e^{\gamma_a}}{e^{\gamma_a} + e^{-\gamma_a}}, & s_a = \sum_{i \in V(a)} w_i \\ \frac{e^{-\gamma_a}}{e^{\gamma_a} + e^{-\gamma_a}}, & s_a \neq \sum_{i \in V(a)} w_i. \end{cases} \quad (23)$$

[1]The independence assumption of the iterative message-passing algorithms assumes that the information received at each node from its neighbors are independent [16], [36].

The second equality in (22) is similar to the factorization in [23] and it follows from the independence assumption. Because $\boldsymbol{G}$ is row full-ranked, the mapping $\boldsymbol{w} \to \boldsymbol{G}^T \boldsymbol{w}$ is one-to-one. We will use $\boldsymbol{w}' = \boldsymbol{G}^T \boldsymbol{w}$ here. Then, (22) can be rewritten into

$$\begin{aligned} P(\boldsymbol{w}'|\boldsymbol{s}; \boldsymbol{\gamma}) &= \frac{1}{Z} e^{-2 < \boldsymbol{\gamma} \cdot (\boldsymbol{w}' - \boldsymbol{s}) >} \\ &= \prod_{a \in C} \sigma_a(w'_a, s_a). \end{aligned} \quad (24)$$

By combining (8) and (5), we have the optimal distribution

$$\begin{aligned} \pi(\boldsymbol{y}) &= \prod_{i=1}^{n} \frac{exp(-\lambda \rho_i(y_i))}{exp(-\lambda \rho_i(y_i = x_i)) + exp(-\lambda \rho_i(y_i = \overline{x}_i))} \\ &= \prod_{a \in C} \frac{exp(-\lambda \rho_a(y_a))}{exp(-\lambda \rho_a(y_a = x_a)) + exp(-\lambda \rho_a(y_a = \overline{x}_a))}. \end{aligned} \quad (25)$$

To make distortion incorporation optimal is to match the probability distributions (24) and (25), then for each $a \in C$, we have the equation set

$$\begin{cases} \frac{exp(-\lambda \rho_a(y_a = \overline{x}_a))}{exp(-\lambda \rho_a(y_a = x_a)) + exp(-\lambda \rho_a(y_a = \overline{x}_a))} = \frac{e^{-\gamma_a}}{e^{\gamma_a} + e^{-\gamma_a}} \\ \frac{exp(-\lambda \rho_a(y_a = x_a))}{exp(-\lambda \rho_a(y_a = x_a)) + exp(-\lambda \rho_a(y_a = \overline{x}_a))} = \frac{e^{\gamma_a}}{e^{\gamma_a} + e^{-\gamma_a}}. \end{cases} \quad (26)$$

By solving the above equation set, we have

$$\gamma_a = \frac{\lambda \rho_a(y_a \neq x_a) - \lambda \rho_a(y_a = x_a)}{2}. \quad (27)$$

Let $\rho_a(y_a = x_a) = 0$, we have

$$\gamma_a = \frac{\lambda \rho_a(y_a \neq x_a)}{2}, \quad (28)$$

which concludes the proof. □

The optimality of our method in Theorem 2 holds under the independence assumption, which means that the factor graph $\mathcal{G}$ corresponding to the generator matrix $\boldsymbol{G}$ is cycle-free. From [40], we have that for any $max\_iter \in \mathbb{N}^+$, the *computation graph* with depth max_iter which is a rooted subgraph of $\mathcal{G}$ becomes a tree with probability 1 when the blocklength n tends to infinity. Since the message-passing algorithms working on the factor graph $\mathcal{G}$ always have a maximum iteration number max_iter, in each iteration, the algorithms are indeed working on the computation graphs for each root node. Then, in the limit of infinitely long blocklengths, our method is optimal because the computation graphs are trees.

IV. The Adaptive Log-BPGD Algorithm

Here we consider the problem of adaptive binary lossy source coding where the modification distortion is not the usual Hamming distortion. We propose the log-domain Belief Propagation Guided Decimation (log-BPGD) algorithm which

employs log-BP updates combined with decimation steps to perform bitwise MAP estimation of the optimal vector $\boldsymbol{w_s}$ for a given source sequence $\boldsymbol{s}$. Hard decimation is considered in Section IV-A while the algorithm using soft decimation will be introduced in Section IV-B. Our method is inspired by [23] which gives a basic framework of weighted binary quantization based on LDGM codes.

A. The Log-BPGD Algorithm Using Hard Decimation

Given source vector $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}$, distortion profile $\boldsymbol{\rho}$ and generator matrix $\boldsymbol{G} \in \{0,1\}^{k\times n}$, as well as its corresponding factor graph $\mathcal{G}$, we first use Theorem 2 to obtain $\boldsymbol{\gamma}$ from $\boldsymbol{\rho}$. Then follows the execution process of the algorithm which is divided into many rounds where each round contains *max_iter* iterations over the factor graph $\mathcal{G}$. A decimation step is performed at the end of each round which decimates the most probable information bits, reduces the graph $\mathcal{G}$ by removing the decimated information bits from $\mathcal{G}$, and changes the source sequence $\boldsymbol{s}$ by performing

$$s_a = s_a + 1, \quad \forall a \in C(i),\ w_i = 1\ is\ decimated. \tag{29}$$

In each iteration, the original method in [23] uses the so-called probability-domain BP update equations while the development of the BP algorithm in the decoding of LDPC codes shows that the log-domain BP (log-BP) has many advantages over the original probability-domain one as it has lower complexity and is more numerically stable [33]. Thus, we modify the original update equations into the log domain and use the new update equations in the algorithm.

1) The Log-BP Update Equations: There are two kinds of values to be updated in the iterations in our method: the log-likelihood message LR and the bias message B. The constant bias message sent from source bit s_a to its corresponding check node a is defined as [23]

$$B_{s_a\to a} = (-1)^{s_a} tanh(\gamma_a). \tag{30}$$

Then, the iteration zero initializes

$$LR^{(0)}_{a\to i} = \log_e\left(\frac{1 - B_{s_a\to a}}{1 + B_{s_a\to a}}\right) \tag{31}$$

for every check node in the first round.

In the l-th iteration with $l > 0$, The algorithm first update messages $LR^{(l)}_i$ and $LR^{(l)}_{i\to a}$ denoting the **message update of information bits** using

$$LR^{(l)}_i = \sum_{b\in C(i)} LR^{(l-1)}_{b\to i}, \quad LR^{(l)}_{i\to a} = \sum_{b\in C(i)\backslash a} LR^{(l-1)}_{b\to i}. \tag{32}$$

Then, the algorithm would update $LR^{(l)}_{a\to i}$ denoting the **message update of check nodes** using

$$LR^{(l)}_{a\to i} = 2(-1)^{s_a+1} tanh^{-1}\Big(tanh(\gamma_a) \prod_{j\in V(a)\backslash i} B^{(l)}_{j\to a}\Big). \tag{33}$$

For the bias message $B^{(l)}_{i\to a}$ in (33) and the bias message $B^{(l)}_i$ used to judge if bit i is to be decimated, the **bias update of information bits** is given by

$$B^{(l)}_i = -tanh\left(\frac{LR^{(l)}_i}{2}\right), \quad B^{(l)}_{i\to a} = -tanh\left(\frac{LR^{(l)}_{i\to a}}{2}\right). \tag{34}$$

The hard decimation step uses $B^{(l)}_i$ denoting the bias of information bit i to choose the most probable bits to be decimated at the end of each round. If information bit i is going to be decimated, the fixing rule is

$$\begin{cases} w_i = 0, & B_i > 0 \\ w_i = 1, & B_i < 0. \end{cases} \tag{35}$$

The derivation details of the log-BP update equations are presented in the Appendix.

With the help of the update equations (32), (33) and (34), our first method using log-BP combined with hard decimation is provided in **Algorithm 1**.

Algorithm 1 Log-BP Combined With Hard Decimation

Input: generator matrix $\boldsymbol{G}$, source vector $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}$, distortion profile $\boldsymbol{\rho}$, and number of iterations per round max_iter.
Output: information sequence $\boldsymbol{w}$.

1: distortion incorporation: use $\boldsymbol{\rho}$ to get $\boldsymbol{\gamma}$ using Theorem 2;
2: initialization: calculate $B_{s_a\to a}$ using (30) and initialize using (31) for every check node;
3: **while** not all bits of $\boldsymbol{w}$ are decimated **do**
4: **for** integer l from 1 to max_iter **do**
5: update $LR^{(l)}_{i\to a}$ according to (32);
6: update $LR^{(l)}_{a\to i}$ according to (33) and (34);
7: update $B^{(l)}_i$ according to (34) and (32);
8: hard decimation: decimate the most probable information bits in $\boldsymbol{w}$, reduce the factor graph (generator matrix $\boldsymbol{G}$), and change the source sequence $\boldsymbol{s}$;
9: **return** $\boldsymbol{w}$.

2) Hard Decimation Stratagies: BP-based algorithms were originally designed to work on a tree [16]. In a tree factor graph, the likelihood would start to converge from the leaves and then gradually approach the root. When the factor graph has cycles, the likelihood of the nodes on a cycle may not converge like the tree situation. Therefore, BP-based algorithms are heuristic for a general factor graph as they are performed under the independence assumption.

The decimation steps are introduced just to break the cycles and make the factor graph locally tree-like. Assume that all nodes left in the factor graph are on cycles. Then by decimating the most probable nodes, some cycles would break and thus new leaves would emerge, which would improve the performance. As nodes on a cycle would never be perfectly decimated, for each decimation step, the fewer nodes are decimated, the better performance the algorithm shows.

We give three hard decimation strategies here:

- Bitwise decimation: decimate one bit per round. This is the natural strategy that yields the best security performance but holds the highest time complexity.

- Fixed ratio decimation: decimate a fixed ratio r of the whole information bits. The distortion performance would decrease as the ratio increases. Note that bitwise decimation is a special case of fixed ratio decimation with $r = \frac{1}{k}$.
- Threshold decimation: set a threshold t of the bias $|B_i|$ and for information bit i that satisfies $|B_i| > t$, this bit i will be decimated. There usually should be a min_num denoting the minimal number of bits decimated per round to prevent the situation that the decimation step never starts when all the biases are small.

The threshold decimation strategy is the commonly used method in the history of LDGM codes-based lossy source coding but we find it not performing well in our adaptive steganographic coding context. In practice, for users who concentrate on distortion performance, we recommend using the bitwise decimation; for users who prefer methods with low time complexity, we recommend using the fixed ratio decimation with a high ratio.

B. The Log-BPGD Algorithm Using Soft Decimation

The log-BPGD algorithm using hard decimation has the inherent computational complexity of $O(n^2)$ in practice because the decimation steps are performed in many rounds. By introducing the soft decimation algorithm, the complexity would be reduced to $O(n)$ with only a little loss in security performance. The original soft decimation method was devised for Hamming distortion in lossy source coding [25], i.e., it can only be used in the constant distortion situation. Here, we describe the main thoughts of the algorithm and give the update equations for the adaptive distortion model in the steganographic coding scheme.

The aim of soft decimation is to simulate the hard decimation algorithm without reducing the factor graph and do the decimation only one time for all bits at last. The modification of soft decimation update equations from the hard decimation update equations can be viewed from the following perspective: adding some intrinsic information to the hard decimation update equations so that the hard decimation procedure is incorporated into the update process.

There are two ways to add the intrinsic information:

- Adding the intrinsic information to $LR_{i\to a}$, which contributes to the update equations

$$LR_i^{(l)} = \sum_{b\in C(i)} LR_{b\to i}^{(l-1)},$$
$$LR_{i\to a}^{(l)} = \frac{1}{\mu} LR_i^{(l-1)} + \sum_{b\in C(i)\backslash a} LR_{b\to i}^{(l-1)}, \tag{36}$$

  where μ is a parameter which determines the convergent speed of the iterative message-passing procedure. The modification is to change (32) into (36);
- Adding the intrinsic information to $LR_{a\to i}$ which contributes to the update equation

$$LR_{a\to i}^{(l)} = \frac{1}{\mu} LR_{i\to a}^{(l)} + 2(-1)^{s_a+1} tanh^{-1} \times (tanh(\gamma_a) \prod_{j\in V(a)\backslash i} B_{j\to a}^{(l)}). \tag{37}$$

Algorithm 2 Log-BP Combined With Soft Decimation

Input: generator matrix $\boldsymbol{G}$, source vector $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}$, distortion profile $\boldsymbol{\rho}$, and maximum number of iterations max_iter.
Output: information sequence $\boldsymbol{w}$.

```
distortion incorporation: use ρ to get γ using Theorem 2;
initialization: randomly set LR_{i→a}^{(0)} to ±0.1;
for integer l from 1 to max_iter do
    if the update process has not converged then
        update LR_{a→i}^{(l)} according to (37) and (34);
        update LR_{i→a}^{(l)} according to (32);
    else
        break;
update B_i^{(l)} according to (34) and (32);
decimate all the information bits in w;
return w.
```

The modification is to change (33) into (37).

Experiments in the paper are conducted using the second form. The method of log-BP combined with soft decimation is provided in **Algorithm 2**. Note that we do not use the initialization (30), instead, we randomly set $LR_{i\to a}^{(0)}$ to ± 0.1 to start the log-BP algorithm [25].

V. Simulations and Analysis

Numerical simulations will be conducted along with some analysis in this section to study the performance of different methods. All the experiments consider binary embedding with $\rho_i(y_i = x_i) = 0$ and the PLS problem. The cover sequences and message sequences will be randomly generated of several sizes. Three distortion profiles will be used to evaluate the security performance: constant distortion profile $\rho_i = 1, i \in \{1, \ldots, n\}$, linear distortion profile $\rho_i = i, i \in \{1, \ldots, n\}$ and square distortion profile $\rho_i = i^2, i \in \{1, \ldots, n\}$. We will give results with various payload rates α and various cover lengths n. All the matrices we use are generated using the Progressive Edge Growth (PEG) [26], [27] algorithm with some $\nu(x)$ given and modified using the **Greedy Algorithm A** in [32]. The security performance is evaluated using embedding efficiency $e = \frac{m}{D(\boldsymbol{y})}$. All the results given in this section are averaged over 100 simulation trials.

A. Parameter Selection

1) Discussion of Different Degree Distributions: It is empirically recognized that a certain fraction of degree two checks is necessary for BP-based algorithms [22], [23], [25]. If there is no degree one check in the factor graph, the BP algorithm gets stuck as it was originally designed to work on a tree. With the hard decimation step introduced, sufficient degree two check nodes would ensure that degree one nodes are produced by the decimation procedure, thus making the BP algorithm perform well. As a consequence, irregular codes usually outperform regular codes because regular codes with check nodes degree two are often not good codes.

As dual codes of LDPC codes, intuitively, the degree distributions of LDGM codes can be obtained from optimization

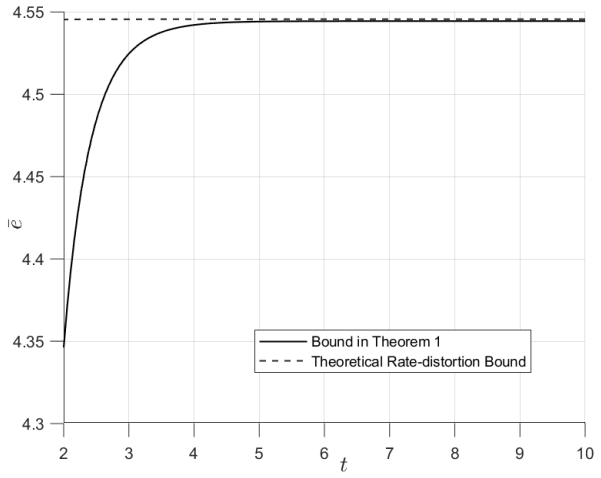


Fig. 5. The bounds of variable-regular LDGM steganographic codes in Theorem 1 with $L(x) = x^d$ when the embedding payload rate $\alpha = 0.5$.

based on density evolution [34]. Codes with these degree distributions optimized for the Binary Symmetric Channel (BSC) indeed show good performance in lossy source coding for the Bernoulli symmetric source [22]. By Theorem 1, we can see that the ideal performance of an LDGM steganographic code G with generator matrix $\boldsymbol{G}$ under the constant distortion profile is relevant to the ideal performance of the corresponding LDGM code for the Bernoulli symmetric source with the same generator matrix $\boldsymbol{G}$. Therefore, using density evolution would yield good results under the constant distortion profile in our scheme. Here we can use Theorem 1 to help us choose the proper degree distributions.

An example is the variable-regular codes[2] where the degrees of all information bits are the same. Suppose that the degree of each information bit is d, then $\tau(x) = x^{d-1}, L(x) = x^d$. Fig. 5 shows the bound of variable-regular LDGM steganographic codes in Theorem 1 when the embedding payload rate $\alpha = 0.5$. It can be seen that the bound in Theorem 1 is monotone increasing with respect to the degree d. Although the bound is strictly bounded away from the theoretical rate-distortion bound, when d is sufficiently large, the ideal performance can still be expected to be near-optimal. Since the computational complexity increases along with the increase of the degree d, we can choose d to be 6 or 7 so that the ideal performance is good enough while the lowest computational complexity would be kept if we consider applying variable-regular LDGM steganographic codes under the constant distortion profile with embedding payload rate $\alpha = 0.5$.

Another problem is whether the degree distributions that work under the constant distortion profile perform well under other distortion profiles. Fortunately, our massive experiments show that this is usually correct. Here we conduct an experiment where several degree distributions are examined. We consider variable-regular LDGM steganographic codes with information bits degree distributions

$$\tau_1(x) = x^3, \quad L_1(x) = x^4,$$

[2]Variable-regular LDGM codes, also known as generator-regular LDGM codes [39], have a similar concept to check-concentrated LDPC codes [40].

TABLE I

THE EMBEDDING EFFICIENCY e OF DIFFERENT DEGREE DISTRIBUTIONS WITH PAYLOAD RATE $\alpha = 0.5$ AND COVER LENGTH $n = 1008$

Distortion Profile	Degree Distribution	Embedding Efficiency e
Constant	$\nu_1(x) - regular$	3.6000
	$\nu_2(x)$	4.0102
	$\nu_3(x)$	**4.0217**
Linear	$\nu_1(x) - regular$	1.0033×10^{-2}
	$\nu_2(x)$	1.1157×10^{-2}
	$\nu_3(x)$	$\mathbf{1.1171 \times 10^{-2}}$
Square	$\nu_1(x) - regular$	2.4939×10^{-5}
	$\nu_2(x)$	2.6565×10^{-5}
	$\nu_3(x)$	$\mathbf{2.7136 \times 10^{-5}}$

$$\tau_2(x) = x^6, \quad L_2(x) = x^7,$$
$$\tau_3(x) = x^7, \quad L_3(x) = x^8. \tag{38}$$

Here group 1 is set to be the control group with regular codes applied. The corresponding degree distributions of the check nodes of the other two groups are obtained using density evolution. The distributions are

$$\begin{aligned}\nu_1(x) &= x,\\ \nu_2(x) &= 0.275698x + 0.25537x^2 + 0.076598x^3 + 0.392334x^8,\\ \nu_3(x) &= 0.23802x + 0.20997x^2 + 0.03492x^3 + 0.12015x^4\\ &\quad + 0.01587x^6 + 0.0048x^{13} + 0.37627x^{14}.\end{aligned} \tag{39}$$

Note that for the control group, all the check nodes have degree 2, which meets the requirements as we have discussed earlier.

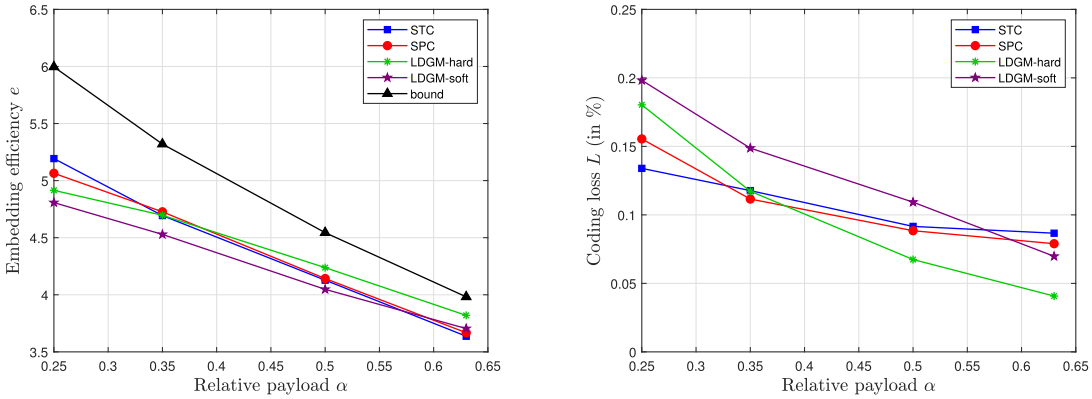
TABLE I shows the performance of these different degree distributions. By Theorem 1, along with the increase of the degrees of the information bits, the bound should get better. From the table, we can see that the actual performance preserves the same pattern no matter what distortion profile is considered. Therefore, we can expect good performance in the adaptive distortion situation from degree distributions optimized using density evolution that would work well under the constant distortion profile.

For general situations, e.g., other embedding payload rates, a similar procedure combining Theorem 1 and density evolution could also be applied in the code construction step.

2) Effect of Parameter μ in Soft Decimation: We use num_iter to denote the number of iterations needed for the soft decimation update process to converge. As μ grows larger, less intrinsic information is added in each iteration. Then the update procedure needs more iterations to converge, meaning that num_iter grows and the security performance gets better. The security performance would get stable and close to the performance of log-BPGD with the bitwise hard decimation method when μ and max_iter are relatively large. To keep comparable security performance, μ should increase along with the increase of embedding rate α or cover length n.

B. Performance of Different Strategies in Hard Decimation

We examine the coding performance of the strategies in hard decimation here. The simulations are conducted with embedding rate $\alpha = 0.5$ and cover length $n = 1008$. We use



(a) Embedding Efficiency (b) Coding Loss

Fig. 6. Security performance of different payload rates with cover length $n = 8320$ in constant distortion profile. The bitwise hard decimation method is used here. For soft decimation method, max_iter is set to be 400 while the soft parameter $\mu = 40$ for $\alpha = 0.25$ and $\mu = 45$ for other payloads.

TABLE II

THE EMBEDDING EFFICIENCY e AND NUMBER OF ITERATIONS num_iter WITH RELATIVE PAYLOAD $\alpha = 0.5$ AND COVER LENGTH $n = 1008$ UNDER THE LINEAR DISTORTION PROFILE

Decimation Strategy	Parameter	e ($\times 10^{-2}$)	num_iter
Bitwise Decimation	$r = 0.002$	1.1171	1008
Fixed Ratio Decimation	$r = 0.004$	1.1150	504
	$r = 0.006$	1.1106	336
	$r = 0.02$	1.0481	102
Threshold Decimation	$t = 0.7$	0.9124	250
	$t = 0.8$	0.9656	216
	$t = 0.9$	0.8944	140

num_iter denoting the number of BP update iterations to evaluate the time complexity of different methods. max_iter is set to 2 for bitwise decimation and fixed ratio decimation. For threshold decimation, the minimal decimation bit number per round min_num is set to 1 to make the method more competitive compared to the fixed ratio decimation. The degree distribution we use here is

$$\nu(x) = 0.23802x + 0.20997x^2 + 0.03492x^3 + 0.12015x^4 + 0.01587x^6 + 0.00480x^{13} + 0.37627x^{14}.$$

TABLE II reports the results under the linear distortion profile. It can be seen that the bitwise decimation method shows the best security performance while holding the highest time complexity, the fixed ratio decimation method with larger rates r performs a little worse than the bitwise decimation method, and the threshold decimation acts no better than the fixed ratio decimation. In the following experiments, we will use bitwise decimation for comparison.

C. Performance for Various Embedding Rates α

Here we use the same $\nu(x)$ as [23] to conduct the experiments. We employ *coding loss* $L = \frac{e_\pi - e}{e_\pi}$ to measure the distance between the practical performance and the theoretical rate-distortion bound and thus, making it clearer when comparing the proposed methods with STC and SPC. Fig. 6 shows the performance comparison of different methods in the constant distortion profile with cover length $n = 8320$. We choose the standard STC with parameter $h = 8$ and SPC with the SC algorithm here for comparison. The proposed LDGM codes-based methods act better as the payload α increases and they outperform STC and SPC when α is relatively large.

D. Performance for Various Cover Lengths n

Fig. 7 shows the performance comparison of different methods with payload $\alpha = 0.63$ of different cover lengths. The degree distribution we use here is

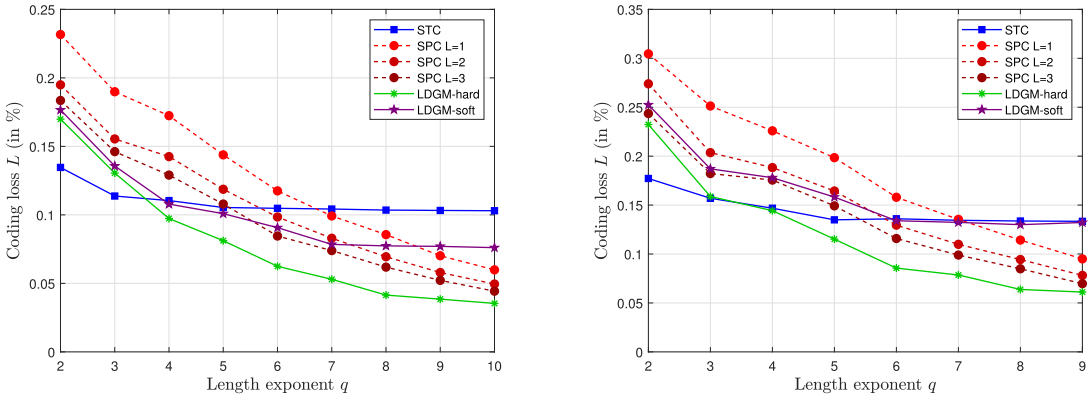
$$\nu(x) = 0.2710x + 0.2258x^2 + 0.1890x^5 + 0.0614x^6 + 0.2528x^{13}.$$

The performance of the proposed method is getting better along with the increase of the cover length as the coding loss L approaches 0. When the cover length is very short, STC performs the best. As the length grows larger, it is gradually surpassed by SPC and the proposed method. It can be noticed that the coding loss curves of the proposed method using soft decimation get flat with the growing length, which indicates that the performance is constrained by the fixed parameter max_iter. Therefore, for long covers with drastic distortion functions, a larger max_iter should be set to get better security performance.

E. Complexity Analysis

In this subsection, we compare the computational complexity of different methods. We use p to denote the average number of non-zero elements in each column of the low-density generator matrix $\boldsymbol{G} \in \{0,1\}^{k\times n}$. Then, pn denotes the number of non-zero elements in $\boldsymbol{G}$ which corresponds to the number of edges in the factor graph. In each iteration, information is sent on all the edges. So, the computational complexity of the proposed algorithm using bitwise hard decimation is $O(\frac{max\_iter}{2}kpn)$. When using the fixed ratio hard decimation, the complexity is $O(\frac{max\_iter}{2r}pn)$. For the soft decimation method, the complexity is $O(num\_iter\, pn)$.

TABLE III shows the complexity of the proposed methods, STC and SPC. When the code length is large, the proposed



(a) Linear Profile (b) Square Profile

Fig. 7. Security performance of different cover lengths $n = 2^q(2^6 + 1)$ in linear distortion profile and square distortion profile with relative payload rate $\alpha = 0.63$. The bitwise hard decimation method is used here. For the soft decimation method, the soft parameter μ increases from 30 to 45 as the length grows while max_iter is set to be 400 uniformly.

TABLE III
THE COMPUTATIONAL COMPLEXITY OF DIFFERENT METHODS

Bitwise hard decimation	Fixed ratio hard decimation	Soft decimation	STC	SPC
$O(\frac{max\_iter}{2}(1-\alpha)pn^2)$	$O(\frac{max\_iter}{2r}pn)$	$O(num\_iter\, pn)$	$O(2^h n)$	$O(l\, n \log\, n)$

method using soft decimation and fixed ratio hard decimation along with STC performs the fastest as their complexity is $O(n)$ in general. The complexity of the method using bitwise hard decimation is generally $O(n^2)$ which is better to be used when the user prefers the security performance to time complexity. The complexity of SPC is generally $O(n \log n)$ which stands in the middle. When the cover length is not too big, SPC could perform faster than STC or the proposed method using bitwise hard decimation.

VI. CONCLUSION

In this paper, we have investigated the potential of using LDGM codes in adaptive steganography. We established rigorous upper bounds on the average embedding efficiency for individual LDGM steganographic codes under the constant distortion profile. The bounds show that LDGM steganographic codes are not optimal. For practical concerns, we solved the essential problem of distortion incorporation in designing LDGM codes-based adaptive steganographic coding methods and proposed the near-optimal log-BPGD algorithm with hard decimation and soft decimation. We considered several decimation strategies in hard decimation to provide users with choices based on their preference for security performance or computational complexity. In soft decimation, by changing the soft parameter μ and max_iter, one can balance the complexity and performance as well. The proposed method confirms that adaptive steganographic coding algorithms can also be designed from the perspective of lossy source coding. Compared to STC and SPC, the proposed method performs better with a large payload α and relatively short cover length. Besides, our method has lower computational complexity when using soft decimation. By setting standards for the underlying generator matrix, the method would fit all conditions needed by users and would work efficiently. The new method provides yet another choice for steganographers to use in real-world applications and thus makes adaptive steganography more practical and secure.

APPENDIX
DERIVATION OF LOG-BP UPDATE EQUATIONS

We will derive log-domain BP (log-BP) update equations for adaptive binary lossy source coding based on irregular LDGM codes in this appendix where the original probability-domain BP update equations will be introduced first.

A. Original Probability-Domain BP Update Equations

In each iteration, there are three values to be updated in the original method in [23]: the likelihood R, the satisfaction message S, and the bias message B. The constant source message sent from source bit s_a to its corresponding check node a is defined as

$$B_{s_a \to a} = (-1)^{s_a} tanh(\gamma_a). \tag{40}$$

The message-passing process starts by setting

$$S^{(0)}_{a \to i} = B_{s_a \to a}. \tag{41}$$

The message update rules for the l-th iteration are as follows.
Message update of information bits:

$$R^{(l)}_i = \prod_{b \in C(i)} R^{(l-1)}_{b \to i}, \quad R^{(l)}_{i \to a} = \prod_{b \in C(i) \backslash a} R^{(l-1)}_{b \to i}. \tag{42}$$

Message update of check nodes:

$$R^{(l)}_{a \to i} = \frac{1 - S^{(l)}_{a \to i}}{1 + S^{(l)}_{a \to i}}, \quad S^{(l)}_{a \to i} = \prod_{j \in \overline{V}(a) \backslash i} B^{(l)}_{j \to a}; \tag{43}$$

Bias update of information bits:

$$B_i^{(l)} = \frac{1 - R_i^{(l)}}{1 + R_i^{(l)}}, \quad B_{i\to a}^{(l)} = \frac{1 - R_{i\to a}^{(l)}}{1 + R_{i\to a}^{(l)}}. \tag{44}$$

$R_{i\to a}^{(l)}, R_{a\to i}^{(l)}, S_{a\to i}^{(l)}$ and $B_{i\to a}^{(l)}$ denote the extrinsic likelihood message sent from information bit i to check node a, the extrinsic likelihood message sent from check node a to information bit i, the extrinsic satisfication message sent form check node a to information bit i and the extrinsic bias message sent from information bit i to check node a respectively; $R_i^{(l)}$ denotes the likelihood of information bit i; $B_i^{(l)}$ denotes the bias of information bit i.

B. Log-BP Update Equations

In log-BP update equations, we use LR to denote the log-likelihood message, and the satisfaction message is no longer needed. Firstly, by performing $\log_e()$ on both sides of (42), we have the **message update rule for information bits**

$$LR_i^{(l)} = \sum_{b\in C(i)} LR_{b\to i}^{(l-1)}, \quad LR_{i\to a}^{(l)} = \sum_{b\in C(i)\backslash a} LR_{b\to i}^{(l-1)}. \tag{45}$$

From the second equation in (43), we have

$$\begin{aligned} S_{a\to i}^{(l)} &= \prod_{j\in\overline{V}(a)\backslash i} B_{j\to a}^{(l)} = B_{s_a\to a} \prod_{j\in V(a)\backslash i} B_{j\to a}^{(l)} \\ &= (-1)^{s_a} tanh(\gamma_a) \prod_{j\in V(a)\backslash i} B_{j\to a}^{(l)}. \end{aligned} \tag{46}$$

Then, we have

$$\begin{aligned} LR_{a\to i}^{(l)} &= \log_e\left(\frac{1 - S_{a\to i}^{(l)}}{1 + S_{a\to i}^{(l)}}\right) = -2tanh^{-1}(S_{a\to i}^{(l)}) \\ &= 2(-1)^{s_a+1} tanh^{-1}\left(tanh(\gamma_a) \prod_{j\in V(a)\backslash i} B_{j\to a}^{(l)}\right) \end{aligned} \tag{47}$$

where the second equality uses the equation

$$tanh^{-1}(x) = \frac{1}{2}\log_e\left(\frac{1+x}{1-x}\right)$$

and the third equality is obtained by substituting (46) into the second equation. (Note that $y = tanh^{-1}(x)$ is an odd function.) So the **message update rule for check nodes** is

$$LR_{a\to i}^{(l)} = 2(-1)^{s_a+1} tanh^{-1}\left(tanh(\gamma_a) \prod_{j\in V(a)\backslash i} B_{j\to a}^{(l)}\right). \tag{48}$$

By replacing $R_i^{(l)}$ with $LR_i^{(l)}$ in (44), we have

$$B_i^{(l)} = \frac{1 - R_i^{(l)}}{1 + R_i^{(l)}} = \frac{1 - e^{LR_i^{(l)}}}{1 + e^{LR_i^{(l)}}} = -tanh\left(\frac{LR_i^{(l)}}{2}\right),$$

where the third equality uses the equation

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}.$$

By doing the same on $R_{i\to a}^{(l)}$, we have the **bias update rule of information bits**

$$B_i^{(l)} = -tanh\left(\frac{LR_i^{(l)}}{2}\right), \quad B_{i\to a}^{(l)} = -tanh\left(\frac{LR_{i\to a}^{(l)}}{2}\right). \tag{49}$$

By substituting (40) and (41) into the first equation in (47), we have the initialization of $LR_{a\to i}^{(0)}$

$$LR_{a\to i}^{(0)} = \log_e\left(\frac{1 - B_{s_a\to a}}{1 + B_{s_a\to a}}\right). \tag{50}$$

ACKNOWLEDGMENT

The authors would like to thank D. J. C. Mackay for providing some irregular matrices that they used in the simulations.

REFERENCES

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. San Mateo, CA, USA: Margan Kaufmann, 2008.
[2] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[3] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 705–720, Dec. 2010.
[4] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.
[5] R. Crandall. *Some Notes Steganography*. Accessed: Dec. 15, 2023. [Online]. Available: http://dde.binghamton.edu/download/Crandall_matrix.pdf
[6] A. Westfeld, "High capacity despite better steganalysis (F5—A steganographic algorithm)," in *Proc. Int. Workshop Inf. Hiding*. New York, NY, USA: Springer-Verlag, 2001, pp. 289–302.
[7] M. Van Dijk and F. Willems, "Embedding information in grayscale images," in *Proc. 22nd Symp. Inf. Commun. Theory*, Jan. 2001, pp. 147–154.
[8] D. Schönfeld and A. Winkler, "Embedding with syndrome coding based on BCH codes," in *Proc. 8th Workshop Multimedia Secur.*, Sep. 2006, pp. 214–223.
[9] R. Zhang, V. Sachnev, and H. J. Kim, "Fast BCH syndrome coding for steganography," in *Proc. Int. Workshop Inf. Hiding*, 2009, pp. 48–58.
[10] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3923–3935, Oct. 2005.
[11] J. Fridrich, M. Goljan, and D. Soukal, "Efficient wet paper codes," in *Proc. Int. Workshop Inf. Hiding*, 2005, pp. 204–218.
[12] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2006, pp. 314–327.
[13] W. Li, W. Zhang, L. Li, H. Zhou, and N. Yu, "Designing near-optimal steganographic codes in practice based on polar codes," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 3948–3962, Jul. 2020.
[14] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
[15] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.
[16] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, 1963. [Online]. Available: http://justice.mit.edu/people/gallager.html
[17] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," *Proc. SPIE*, vol. 6505, Feb. 2007, Art. no. 650502.
[18] M. Me'zard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," *Science*, vol. 297, no. 5582, pp. 812–815, Aug. 2002.
[19] E. Maneva, E. Mossel, and M. J. Wainwright, "A new look at survey propagation and its generalizations," in *Proc. 16th Annu. Symp. Discrete Algorithms (SODA)* 2005, pp. 1089–1098.

[20] M. J. Wainwright and E. Maneva, "Lossy source coding by message-passing and decimation over generalized codewords of LDGM codes," in *Proc. Int. Symp. Inf. Theory*, Sep. 2005, pp. 1493–1497.
[21] E. Martinian and M. J. Wainwright, "Analysis of LDGM and compound codes for lossy compression and binning," 2006, *arXiv:cs/0602046*.
[22] M. J. Wainwright, E. Maneva, and E. Martinian, "Lossy source compression using low-density generator matrix codes: Analysis and algorithms," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1351–1368, Mar. 2010.
[23] T. Filler and J. Fridrich, "Binary quantization using belief propagation over factor graphs of LDGM codes," in *Proc. 45th Annu. Allerton Conf. Commun., Control, Comput.*, Allerton, IL, USA, Sep. 2007, pp. 495–501.
[24] A. Braunstein, F. Kayhan, and R. Zecchina, "Efficient LDPC codes over GF(q) for lossy data compression," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2009, pp. 1978–1982.
[25] D. Castanheira and A. Gameiro, "Lossy source coding using belief propagation and soft-decimation over LDGM codes," in *Proc. 21st Annu. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2010, pp. 431–436.
[26] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth tanner graphs," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 1, Nov. 2001, pp. 995–1001.
[27] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
[28] S. Gel'fand and M. Pinsker, "Coding for channel with random parameters," *Problems Control Inf. Theory*, vol. 9, no. 1, pp. 19–31, 1980.
[29] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 2006.
[30] T. Filler and J. Fridrich, "Using non-binary embedding operation to minimize additive distortion functions in steganography," in *Proc. 2nd IEEE Int. Workshop Inf. Forensics Secur.*, Seattle, WA, USA, Dec. 2010.
[31] X. Zhang, W. Zhang, and S. Wang, "Efficient double-layered steganographic embedding," *Electron. Lett.*, vol. 43, no. 8, pp. 482–483, 2007.
[32] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.
[33] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[34] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[35] A. Winkler, "Advances in syndrome coding based on stochastic and deterministic matrices for steganography," Ph.D. dissertation, Dept. Comput. Sci., Technische Universität Dresden, Dresden, Germany, 2011.
[36] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[37] E. Arkan, "A performance comparison of polar codes and Reed–Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.
[38] Q. Yao, W. Zhang, and N. Yu, "Optimality of polar codes in additive steganography under constant distortion profile," in *Proc. 14th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nov. 2022, pp. 404–408.
[39] S. Kudekar and R. Urbanke, "Lower bounds on the rate-distortion function of individual LDGM codes," in *Proc. 5th Int. Symp. Turbo Codes Rel. Topics*, Lausanne, Switzerland, Sep. 2008, pp. 379–384.
[40] T. Richardson and R. Urbanke, *Modern Coding Theory* New York, NY, USA: Cambridge Univ. Press, 2008.

Qiyi Yao received the B.S. degree in computer science and technology from Sun Yat-sen University, Guangzhou, China, in 2020. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China, Hefei, China.

His research interests include information theory, coding theory, and their applications towards communications and security (covert communication).

Weiming Zhang received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding coding and multimedia security.

Kejiang Chen (Member, IEEE) received the B.S. degree from Shanghai University (SHU) in 2015 and the Ph.D. degree from the University of Science and Technology of China (USTC) in 2020. He is currently an Associate Research Fellow with the USTC. His research interests include information hiding and deep learning security.

Nenghai Yu received the B.S. degree from the Nanjing University of Posts and Telecommunications in 1987, the M.E. degree from Tsinghua University in 1992, and the Ph.D. degree from the University of Science and Technology of China in 2004. He is currently a Professor with the University of Science and Technology of China. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.