

Neural Rate-Adaptive LDPC Decoding for the Slepian-Wolf Problem

BRENT DE WEERDT ^{1,2} (Graduate Student Member, IEEE)
AND NIKOS DELIGIANNIS ^{1,2} (Senior Member, IEEE)

¹Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, B-1050 Brussels, Belgium

²imec, B-3001 Leuven, Belgium

CORRESPONDING AUTHOR: BRENT DE WEERDT (email: brent.de.weerdt@vub.be).

This work was supported in part by the FWO Flanders Ph.D. Fellowship Strategic Basic Research under Grant 1S44523N, and in part by the ERC under Consolidator Grant IONIAN under Grant 101171240, DOI: 10.3030/101171240.

ABSTRACT Distributed source coding (DSC) enables efficient compression of correlated sources by performing independent encoding and joint decoding. Slepian-Wolf coding plays a central role in DSC, as it allows near-lossless compression of correlated data at rates asymptotically approaching the joint entropy. Traditional approaches to constructive Slepian-Wolf coding employ algebraic binning techniques using channel codes, with decoding performed via algorithms such as belief propagation (BP). While effective, these methods suffer from high computational cost and suboptimal compression performance. We propose the first constructive learned Slepian-Wolf decoder for rate-adaptive coding, using a single multi-rate Transformer model. The architecture is inspired by neural channel decoding, but addresses the unique challenges of syndrome-based Slepian-Wolf coding based on side information. Furthermore, we incorporate the proposed rate-adaptive neural Slepian-Wolf decoder into a novel neural layered Wyner-Ziv code design for the quadratic Gaussian case and into a new layered Wyner-Ziv design for distributed stereo image coding. For Slepian-Wolf coding of binary sources, our neural decoder improves the compression performance over traditional BP decoding by up to 11%. In our monolithic and layered Wyner-Ziv designs, we are between 0.05 and 0.2 bits/sample away from the estimated ideal rate bound, while entropy coding needs an additional 0.4-1.2 bits/sample compared to the proposed Slepian-Wolf codec. Moreover, our stereo image coding design reduces the coding rate by 9-19% in low rate settings compared to the state-of-the-art with minimal loss in image quality. Finally, the proposed decoder is 15 times faster than BP decoding on a GPU.

INDEX TERMS Neural Slepian-Wolf coding, neural LDPC Accumulate codes, transformers, Wyner-Ziv coding, layered coding.

I. INTRODUCTION

Distributed source coding (DSC) addresses the compression of correlated sources that are encoded separately but decoded jointly, enabling low-complexity and power-limited encoders while still exploiting inter-source correlation at the decoder. This setting arises naturally in distributed sensing and imaging systems such as multi-camera or multi-view image compression [1] and cooperative perception for autonomous vehicles [2] as well as data-parallel distributed training of deep-learning models [3], where multiple terminals observe related signals and must transmit compressed data independently.

A central setting in DSC is to encode a data source X in isolation at the encoder side, while the decoder will have access to side information Y that is correlated with X . A theoretical framework was first developed for the noiseless compression of discrete independent and identically distributed (i.i.d.) sources [4], named Slepian-Wolf (SW) coding. It is based on a *random binning* argument: all possible source realizations are randomly put into different bins, and only the index of the bin in which the source data belongs is transmitted. The decoder can then select the most likely source realization from the received bin, given the side information. The result was later

extended to lossy compression of continuous sources, yielding the Wyner-Ziv rate-distortion bound [5]. Wyner-Ziv (WZ) coding is often realized by quantization followed by compression of the quantization indices with SW coding. Under certain conditions [4], [5], [6], [7], the minimum achievable rate for DSC is equal to the minimum rate of a system where both the encoder and decoder have access to Y . Furthermore, successive refinement for WZ coding [8] allows quality scalability by sending information in stages, which can be done without rate loss in certain cases.

The random binning argument used in the information-theoretic proofs is non-constructive [4], [5], and it took decades to design practical code constructions. The first constructive SW designs realized algebraic binning with channel codes such as low-density parity-check (LDPC) [9] or Turbo [10] codes. WZ coding was implemented using SW coded scalar quantization [11], nested lattice quantization [12], and trellis coded quantization [13], [14]. Layered WZ code design based on nested scalar quantization, bit-plane coding, and LDPC code-based SW coding has also been proposed for successive refinement [15].

An additional challenge in DSC is determining the SW compression rate at the encoder. If the correlation statistics between X and Y are well known, we can choose a suitable code beforehand. However, when these statistics are not known or are prone to change over time, rate-adaptive schemes with a feedback channel can be used. With this approach, SW decoding can be performed at the lowest possible rate, depending on the side information quality. Rate-adaptivity was originally implemented with punctured Turbo codes [16] and later on with LDPC accumulate (LDPCA) codes [17].

While this work focuses on decoder-level rate-adaptivity, which achieves the lowest possible rate for each codeword, a feedback channel can cause delays and a need for an increased buffer size at the encoder to store the syndrome bits. However, in the traditional WZ literature there are works that fully eliminate the feedback channel; this is achieved by increasing the encoder complexity to perform an estimation of the side information and the rate [18]. Moreover, there are hybrid techniques that allow a feedback channel but optimize it to allow certain delay from the application [19].

Traditional DSC constructions can approach information-theoretic limits in wellstudied settings such as the doubly symmetric binary channel or the quadratic Gaussian case. However, they struggle when the source statistics are complex or unknown, as in distributed image [20] or video [21] coding. Recently, researchers have been increasingly replacing handcrafted codecs with neural networks that learn both the encoder and decoder, and compress the resulting quantized latent representations with entropy coding [22]. Additionally, joint source-channel coding papers like [23], [24], [25] use an analog approach, sending features directly over a noisy channel. In contrast, our work focuses solely on distributed data compression, leaving the transmission to an outer channel code. This is justified since the source-channel separation principle holds in WZ coding over orthogonal links [26].

Moreover, the performance of an analog transmission scheme degrades as the ratio of maximum energy to minimum energy of the source component increases [27].

Regarding research on learned distributed compression, the authors of [28] adapted the single image compression model from [22] by fusing a side information image into the decoder network through cross-attention modules. The latent features generated by the encoder are quantized and entropy coded. In a similar fashion, [29] presented a vector quantized variational autoencoder (VQ-VAE), improving over [28]. However, these models do not explicitly make use of SW coding. Indeed, when testing the method from [29] on synthetic data like i.i.d. Bernoulli sources, it exhibits higher error rates than algorithmic DSC solutions like DISCUS [13]. This shows the inherent difficulty in learning good distributed source codes with neural networks. Another line of work trains simple feedforward neural networks for Wyner-Ziv coding of quadratic Gaussian sources, showing they can learn a form of “binning” behavior reminiscent of SW random binning [30]. The work in [31] extends this idea with recurrent neural networks (RNNs) that implement layered Wyner-Ziv coding for successive refinement. Both studies, however, estimate the required rate using an *ideal* SW coder rather than an explicit code construction, so the resulting schemes remain non-constructive. Moreover, with the exception of [31], none of the above-mentioned neural DSC models allow for any kind of quality scalability, nor does any work consider the selection of a proper coding rate, e.g., with rate-adaptivity. There is one work that tries to estimate a good rate for each modality to minimize transmission time [32], but it does not allow changing the rate dynamically during transmission to achieve the optimal rate for the current data sample.

To address the current shortcomings in neural SW coding, we present the following contributions:

- We introduce the first data-driven Slepian-Wolf framework, based on learned channel decoding. Since good channel codes often serve as good SW source codes [6], [33], it is natural to leverage modern channel-coding advances for DSC. Inspired by [34], we present a Transformer-based SW decoder in this paper. The model-free approach allows the network to learn efficient decoding strategies for several data statistics, achieving better compression performance compared to traditional belief propagation decoding in various scenarios.
- Our SW coding design is also made rate-adaptive, using a single Transformer network to decode all possible code rates. This allows to select the right rate for each data block using a feedback channel, realizing the best possible compression rates.
- When a Graphics Processing Unit (GPU) is available, the proposed learned SW decoder decodes 15 times faster than belief propagation, on top of achieving better compression performance.
- We extend our recent coding scheme in [31] and the original monolithic scheme in [30] by incorporating the proposed neural Slepian-Wolf code, leading to

constructive neural-network-based monolithic and layered Wyner-Ziv coding designs. Earlier work in [30], [31] was not constructive as it assumed an ideal SW codec.

- Our SW decoder is used to design a neural SW coded VQ-VAE system for distributed stereo image coding. Unlike previous works that use entropy coding [28], [29], our system is the first SW coded solution, fully exploiting data correlations and achieving better compression performance at low code rates.

The rest of this paper is organized as follows: Section II provides background on SW coding, LDPC decoding, rate adaptivity, and WZ coding. Section III presents the proposed rate-adaptive Transformer architecture for LDPCA decoding. Section IV describes our SW coded WZ frameworks and our SW coded distributed coding system for stereo image coding. Section V provides the experiments on SW coding, WZ coding, and stereo image coding, and Section VII concludes the work.

II. BACKGROUND AND RELATED WORK

This section gives an overview of DSC theory and implementations, starting with the Slepian-Wolf theorem for lossless coding. We then provide the theoretical bound for lossy compression, provided by Wyner and Ziv, which is extended to successive refinement through layered WZ coding. In Section II-B, we describe neural DSC approaches, mainly for stereo image coding, which do not implement SW coding of the quantization indices. Section II-C discusses the duality between SW coding and channel coding, allowing us to leverage existing work on neural channel decoding to create the proposed architecture for SW decoding.

A. DISTRIBUTED SOURCE CODING

1) SLEPIAN-WOLF CODING

Consider the compression of two correlated, discrete, i.i.d. random sources X and Y , encoded at rates R_X and R_Y respectively. When both sources are available at the encoder and decoder, coding can achieve the optimal rate region defined by:

$$R_X \geq H(X|Y), R_Y \geq H(Y|X), R_X + R_Y \geq H(X, Y), \quad (1)$$

where $H(X|Y)$ is the conditional entropy of X given Y , and $H(X, Y)$ is the joint entropy of X and Y . Surprisingly, Slepian and Wolf proved that this bound is still achievable when the sources are encoded separately and decoded jointly [4]. This result has also been extended to arbitrary ergodic processes [35].

Achievability of the Slepian-Wolf rate region is proven via an asymptotic (i.e., the source dimensionality $n \rightarrow \infty$) random binning argument. The probability of a decoding error using this binning argument is asymptotically very small [4]. In constructive implementations, however, n is finite, which leads to finite-block-length losses, that is, a non-zero gap to the SW bound and/or non-zero decoding error probability.

For a given block length and error probability, more specific bounds can be derived [36].

Constructive designs use channel codes to do algebraic binning, achieving compression rates close to the Slepian-Wolf bound. In this paper we use syndrome-based coding, originally proposed in [37]. Assume the correlated sources X and Y are binary. Syndrome coding maps the vector $\mathbf{x}$ to a syndrome $\mathbf{s} = \mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is an $r \times n$ parity-check matrix. The decoder receives the syndrome and applies a decoding function to $\mathbf{s}$ and $\mathbf{y}$ to get the error vector $\mathbf{e}$. Finally, the sequence is reconstructed as $\hat{\mathbf{x}} = \mathbf{y} \oplus \mathbf{e}$, where $\oplus$ is the XOR operation. Syndrome-based coding has been implemented with, e.g., trellis codes [13] and LDPC codes [9].

An important issue in SW coding is rate control when the correlation between X and Y is not well known and/or not stationary. The work in [17] proposed LDPCA codes for rate-adaptive coding. The encoder begins transmission with a short syndrome corresponding to a high-compression (low-rate) code. If decoding fails, signaled by the decoder through a feedback channel, additional syndrome bits are sent incrementally. This process continues until successful decoding is achieved.

2) WYNER-ZIV CODING

Wyner-Ziv coding is a generalization of the asymmetric Slepian-Wolf scenario, where $\mathbf{x}$ is encoded in a lossy manner. The decoded sequence $\hat{\mathbf{x}}$ has a maximum average distortion $D = \mathbb{E}[d(\mathbf{x}, \hat{\mathbf{x}})]$, with d being a distortion metric like the mean squared error (MSE). The WZ rate-distortion function is then given by [5]:

$$R_{WZ}(D) = \inf_{p(u|x)} I(X; U) - I(Y; U), \quad (2)$$

where the infimum is over all auxiliary variables U satisfying the Markov chains $U \leftrightarrow X \leftrightarrow Y$ and $X \leftrightarrow (U, Y) \leftrightarrow \hat{X}$. It is proven that under an MSE distortion metric and sources defined as $X = Y + N$, where Y is arbitrarily distributed and N is independent Gaussian noise, there is no rate loss compared to the case where the encoder also has access to the side information Y [6].

WZ coding is typically realized by sample quantization and SW coding of the quantization indices using channel codes. The DISCUS framework [13] uses scalar quantization or trellis coded quantization, combined with scalar or trellis-based coset codes for SW coding. Other approaches use nested lattice codes followed by SW coding with LDPC codes [38] or turbo-trellis WZ codes [39].

3) LAYERED WYNER-ZIV CODING

Successively refined WZ coding extends the traditional WZ framework by encoding a source in multiple stages, where each stage refines the previous reconstruction under progressively stricter distortion constraints. Each stage consists of an encoder ϕ_k and decoder ψ_k :

$$\phi_k : \mathcal{X}^n \rightarrow \{1, 2, \dots, C_k\}, \quad (3)$$

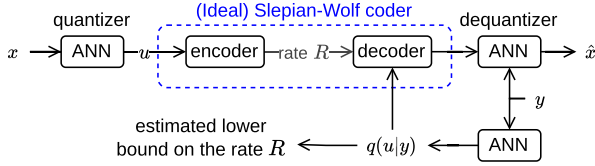


FIGURE 1. Overview of the learned Wyner-Ziv scheme from [30], which assumes an ideal Slepian-Wolf coder.

$$\psi_k : \{1, \dots, C_1\} \times \dots \times \{1, \dots, C_k\} \times \mathcal{Y}^n \rightarrow \hat{\mathcal{X}}^n, \quad (4)$$

such that $\mathbb{E}[d(\mathbf{x}, \hat{\mathbf{x}}_k)] \leq D_k$, with C_k the alphabet size of encoder k . WZ coding is said to be successively refinable when the rate bound at each stage is equal to the bound of monolithic WZ coding at the same distortion level D_k . Sources $X = Y + N$, where Y is arbitrarily distributed and N is independent Gaussian noise, are successively refinable [15]. The authors of [15] proposed a layered WZ coding implementation, where the indices of the quantized source vectors are SW encoded bitplane by bitplane using LDPC codes.

B. NEURAL DISTRIBUTED SOURCE CODING

Constructing coding schemes that approach the theoretical WZ or SW bound is hard, especially for data sources with complex statistics. Recent work [30] leverages artificial neural networks (ANNs) for WZ coding. They learn a one-shot encoder and decoder, as well as a probability model for rate estimation. Each sample in the vector $\mathbf{x}$ is quantized independently by an ANN. The quantization indices $\mathbf{u}$ can then be further compressed with entropy coding or SW coding, using the learned probability model $q(u)$ or $q(u|y)$ respectively. After decompression of $\mathbf{u}$ at the receiver side, the source vector is reconstructed from $\mathbf{u}$ and $\mathbf{y}$ using the decoder ANN. An illustration of this approach is shown in Fig. 1. However, the work in [30] assumes an ideal (non-constructive) entropy coder or SW coder for the compression of $\mathbf{u}$.

There are also works that tackle distributed image coding with neural networks, but they do not consider a SW code either. Both [1] and [28] adapt an image compression network, respectively using [22] and [40] as backbone, adding in extra modules and modifications to incorporate the side information. Similarly, the authors of [29] train a VQ-VAE and a Transformer-based autoregressive prior. After encoding and vector quantization of the source image, the quantization indices are compressed with entropy coding using the learned prior. The decoder network then reconstructs the image from the quantized features and a side information image. The authors of [25] rely on a third party to estimate homography parameters, violating the WZ constraint that the encoder does not receive any data derived from the side information. Additionally, they use an analog joint source-channel coding pipeline with convolutional neural network (CNN) autoencoders, whereas our method focuses on data compression with rate-adaptive LDPCA codes for better performance under correlation uncertainty.

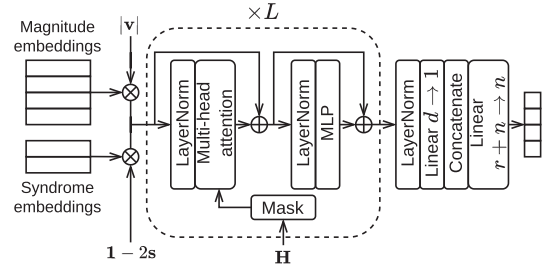


FIGURE 2. The error correction code Transformer (ECCT) architecture from [34]. The values of $|\mathbf{v}|$ and $\mathbf{s}$ depend on whether the model is used for channel decoding or Slepian-Wolf decoding.

These prior works use entropy coding to compress the quantized features. However, as long as there is correlation between the quantization indices and the side information, which is generally the case, SW coding should be able to achieve a lower coding rate than entropy coding. However, SW coding is more complex than entropy coding, making it challenging to implement in practice. To the best of our knowledge, there exists no constructive implementation of a fully-learned distributed coding framework that employs SW coding.

C. SW CODING AND CHANNEL CODING DUALITY

The work in [6] showed the duality between SW coding and channel coding, and [33] proved that each SW coding problem is equivalent to a channel coding problem under density evolution. Neural decoders for channel codes have recently been proposed to improve performance and reduce the decoding time compared to traditional decoders. There are works using learnable weights [41], recurrent neural networks (RNNs) [42], graph neural networks [43], and Transformer networks [34]. In this work, we design a multi-rate SW decoder based on the Transformer from [34], which we describe below.

The error correction code Transformer (ECCT) [34] (see Fig. 2) assumes that a codeword $\mathbf{u}$ is modulated with binary phase shift keying (BPSK) and sent over an additive white Gaussian noise channel, resulting in a noisy received vector $\mathbf{v} = \mathbf{1} - 2\mathbf{u} + \mathbf{n}$ with $\mathbf{n}$ being Gaussian noise. The first part of the input to the ECCT consists of the absolute values of the received vector $|\mathbf{v}|$, and the other part is the syndrome of the binarized received vector: $\mathbf{s} = \mathbf{H}\mathbf{v}_{bin} = \frac{1}{2}\mathbf{H}(\mathbf{1} - \text{sign}(\mathbf{v}))$. Each magnitude value in $|\mathbf{v}|$ and each syndrome bit in $\mathbf{s}$ becomes its own Transformer token through multiplication with learned embeddings. The token sequence is then processed with Transformer encoder blocks. The Transformer self-attention is masked based on the Tanner graph of the parity-check matrix $\mathbf{H}$. Finally, the tokens are transformed by linear projection and concatenation into a single vector of length n . The target output of the network is the difference between $\mathbf{v}_{bin}$ and the original codeword $\mathbf{u}$. Denoting the Transformer by f_θ , the full decoding operation can be written as

$$\hat{\mathbf{u}} = \mathbf{v}_{bin} \oplus f_\theta(|\mathbf{v}|, \mathbf{H}\mathbf{v}_{bin}). \quad (5)$$

There are several challenges in the design of a SW decoder from a channel decoder. First of all, the SW decoder only receives a target syndrome, and the rest of the information should be derived from the side information and the correlation model. This requires changes in the decoding strategy. Furthermore, SW coding requires rate-adaptivity to cope with varying correlation between the sources. Whereas ECCT only supports decoding of a single code rate, our goal is to train a model that can decode the full rate range, without having to train, store, and load separate networks for each rate.

III. MULTI-RATE NEURAL SLEPIAN-WOLF CODING

This section describes our proposed multi-rate decoder network for SW coding. We first derive the input processing necessary to adapt the ECCT model for channel coding [34] towards an architecture for LDPCA SW decoding. Then, our core contribution is the proposed multi-rate Transformer decoder, which allows decoding at all rate settings of an LDPCA code with a single neural network. Finally, we discuss how to generate LDPCA codes with small block sizes for this architecture.

A. SW DECODING WITH TRANSFORMERS

The proposed design is inspired by Wyner's scheme [37], where a parity-check code is adapted for SW coding: Consider a binary communication channel with input an n -length binary vector $\mathbf{u}$ and output $\mathbf{v}_{bin} = \mathbf{u} \oplus \mathbf{e}$. In channel coding, the codeword $\mathbf{u}$ is constructed such that the syndrome is:

$$\mathbf{s} = \mathbf{H}\mathbf{v}_{bin} = \mathbf{H}(\mathbf{u} \oplus \mathbf{e}) = \mathbf{H}\mathbf{e}, \quad (6)$$

where $\mathbf{H}$ is an $r \times n$ parity-check matrix. This results in a channel rate of $1 - \frac{r}{n}$. A decoding function f can then be applied to $\mathbf{s}$ to derive the error vector $\mathbf{e}$ and reconstruct the codeword as $\hat{\mathbf{u}} = \mathbf{v}_{bin} \oplus f(\mathbf{H}\mathbf{e})$, analogous to the soft decoding in (5). For any code rate up to the channel capacity C , the decoding error probability can be made arbitrarily small for $n \rightarrow \infty$.

For SW coding, take the same matrix $\mathbf{H}$ and decoding function f . The SW encoder sends the r -length syndrome $\mathbf{s}_{target} = \mathbf{H}\mathbf{u}$ to the decoder, where $\mathbf{u}$ is now arbitrary source data. At the decoder, we generate an estimation $\mathbf{v}_{bin}$ of $\mathbf{u}$ using the side information $\mathbf{y}$ and calculate

$$\mathbf{s}_{target} \oplus \mathbf{H}\mathbf{v}_{bin} = \mathbf{H}(\mathbf{u} \oplus \mathbf{v}_{bin}) = \mathbf{H}\mathbf{e}. \quad (7)$$

In this case, we can also decode $\mathbf{e}$ and reconstruct with $\hat{\mathbf{u}} = \mathbf{v}_{bin} \oplus f(\mathbf{H}\mathbf{e})$, with the same arbitrarily low error probability for $n \rightarrow \infty$. In SW coding, the correlation between $\mathbf{u}$ and $\mathbf{y}$ is considered as a virtual channel with capacity C . Hence the SW rate $\frac{r}{n}$ must be $\geq 1 - C$.

In our approach, the estimate of $\mathbf{u}$ from side information is represented with log-likelihood ratios (LLRs):

$$v_i = \log \left(\frac{p(u_i = 0|y_i)}{p(u_i = 1|y_i)} \right), \quad (8)$$

where the priors $p(u|y)$ are estimated by a separate neural network $q_\zeta(u|y)$ (see Section IV-A and Section IV-B2), or

determined analytically for known correlation channels (see Section V-B). Binarizing $\mathbf{v}$ yields the estimated code block $\mathbf{v}_{bin}$, while the magnitude of $\mathbf{v}$ provides the confidence for each bit.

In summary, the ECCT decoding procedure [34] can be changed to SW decoding as follows:

$$\hat{\mathbf{u}} = \mathbf{v}_{bin} \oplus f_\theta(|\mathbf{v}|, \mathbf{H}\mathbf{v}_{bin} \oplus \mathbf{s}_{target}), \quad (9)$$

where $\mathbf{v}$ is derived using the LLRs in (8). The ECCT architecture in Fig. 2 remains unmodified, but now $\mathbf{s} = \mathbf{H}\mathbf{v}_{bin} \oplus \mathbf{s}_{target}$. In this work, we always train the ECCT in the SW regime for simplicity. Similar to [34], we aim to minimize the binary cross-entropy between the target error and the output of the decoder. Directly minimizing the average bit error rate actually would lead to the soft bit error loss as derived in [41], which is subtly different from the cross-entropy loss. However, we observed no significant differences in training dynamics between the two loss functions, so for simplicity we used binary cross-entropy.

B. MULTI-RATE SW TRANSFORMER

One of our main contributions is the design of a multi-rate neural decoder. To facilitate rate-adaptive compression, the Transformer should be able to decode multiple code rates, without requiring a separately trained model for each rate. Therefore, we adapt the architecture to accommodate all rate settings of an LDPCA code.

We start with a Transformer model for an n -bit LDPCA code at rate 1, with n token embeddings for the magnitude bits, as well as n embeddings for the syndrome bits. The embeddings are processed by L Transformer encoder blocks, where the self-attention mask is based on the $n \times n$ parity-check matrix. The $2n$ tokens are then projected back to a soft n -bit output.

Rate adaptivity in LDPCA codes is achieved by sending a number of accumulated syndrome bits, which combines multiple syndrome bits into one as illustrated in the left part of Fig. 3. To reflect this in the Transformer, the embeddings of merged syndrome bits are averaged into a single token: $\mathbf{e}_{i,merged} = \frac{1}{l_i - l_{i-1}} \sum_{j=l_{i-1}+1}^{l_i} \mathbf{e}_j$, where l_i are the indices of the received accumulated syndrome bits. Since each rate corresponds to a different parity-check matrix, the self-attention mask is updated accordingly, while the remaining encoder structure is unchanged. At the output stage, the number of syndrome tokens depends on the code rate. To preserve a common projection layer for all rates, zero padding is inserted at positions corresponding to merged syndrome bits, allowing for a fixed output projection layer. These modifications together enable training across all code rates without blowing up the size of the decoder model.

The described modifications to the architecture are highlighted in the right part of Fig. 3 in blue. In the remainder of the paper, we refer to this multi-rate SW Transformer decoder as SW-ECCT.

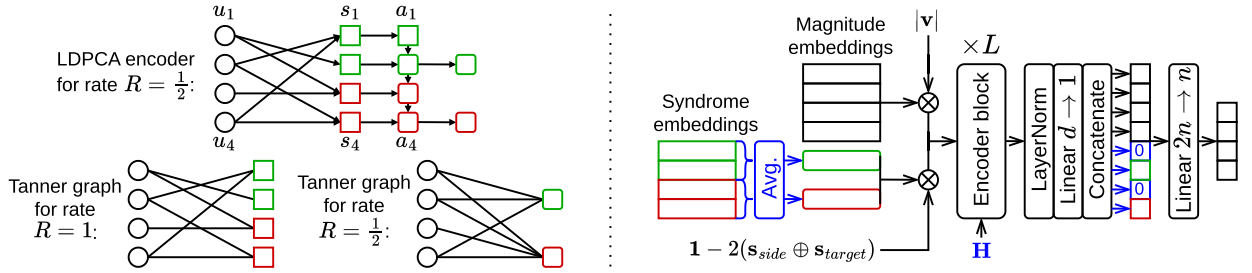


FIGURE 3. Illustration of the proposed multi-rate Slepian-Wolf ECCT decoder (SW-ECCT) for a 4-bit code at rate $R = \frac{1}{2}$. The left part shows the LDPCA encoding of codeword bits u_i into syndrome bits s_i and accumulated syndrome bits a_i , as well as the Tanner graphs when decoding with a_1, a_2, a_3, a_4 ($R = 1$) and with a_2, a_4 only ($R = \frac{1}{2}$). On the right, the multi-rate decoder is presented, where the blue color highlights the modifications compared to the single-rate architecture in Fig. 2.

C. LDPCA CODE DESIGN FOR SMALL BLOCK SIZES

Since SW-ECCT uses a token for each codeword bit, the computational cost is quadratic with respect to the code's block size. Existing LDPCA codes [17] have block lengths (e.g., 396, 6336) that lead to a high computational cost in training and inference. We therefore create LDPCA codes with smaller block sizes, using standard algorithms available in the literature [44], [45], [46]. As observed in [17], using smaller block lengths has no significant effect on rate-adaptive compression performance.

We first generate a low-rate mother code with the value-node degree distribution $\lambda(x) = 0.3x^2 + 0.4x^3 + 0.3x^4$ from [17], using the constrained progressive edge-growth method [44], [45]. Next, the mother code is successively split according to Algorithm 2 in [46], while making sure to increase the rank of the matrix with every split, until an invertible square parity-check matrix is obtained. Finally, the transmission order of accumulated syndrome bits is determined using Algorithm 1 in [46].

The minimum number of LDPCA syndrome bits is 4, dictated by the value-node degree distribution. The desired number of rate settings then determines the block size. In our case, we opt for 16 rate settings, which results in a block size of 64 and a 4×64 mother code. This small code size makes it easier to learn, but also reduces the decoding complexity and hence the latency. Additionally, adapting to changing source statistics is faster with shorter codes. To show the versatility of our decoder, we also created a 128-bit code with the same 16 rate settings, as well as a 4×32 mother code, which is limited to 8 rates from $\frac{4}{32}$ to $\frac{32}{32}$. The performance of our decoder on these codes is compared in Section V-B.

Since the code construction relies on randomness, multiple candidate codes for each block size are generated and tested using rate-adaptive BP decoding on random data and a binary symmetric correlation channel, as described in Section V-B. The best performing code is then selected for all further experiments. As this selection procedure might introduce a bias, we also tested our proposed decoder on the worst and median code. Since we score the codes based on BP decoding, this bias works against us, and we actually observed bigger gains over BP by our decoder for the non-selected codes.

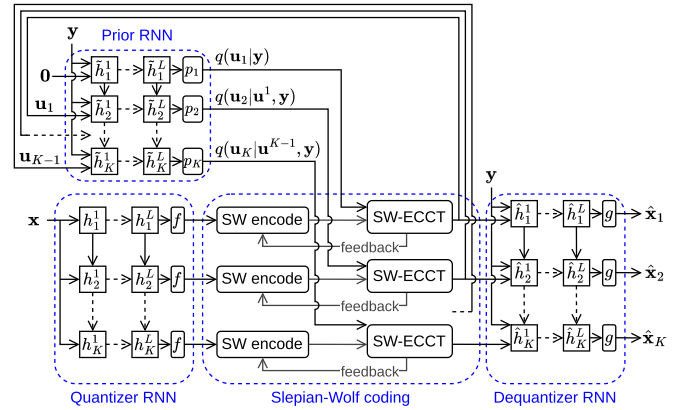


FIGURE 4. Schematic representation of the proposed learned layered Wyner-Ziv framework using neural Slepian-Wolf coding realized with the proposed SW-ECCT model.

IV. LEARNED LAYERED WYNER-ZIV CODING WITH NEURAL SLEPIAN-WOLF DECODER

Another core contribution of this paper is the integration of our multi-rate SW architecture into two learned WZ coding designs. This gives rise to the first constructive neural WZ designs with SW coding. The first design replaces the ideal rate estimation in the learned WZ quantizer scheme [31] with our SW-ECCT decoder, and the second applies our SW decoder to stereo image coding using a VQ-VAE [29].

A. LAYERED WZ CODING WITH SW-ECCT

We first describe our WZ design [31], and then show how to integrate our proposed neural SW design therein, leading to a constructive neural layered WZ coding scheme.

The system in our recent work [31] consists of three RNNs, namely, a quantizer, dequantizer and prior model, illustrated in Fig. 4. Each timestep of the RNNs constitutes a refinement stage k . The quantizer RNN outputs for every stage a distribution over the quantization indices $p_\theta(u_k | u^{k-1}, x)$ given the input sample x . The notation u^{k-1} indicates the series of previous quantization indices $u_1, u_2, \dots, u_{k-1}$, and u_0 is an empty sequence. Samples u^k are drawn from this distribution, SW encoded, and transmitted to the decoder. At the

decoder side, the prior RNN estimates the probability distribution $q_\zeta(u_k|u^{k-1}, y)$ conditioned on the side information y and previously decoded stages. Once the quantization indices u^k are decoded from the received syndrome bits and side information, they are passed to the dequantizer RNN together with y . The dequantizer outputs a reconstruction $\hat{x}_k = g_\eta(u^k, y)$ for each stage.

The differential rate $\Delta R_{WZ}(D_k)$ is defined as the rate required to send the quantization indices of refinement stage k to the decoder, while achieving a distortion not more than D_k :

$$\begin{aligned} \Delta R_{WZ}(D_k) &= R_{WZ}(D_k) - R_{WZ}(D_{k-1}) \\ &= I(X; U_k | U^{k-1}, Y). \end{aligned} \quad (10)$$

Similar to [30], we define a variational upper bound on the differential rate:

$$\Delta R_{WZ}(D_k) \leq \mathbb{E}_{x, y \sim p(x, y)} \left[\log \frac{p_\theta(u_k | u^{k-1}, x)}{q_\zeta(u_k | u^{k-1}, y)} \right]. \quad (11)$$

The hard constraint on the maximum distortion, $\mathbb{E}_x[d(x, \hat{x})]$, with d the mean square error, is relaxed using a Lagrange multiplier:

$$\mathcal{L}_k = \mathbb{E} \left[\log \frac{p_\theta(u_k | u^{k-1}, x)}{q_\zeta(u_k | u^{k-1}, y)} + \lambda d(x, g_\eta(u^k, y)) \right]. \quad (12)$$

The subscripts of the expectation operator are omitted for brevity. The whole system can then be trained end-to-end by minimizing the sum of per-stage losses $\sum_k \mathcal{L}_k$.

1) LOWER BOUND ON THE RATE

Prior works [30] and [31] do not implement a constructive SW codec, and instead take the cross-entropy

$$R_k = \mathbb{E} \left[-\log q_\zeta(u_k | u^{k-1}, y) \right] \quad (13)$$

as a lower bound on the rate, using the learned prior. The cross-entropy is estimated by averaging over $N = 10^5$ samples drawn from $p(x, y)$. The corresponding average distortion is calculated by passing the x samples through the encoder and assuming perfect transmission of the quantization indices u_k :

$$D_k = \frac{1}{N} \sum_{n=1}^N d(x_n, g_\eta(u_n^k, y_n)). \quad (14)$$

In a real implementation, the quantization indices should be compressed using SW coding. This will introduce additional overhead, resulting in a gap between the theoretical bound and the achieved rate. In the next paragraph, we will describe how our SW-ECCT coder is used to create a constructive system.

2) SW-ECCT

After training the RNN-based quantizer, dequantizer, and prior, their weights are frozen, and SW-ECCT is trained to decode the quantization indices u_k using the encoded syndrome bits and the LLRs derived from the prior. The same SW-ECCT model is trained to decode all stages. At inference, decoding starts from the lowest rate and progressively increases the

number of transmitted syndrome bits until successful decoding. Finally, the decoded indices are passed through the frozen dequantizer to reconstruct the samples.

B. LEARNED STEREO IMAGE CODING WITH VQ-VAE AND SW-ECCT

Furthermore, we propose a SW coded VQ-VAE system for distributed coding of stereo images, using the VQ-VAE from [29] as a baseline.

1) BASELINE

In [29], a source image $\mathbf{x}$ is encoded into a feature tensor with smaller spatial dimensions. Each vector of channel values is then quantized using a learned codebook. The indices of the codebook vectors $\mathbf{u}$ are compressed and sent to the decoder, which also has access to a correlated side information image $\mathbf{y}$. At the decoder, the codebook indices are decompressed, mapped back to codebook vectors, and combined with features extracted from $\mathbf{y}$ to reconstruct the image.

The uncompressed data rate is determined by the spatial downscaling factor and the codebook size, and can be further reduced by exploiting the non-uniform distribution of the codebook indices. Therefore, [29] also trains an autoregressive Transformer prior and uses it for entropy coding of the vector indices. While the authors of [29] claim to use arithmetic coding to compress the vector indices, in reality they report the theoretical cross-entropy, which is only a lower bound on the performance of a real entropy coder. For a fair comparison, we adapt their implementation to use a rANS coder [47].

2) PROPOSED APPROACH

To fully exploit the correlation between stereo images, we replace entropy coding of the VQ-VAE indices by SW coding. This is supported by distributed source coding literature showing that a combination of a good sphere covering code (i.e., vector quantizer) and a sphere packing code (e.g., channel code) leads to compression performance close to the WZ bound [38]. To this end, we introduce a conditional prior $q(\mathbf{u}|\mathbf{y})$ on the quantization indices, where $\mathbf{y}$ denotes the side information. This prior models the correlation between the source and the side information required for WZ coding [48]. The resulting architecture is shown in the top part of Fig. 5. The encoder, codebook, decoder, and side-information modules are trained as in [29] on the KITTI dataset, which is described in more detail in Section V-A. After VQ-VAE training, its weights are frozen, and the conditional prior is trained separately using the quantized source indices as targets. It consists of one 3 x 3 convolutional layer and two linear layers, joined by LeakyReLU activations. Similar to [30], [31], a relaxed Gumbel-softmax distribution is used to sample the prior during training. We also evaluated more complex prior models, but these did not perform as well as the three-layer one.

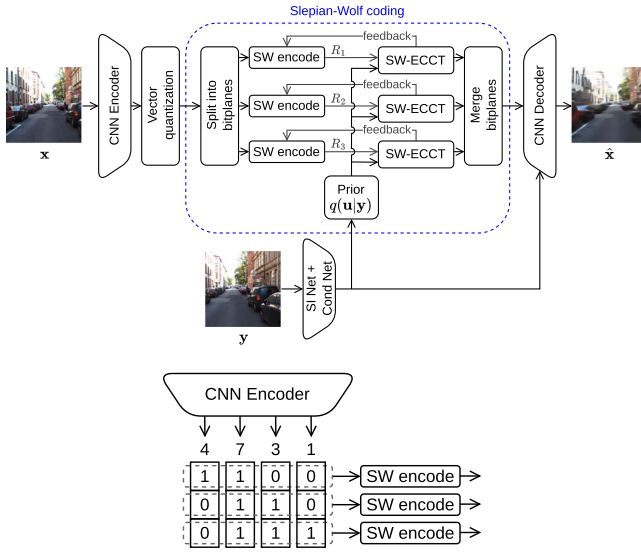


FIGURE 5. Top: diagram illustrating our proposed layered neural Slepian-Wolf coded VQ-VAE architecture for distributed stereo image compression. Our contribution as opposed to [29] is highlighted with the dashed blue rectangle. Bottom: Example visualizing the bitplane splitting operation.

Since the proposed SW decoder operates on binary data, the quantization indices are split into bitplanes, which are coded progressively. This bitplane splitting strategy is standard in DSC literature and avoids the added complexity of non-binary channel codes while retaining competitive performance [38], [48], [49], [50]. The splitting process is illustrated at the bottom of Fig. 5. The bitplane priors are derived from $q(u|y)$ by summing all the values $q(u = M_i|y)$ where M_i has a one (or zero) at the relevant bit location. For example, for a codebook size of 4, the prior for the least significant bit is given by

$$q(b_2 = 0|y) = q(u = 0|y) + q(u = 2|y),$$

$$q(b_2 = 1|y) = q(u = 1|y) + q(u = 3|y).$$

The LLRs can then be calculated with (8).

After training the VQ-VAE and the prior, the SW-ECCT decoder is trained and inserted into the full architecture as illustrated in Fig. 5, using rate-adaptive coding to gradually increase the rate until successful recovery of $\mathbf{u}$.

V. EXPERIMENTS AND RESULTS

In Section V-A we discuss the implementation of our SW-ECCT decoder, the learned WZ quantizers, and the VQ-VAE. The models are then evaluated in several settings. Section V-B considers SW coding of binary data with simple correlation statistics. Sections V-C and V-E study monolithic and layered learned WZ coding on quadratic Gaussian sources, while Section V-D evaluates learned BP [41] on monolithic WZ coding. Subsequently, Section V-F presents distributed stereo image coding experiments using the proposed SW-coded VQ-VAE. Finally, Sections V-G, V-H, and V-I report throughput

TABLE 1. Average Rate Reduction [bits/sample] of SW-ECCT Over BP Decoding With BSC and BIAWGN Correlations

Block size	BSC rate reduction	BIAWGN rate reduction
32 bits	0.0214	0.0280
64 bits	0.0131	0.0196
128 bits	0.0020	0.0029

comparisons, ablation studies, and the effect of SW decoding errors on image reconstruction quality, respectively.

A. IMPLEMENTATION DETAILS

Throughout all experiments, unless mentioned otherwise, SW-ECCT consists of 15 Transformer encoder blocks, with a token embedding dimension of 64, resulting in 767×10^3 learnable parameters. The network is trained similarly to [34], for 1000 epochs of 10^5 codewords each, using the Adam optimizer. The learning rate is initialized to 10^{-3} with a cosine decay down to 10^{-6} . However, whereas the ECC Transformer from [34] is trained on a single code rate, our SW-ECCT model is made to decode all the possible rate settings of the given LDPCA code. Each batch of 100 codewords is encoded at a different rate, cycling through all code rates.

For the BP implementation, the code from [41] is used without any parameter learning or adaptivity. All weights are set to 1, resulting in plain BP. Unless mentioned otherwise, we stop the BP algorithm after 100 iterations. Experiments were also conducted using adaptive learned BP in Section V-D. However, as no improvements over plain BP could be observed for decoding LDPCA codes, all other experiments only compare SW-ECCT with plain BP decoding.

In order to evaluate our decoder in rate-adaptive settings and simplify the comparison with BP in Sections V-B, V-C, V-E, V-F, and V-H, we assume perfect detection of SW decoding errors. While this isolates decoder performance, it also yields optimistic results: in practice, undetected decoding errors would cause erroneous reconstructions, and incorrectly flagged errors would increase rate and latency by requesting unnecessary additional bits. The oracle assumption is relaxed in Section V-I, where we report results with a practical SW error detection method for stereo image coding.

The layered WZ quantizer, dequantizer, and prior model all use RNNs of 3 hidden layers with 100 hidden units each, LeakyReLU activations, and linear layers at the output. All WZ networks, including the monolithic ones from [30], are trained using the Adam optimizer for 180 epochs. The initial learning rate is 10^{-3} , which is decreased by a factor of 0.3 every 40 epochs. One epoch consists of 2×10^5 randomly generated samples (x_n, y_n) , which are passed through the networks in batches of 1000. The output of a quantizer or quantizer stage is always binary to simplify the SW coding. For the layered quantizers we use 3 stages.

The training of the VQ-VAE and autoregressive prior on the KITTI dataset is identical to [29]. The VQ-VAE itself uses

the Adam optimizer for 1000 epochs with an initial learning rate of 2×10^{-4} with a linear warmup, cosine annealing decay, and a batch size of 32. The autoregressive prior is trained instead with an initial learning rate of 3×10^{-4} , a batch size of 128, and 100 epochs. Analogous to [51], stereo image pairs are sampled from the KITTI Stereo 2012 and 2015 datasets [52], [53]. These stereo images are taken with two RGB cameras mounted on a car 54 cm apart. As such, they provide slightly different viewpoints of the same street scene. The conditional prior used for SW coding is trained on the same KITTI dataset or on random data with BIAWGN correlation for 100 epochs with a batch size of 500 using the Adam optimizer. The initial learning rate of 10^{-3} is decreased by a factor of 0.3 every time the validation loss stops improving.

All rate-distortion curves in the paper are created using 10^5 codewords per bitplane or correlation level. Bit error rates (BERs) and block error rates (BLER) are calculated with at least 10^5 codewords, increasing the number of codewords until we observe at least 100 errors or until we hit 10^7 codewords.

The experiments on binary symmetric channel (BSC) correlation in the next section were run 5 times to measure the variance in the results. For BP, the standard deviation is 0.00036 bits/sample and is below the rounding error of the reported numbers. For SW-ECCT, the standard deviation is only slightly higher at 0.00076 bits/sample. To reduce the computational cost and because this variance does not affect any conclusions of the paper, all subsequent results are based on outputs from a single run.

B. RESULTS ON SLEPIAN-WOLF CODING

We first evaluate SW-ECCT in a controlled SW coding setting with an i.i.d. uniform binary source $X \sim \mathcal{B}(0.5)$ and side information Y generated through a BSC or BIAWGN correlation channel. For the BSC case, 9 bit error probabilities are chosen: $p_e \in \{0.013, 0.0311, 0.0532, 0.0794, 0.11, 0.146, 0.189, 0.243, 0.316\}$, resulting in a conditional entropy $H(X|Y)$ from 0.1 to 0.9. This covers the full range of useful correlation levels. For $H(X|Y) = 0.1$ and a block size of 64, 43% of the side information vectors will be exactly equal to $\mathbf{x}$, and for $H(X|Y) = 0.9$ the average rate is almost one, as around 70% of the blocks do not benefit from the side information and are transmitted uncompressed. For the BIAWGN correlation, the bits X are mapped to $\{+1, -1\}$ akin to BPSK modulation. Subsequently, Gaussian noise is added with 9 logarithmically spaced power levels σ^2 from 0.3 to 2. Similarly to the BSC, this covers both extremely high and low correlations.

A single SW-ECCT model is trained on all code rates and all 9 correlation levels of each correlation channel. As described in the implementation details, during training the SW-ECCT already cycles through the LDPCA code rates. To enable training on all 9 correlation levels, we add an inner loop that cycles through the correlation levels. This setup is

significantly more challenging than those in previous works, which train on a single code rate or a limited communication noise range [34], [42], [43].

At test time, rate-adaptive coding is used, increasing the rate for each data block until perfect decoding. We evaluate three LDPCA codes of different block sizes, namely 32, 64, and 128 bits. At moderate correlation levels, SW-ECCT can reduce the rate by up to 0.030 bits/sample (11%) for the BSC and 0.034 bits/sample (10%) for the BIAWGN channel. For each code size, the rate gains of SW-ECCT are averaged over all correlation levels and reported in Table 1 for the BSC and BIAWGN channel. For the BSC channel, the rate is only 17% above the theoretical SW bound. SW-ECCT improves over BP for both channels and all block sizes, but the performance is the best for the smaller codes. These results show that our proposed learned decoder can outperform classical BP decoding in a variety of settings. Consequently, we will experiment with our decoder incorporated in more elaborate frameworks with more complex correlation statistics. In the remainder of the experiments, we focus on the 64-bit code.

C. RESULTS ON WYNER-ZIV CODING

The first WZ scheme we evaluate is the learned monolithic WZ scheme of [30]. In contrast to [30], which assumes an ideal SW coder, we use a real SW codec employing either SW-ECCT or BP for decoding.

The continuous source X and side information Y follow quadratic Gaussian statistics,

$$X = Y + N, \text{ where } Y \sim \mathcal{N}(0, 1) \text{ and } N \sim \mathcal{N}(0, \sigma_n^2), \quad (15)$$

with a correlation noise variance σ_n^2 of either 0.1 or 0.01. Samples are grouped into blocks of 64, matching the LDPCA block size. The frozen quantizer and prior from [30] generate binary indices u_i and prior probabilities $p(u_i|y_i)$, from which syndrome bits and LLRs are computed and supplied to the SW-ECCT or BP decoder. We trained one SW-ECCT model on all code rates for $\sigma_n^2 = 0.1$ and another one for $\sigma_n^2 = 0.01$.

We evaluate the SW coding with rate-adaptivity, by using a feedback channel to the encoder to request more syndrome bits when decoding fails [54]. This ensures lossless transmission of the quantization indices, allowing a direct comparison of the theoretical rate and the rates achieved by SW-ECCT and BP, since the distortion will remain the same. This can be seen near the bottom of the graphs in Fig. 6, which shows the estimated rate-distortion point for an ideal SW coder [30], as well as the rates achieved when using SW-ECCT or BP decoding. For $\sigma_n^2 = 0.1$, the SW-ECCT decoder gains 0.023 bits/sample or 5.7% over BP, while for $\sigma_n^2 = 0.01$ the gain is 0.013 bits/sample or 2.3%. These rates are still 0.054 and 0.077 bits/sample away from the Slepian-Wolf bound estimated by the ANN prior model.

Additionally, we test the SW decoders without rate-adaptivity on each rate setting of the LDPCA code. For each fixed code rate, the reconstructed indices u_i are passed to the frozen dequantizer of [30] to evaluate the distortion. We provide the average distortion for each code rate in Fig. 6,

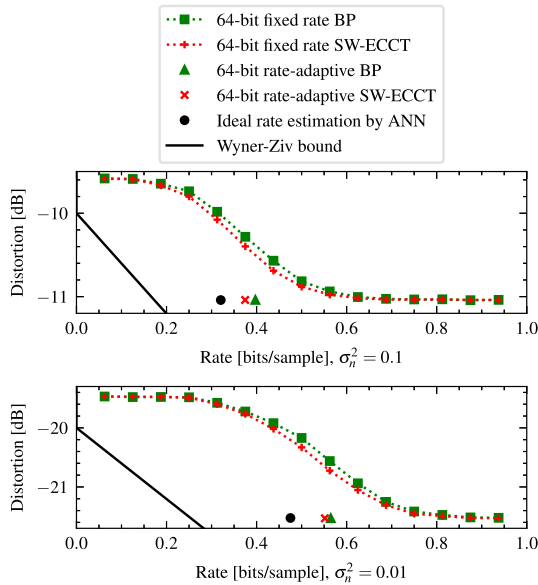


FIGURE 6. Rate-distortion points for the WZ monolithic coding case, comparing BP with SW-ECCT. The single points show the average rate when using rate-adaptive coding, while the curves show the average distortion when coding at a fixed rate.

averaged over 10^5 samples (x, y) . Since the minimum required rate for successful decoding is different for each codeword, some codewords will be encoded with too many bits, while others will fail to decode when using a single code rate. This combination of excess bits and decoding failures explains the gap between fixed rate and rate-adaptive coding, considering that rate-adaptive coding can select the best rate for each codeword to ensure successful decoding.

In the non-rate-adaptive case, our SW-ECCT can achieve up to 0.16 dB lower distortion levels in the middle of the rate range, and the gain averaged over all code rates is 0.039 dB. At very low rates, the amount of information available is so low that even a very good decoder will have a high error rate, masking any differences in decoder performance. On the other hand, at a high enough rate, the BERs are so low that further gains do not noticeably affect the average distortion anymore. Nevertheless, when looking at the BERs and block error rates (BLERs) in Fig. 7, we see that SW-ECCT still improves over BP at high rates.

D. COMPARISON WITH LEARNED BP IN WZ CODING

In order to compare with another neural channel decoder, we also tested learned BP (LBP) [41] inside the monolithic WZ framework of the previous subsection. We trained LBP models consisting of 183 learnable weights with a batch size of 100 and a learning rate of 10^{-3} for 50 epochs. We used the same optimizer, multi-loss function, learning rate decay, and gradient clipping as [41]. Unlike SW-ECCT, which is trained across all code rates from $\frac{4}{64}$ to $\frac{64}{64}$ with negligible performance loss, multi-rate training of LBP led to significant performance degradation. Therefore, a separate LBP model was trained for each code rate and evaluated on the same fixed

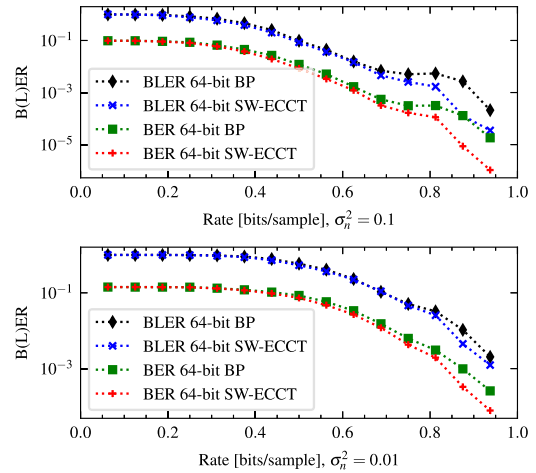


FIGURE 7. Bit error rates (BER) and block error rates (BLER) for each 64-bit LDPCA code rate for the WZ monolithic coding case, comparing BP with our SW-ECCT.

rate, following the same approach as Section V-C. Comparing the performance of BP, SW-ECCT and LBP on the fixed-rate coding, we observed that while our SW-ECCT model lowers the average distortion by 0.039 dB compared to BP, the LBP models actually *increased* the distortion by 0.010 dB.

This experiment with LBP shows two key advantages of our multi-rate SW decoder. Firstly, the large, model-free network enables improvements over BP decoding of LDPC codes, while other methods only show significant gains on code families not specifically designed for BP decoding [41], [42], [43]. Secondly, a single SW-ECCT model can be trained on the full rate range on an NVIDIA GeForce RTX 3090 GPU in 5.5 hours, while LBP requires 15 separate models trained in 21 minutes each, resulting in a similar training cost for an inferior decoding performance. Due to the poor performance of LBP, in subsequent experiments we will only retain BP as a baseline method for comparisons.

E. RESULTS ON LAYERED WZ CODING

Following the experiments on monolithic WZ coding, we continue with layered WZ coding, using the RNN-based framework from [31] with a conditional prior. The RNNs are trained on the same WZ statistics (15) as in Section V-C and our SW-ECCT decoder is integrated into the WZ RNNs as discussed in Section IV-A. This experiment is more challenging than the monolithic WZ scenario, as our proposed decoder leverages inter-bitplane correlation. It also links to Section V-F, which discusses the layered WZ coding of stereo images.

In Fig. 8, the rate-distortion curves are shown for the theoretical rate estimation from [31], rate-adaptive and fixed rate coding using both SW-ECCT and BP as decoders, as well as entropy coding. For the fixed rate coding, we chose for each bitplane the rate that achieves the best rate-distortion point. In this setting we may encounter SW decoding errors, resulting in a higher distortion at each bitplane compared to

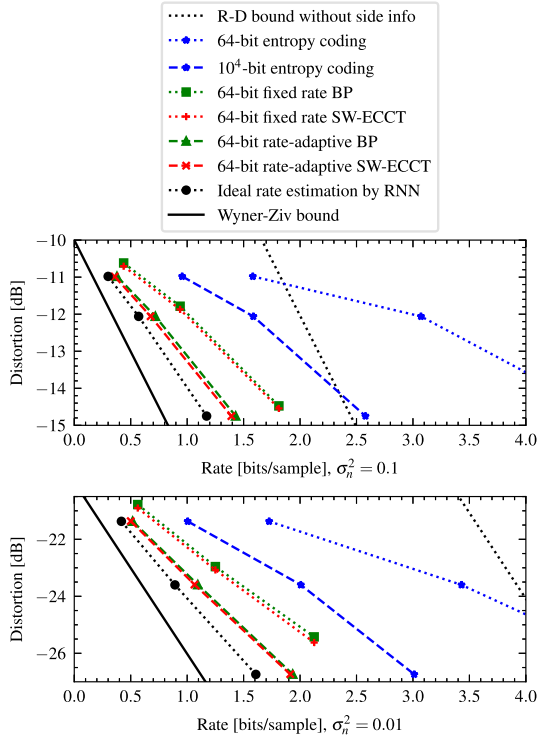


FIGURE 8. Rate-distortion curves for layered WZ coding, using our 64-bit LDPCA code. We compare rate-adaptive and fixed rate SW coding using BP or the SW-ECCT for decoding, as well as entropy coding.

the other approaches. In order to show that there is still considerable correlation between the output of the WZ quantizer and the side information, we also check the rate achieved by entropy coding of the indices, which does not make use of the side information. To create a probability model that is not conditioned on the side information, 10^6 values are sampled from the source X and passed through the quantizer RNN. For each possible set of earlier stage outputs, we count the frequencies of every quantization index and normalize. Consequently, estimates of the probability distributions $p(u_1)$, $p(u_2|u_1)$, $\dots$, $p(u_K|u^{K-1})$ are obtained. The quantization indices are then compressed at the encoder using rANS coding [47]. However, the implementation of the rANS entropy coder used is limited to outputting 32-bit words. If blocks of 64 bits are coded using the rANS coder, this has a big impact on the resulting rate. Therefore, we also show the entropy coding results for a block size of 10^4 , in order to reduce the effect of the excess bits.

Overall, rate-adaptive SW coding comes closest to the ideal rate as estimated by the learned prior model, with a rate overhead of around 19%. When comparing SW-ECCT and BP, SW-ECCT achieves rate gains of 0.021, 0.042, and 0.038 bits/sample or 5.4, 5.8, and 2.7% respectively for each bitplane at $\sigma_n^2 = 0.1$, and 0.017, 0.030, and 0.022 bits/sample for $\sigma_n^2 = 0.01$, or 3.3, 2.7, and 1.1%, respectively. When using fixed code rates for each bitplane, both the distortion and rate are higher than for rate-adaptive coding. Nevertheless, SW-ECCT gives a distortion 0.068-0.088 dB below BP for

$\sigma_n^2 = 0.1$, and 0.12-0.20 dB below BP for $\sigma_n^2 = 0.01$. It is also clear from the graphs that entropy coding gives significantly worse rates than SW coding, even when accounting for excess bits due to implementation details. This shows there is still correlation between the quantized source samples and the side information, necessitating the use of SW coding to achieve the maximum possible compression.

F. RESULTS ON WZ CODING OF STEREO IMAGES

As described in Section IV-B, we finally evaluate the proposed SW-ECCT in a learned VQ-VAE system for distributed stereo image coding. We compare four coding strategies for the quantized vector indices: uncoded, entropy coding with the autoregressive prior of [29], and SW coding with a learned conditional prior using BP decoding or our SW-ECCT decoder. This is the most challenging scenario, as we depart from known synthetic data distributions to complex, unknown correlation statistics.

Table 2 reports the rate-distortion performance for a scale factor of 8 and codebook sizes 2, 4, 8, and 256. Image quality is measured in peak signal-to-noise ratio (PSNR) and multi-scale structural similarity index measure (MS-SSIM), and rates are provided for uncoded quantization indices, entropy coding with rANS [47], SW coding with BP and SW coding with SW-ECCT. We consider two versions of SW-ECCT: one trained on codebook indices and prior probabilities derived from the KITTI Stereo data (12,608 training examples), which we refer to as SW-ECCT KITTI in Table 2, and one trained on random source bits with a BIAWGN side information channel as in Section V-B (10^8 training samples), which we refer to as SW-ECCT Synthetic. This allows us to assess the robustness of SW-ECCT under a correlation model mismatch between training and inference. The results in Table 2 show that SW-ECCT KITTI performs worse than SW-ECCT Synthetic, indicating that the limited size of the KITTI training set is insufficient to properly learn a general decoding procedure. On the other hand, SW-ECCT Synthetic performs well despite the data mismatch between train and test time, which shows we obtained a robust decoder that does not heavily depend on a specific correlation model.

For small codebook sizes, SW coding with a BP decoder already outperforms entropy coding, and SW-ECCT Synthetic further improves the rate, yielding a rate reduction of 18 and 8% over rANS for the two smallest codebooks. At larger codebook sizes however, entropy coding outperforms SW coding. One reason is that for higher bitplanes, the correlation between the source and side information decreases, leading to SW coding becoming less efficient than entropy coding. This inefficiency also accumulates as the number of bitplanes increases. One mitigation would be to implement different coding modes and dynamically switch between them during coding, as proposed in [55]. Another approach would be to use e.g., disparity compensation algorithms like [56] to increase the quality of the side information, since the conditional prior we trained is very simple, and likely does not fully exploit

TABLE 2. Compression Rates for KITTI Stereo Images Using a VQ-VAE, and Entropy Coding or SW Coding Trained on KITTI or Synthetic Data

Codebook size	PSNR [dB]	MS-SSIM	Uncoded [bpp]	rANS coding [bpp]	BP [bpp]	SW-ECCT KITTI [bpp]	SW-ECCT Synthetic [bpp]	SW bound [bpp]
2 (1 bit)	20.593	0.866	0.0156	0.0125	<u>0.0106</u>	0.0109	0.0102	0.0087
4 (2 bits)	21.873	0.902	0.0313	0.0250	<u>0.0234</u>	0.0242	0.0230	0.0194
8 (3 bits)	22.061	0.900	0.0469	0.0387	0.0398	0.0409	<u>0.0395</u>	0.0326
256 (8 bits)	24.240	0.941	0.1250	0.0999	<u>0.1220</u>	0.1246	0.1221	0.0912

TABLE 3. SW Decoding Throughput and Memory Usage for Different Numbers of Iterations/Layers N Using a 64-Bit LDPC Code

Decoder	CPU [codes/ms]	GPU [codes/ms]	GPU memory [MB]
BP $N=15$	1.12	6.7	7
BP $N=100$	0.17	1.0	11
SW-ECCT $N=15$	0.03	15.1	98
LBP $N=15$	0.80	6.1	16

the spatial correlations and inter-bitplane relations. This is reflected in the SW bound estimated by the learned conditional prior, shown in the last column of Table 2. For codebook size 2, the estimated SW bound is 30% lower than the entropy coding rate, while for codebook size 256, the gap is less than 9%. This indicates that the prior is less able to model the image correlation at high codebook sizes. With a more elaborate design of the prior, modeling correlations between bitplanes, considering the properties of natural (stereo) images and training on large image datasets, it may be possible to further improve the prior model and SW coding, but this is outside the scope of this paper.

G. COMPLEXITY EVALUATION

We also compare the decoding speed of SW-ECCT, BP, and LBP on the CPU and GPU under the task of SW coding for BSC correlation using a fixed rate of $\frac{32}{64}$ bits/symbol. All decoders are implemented in PyTorch. The same settings are used for the decoders as in Section V-B, but we also provide the throughput of BP limited to 15 iterations, matching the 15 layers of SW-ECCT. The 15-layer SW-ECCT decoder consists of 767×10^3 parameters, while LBP has 183 learnable parameters, and BP has no trainable weights. After a warmup run, all decoders are run on 10^4 codewords (each with a length of 64 bits), using a batch size of 100. We report the decoding throughput of each method in codes per millisecond in Table 3, tested on an AMD Ryzen 9 5950X CPU and an NVIDIA GeForce RTX 3090 GPU.

Since Transformers are not designed to run on a CPU, the SW-ECCT runs 5 times slower on the CPU than 100 iterations of BP. On the GPU on the other hand, SW-ECCT is 15 times faster than BP, and simultaneously provides lower error rates than BP as shown in Section V-B. For reference, we also

TABLE 4. Average Rate Increase [bits/sample] for Model Variations on the Chosen SW-ECCT Architecture

Model variant	BSC	BIAWGN
Embed. dim. 32	0.006	0.011
Embed. dim. 96	0.000	0.000
No syndrome emb. averaging	0.000	0.012
Separate syndrome weights	0.041	0.005

compare with 15 iterations of BP and LBP. However, even compared to these settings, our SW-ECCT is still more than twice as fast while achieving better decoding results. It is worth noting that, while the BP and LBP implementations make use of the sparsity of the parity-check matrix to reduce the computational time, our SW-ECCT is not optimized in this regard. It has been shown that by incorporating this sparsity in the self-attention layer and with quantization of the model parameters, significant reductions in computational cost can be achieved for (SW-)ECCT [57]. Alternatively, the cross-attention architecture from [58] can be considered, which also provides a more efficient channel decoder. Overall, the decoding performance and throughput of SW-ECCT, even without an optimized architecture, make it an attractive decoder for neural DSC frameworks, where the presence of a GPU is already required for the rest of the system.

H. SW-ECCT VARIANTS AND HYPERPARAMETER SELECTION

1) SW-ECCT EMBEDDINGS

To justify our architectural decisions on the SW-ECCT model, we assess the SW coding performance of different variants of the model on random binary data with BSC and BIAWGN correlation statistics as in Section V-B. The variants are trained in exactly the same way on 9 correlation levels and all code rates, but have different token dimensions or use another method to handle the different code rates. The results are listed in Table 4. They show the increase in rate, averaged over all correlation levels, between the tested model variant and the SW-ECCT architecture used in the rest of the paper.

Firstly, we look at the effect of the token embedding dimension on the model performance. By reducing the token dimension from 64 to 32, we see a significant drop in performance, almost halving the gain of SW-ECCT over traditional BP. On the other hand, increasing the token dimension to 96

yields no noticeable difference. Increasing the dimension even further to 128 results in divergence during training. Therefore, the chosen 64-dimensional embeddings provide a suitable tradeoff between model capacity and complexity.

Additionally, we evaluate alternative options to handle the multiple code rates in the SW-ECCT model instead of syndrome embedding averaging. For the *no syndrome embedding averaging* row in Table 4, we only select the syndrome embeddings corresponding to the current code rate, and do not average the embeddings of merged syndrome bits. For the BSC correlation channel this does not make a difference, but for BIAWGN statistics there is a significant performance penalty. Another approach is to use a separate set of syndrome embeddings and final projection layer for each code rate. The performance of this model variant is shown in the last row of Table 4, and also results in an increase in average rate. In conclusion, our method of averaging syndrome embeddings is a valid choice for the SW-ECCT. This is further confirmed in the next section, which shows that our multi-rate model achieves almost the same performance as training a separate model for each code rate or correlation level.

2) SINGLE-RATE AND SINGLE-CORRELATION MODELS

While the multi-rate SW-ECCT model outperforms traditional BP decoding, it is still possible that we lose performance by training on different correlation levels and code rates simultaneously. Therefore, we revisit the experiment on BSC statistics from Section V-B and train a separate (multi-rate) SW-ECCT model for each of the 9 correlation levels. Moreover, we take the monolithic WZ experiment from Section V-C for $\sigma_n^2 = 0.1$, and train a SW Transformer decoder for each code rate separately.

Using a separate model for each BSC correlation level lowers the achieved rate by 0.003 bits/sample, averaged over all correlation levels. For the monolithic WZ coding, using separate decoders for each code rate reduces the average distortion by 0.004 dB. Especially for the monolithic WZ coding, the gain in performance is negligible compared to the increase in model parameters and training compute by a factor of 15. When using a single SW-ECCT model for multiple BSC correlation levels, the increase in rate compared to separate models for each level starts to become noticeable. However, this difference of 0.003 bits/sample is still less than 1% of the overall data rate, and reduces training cost and model storage by an order of magnitude. Consequently, our proposed multi-rate SW-ECCT, trained on the whole useful correlation range, has almost the same performance as a group of specialized models, while reducing the storage and training costs by two orders of magnitude. This is a significant improvement over existing neural decoders, which are limited to single code rates and narrow communication noise ranges.

3) COMPARING SW-ECCT MODEL SIZES AND BP CONVERGENCE

In this section, we justify our choice for a 15-layer SW-ECCT model and a BP iteration limit of 100. The performance of

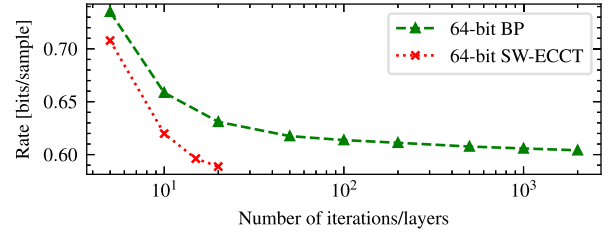


FIGURE 9. Performance of BP and SW-ECCT for different numbers of iterations/layers on rate-adaptive SW coding with BSC correlation statistics.

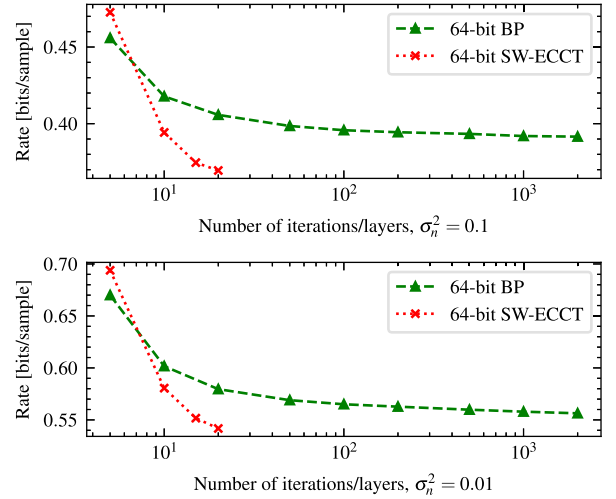


FIGURE 10. Performance of BP and SW-ECCT for different numbers of iterations/layers on rate-adaptive monolithic WZ coding.

our SW-ECCT as a function of the number of Transformer layers is analyzed and compared with the convergence of BP. Other model parameters, like the token embedding size and the number of attention heads, are kept fixed. For rate-adaptive SW coding with BSC statistics like in Section V-B, we use a single error probability of 0.11, which results in a conditional entropy $H(X|Y)$ of 0.5 bits/sample. The achieved rates for different numbers of BP iterations and different SW-ECCT model sizes are shown in Fig. 9. We also test the SW decoders in the WZ setting of Section V-C. With rate-adaptivity, the distortion is always the same, and we can also just compare the achieved rates. The results for the WZ setting are shown in Fig. 10.

For the BSC correlation channel, SW-ECCT always outperforms BP when the number of layers is equal to the number of BP iterations. For the WZ coding, there is a crossover point between 5 and 10 layers, after which SW-ECCT also beats BP at an equal number of iterations. On top of that, a 15-layer SW-ECCT model achieves lower rates than BP *at convergence*. Adding another 5 layers to the model reduces the rate further, but the gain in compression rate becomes smaller. Hence, we choose to use 15 layers for the SW-ECCT in all experiments, in order to limit the computational cost. As for the choice of the BP iteration limit, we observe that

TABLE 5. Stereo Image Coding Performance With SW Decoding Errors and a Finetuned VQ-VAE Decoder

	PSNR	MS-SSIM	Rate [bpp]	PSNR	MS-SSIM	Rate [bpp]	PSNR	MS-SSIM	Rate [bpp]
Codebook size	No decoding errors (rANS)			BP decoding errors			SW-ECCT decoding errors		
2 (1 bit)	20.593	0.866	0.0125	20.583	0.869	0.0106	20.558	0.872	0.0101
4 (2 bits)	21.873	0.902	0.0250	21.840	0.901	0.0233	21.707	0.899	0.0225
8 (3 bits)	22.061	0.900	0.0387	22.184	0.903	0.0396	22.038	0.903	0.0390
256 (8 bits)	24.240	0.941	0.0999	24.015	0.939	0.1235	24.203	0.940	0.1234

the difference between the rates at 100 iterations and 2000 iterations is just below 0.01 bits/sample for both the BSC channel and the WZ settings. The iteration limit of 100 thus provides a good tradeoff between decoding performance and computational costs.

I. EFFECT OF SW DECODING ERRORS ON STEREO IMAGE CODING

Up until now, we have assumed perfect decoding of the rate-adaptive SW decoders. However, in real applications the decoder cannot compare with the ground truth data to have a perfect classification of decoder success or failure. To create a framework that is usable in the real world, we replace the oracle implementation with practical decoder checks based on the syndrome of the reconstructed codeword and the decoder confidence. First, we check that the syndrome of the reconstructed codeword is equal to the syndrome we received from the encoder. This allows us to already discard more than 90% of the incorrect codewords. Subsequently, we define a confidence threshold on the magnitude of the output LLRs of the BP decoder or the logits of the SW-ECCT decoder. Unfortunately, the distributions of confidence values for correct and incorrect codewords overlap, so there is a tradeoff between block error rate and data rate.

In this section, we test the effect of SW decoding errors on the reconstruction performance of the VQ-VAE model for stereo image coding from Section V-F. We choose the decoder confidence thresholds to obtain a good rate-distortion trade-off, while keeping the rates similar to the SW-ECCT values in Table 2. However, SW decoding errors have a significant impact on the reconstruction performance, which drops by up to 1.1 dB in PSNR. In order to make the VQ-VAE decoder more robust to incorrect codebook vectors, we finetuned the decoder on SW-decoded indices containing decoding errors. We finetuned for 100 epochs using the AdamW optimizer, a learning rate of 10^{-3} and a weight decay of 0.1. The reconstruction performance after finetuning is shown in Table 5, together with the numbers for the entropy coded VQ-VAE which does not suffer from decoding errors.

Finetuning the VQ-VAE decoder helps to recover from SW decoding errors, especially for the small codebooks, resulting in an image quality that is on par with the performance of the entropy coded VQ-VAE without decoding errors. Additionally, the SW decoders with realistic (yet imperfect) error detection even achieve slightly lower rates than the

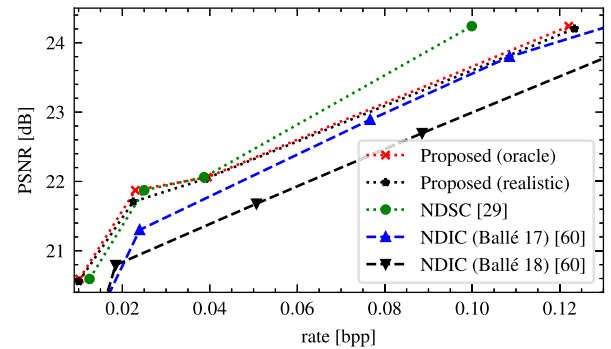


FIGURE 11. Comparison between NDIC [59], NDSC [29], and the proposed SW-ECCT coded VQ-VAE with perfect error detection (oracle) and practical error detection (realistic).

SW decoders with perfect decoder success classification from Table 2. Overall, our proposed practical decoding error detection performs similarly to the oracle detection results, demonstrating the practical viability of the SW-ECCT-based stereo image coding framework. We also provide a visualization in Fig. 11, which shows the rate-distortion curves for the VQ-VAE model with rANS coding from [29] (NDSC) and the VQ-VAE model with our proposed SW-ECCT codec, using either perfect error detection (oracle) or practical error detection checks (realistic). Additionally, we compare with another state-of-the-art method NDIC [59], using [22] and [40] as backbones. It is clear in Fig. 11 that our proposed approach outperforms both versions of NDIC at all rate-distortion points. Moreover, we also outperform NDSC at low rates, while for the highest rate we need a better prior or disparity compensation to match the performance of NDSC, as discussed in Section V-F. In conclusion, our proposed practical SW coding can significantly reduce the data rate compared to the state-of-the-art in distributed stereo image coding, with minimal losses in image quality.

VI. LIMITATIONS

As also mentioned in Section III-C, SW-ECCT is limited to small block sizes due to the computational complexity of the self-attention layer. Furthermore, larger block sizes are more difficult to learn due to the exponentially increasing code space, which results in reduced improvements of the learned decoder. Consequently, our model is mostly suited for

small to medium block sizes. However, our architecture is not really optimized, so there are multiple options [57], [58] to create more efficient models. A related issue is the relatively poor decoding throughput of our network on a CPU. This can likely also be improved by more optimized architectures. Nevertheless, SW-ECCT is an attractive decoder for neural DSC frameworks, where the presence of a GPU is already required for the rest of the system.

In the application of stereo image coding, our SW-ECCT model performs worse than entropy coding at high rates. This is, however, mostly a limitation of the simple conditional prior model. This issue may be resolved with more complex correlation models and/or disparity compensation algorithms.

VII. CONCLUSION

We present the first neural architecture for learned Slepian-Wolf decoding of LDPCA codes. It can be trained simultaneously on all possible code rates and side information correlation levels, reducing storage and training costs by at least an order of magnitude compared to existing neural decoders, which are limited to single code rates and narrow communication noise ranges. Compared to classical belief propagation decoding, our decoder reduces the SW rate of synthetic sources by up to 11%, while simultaneously achieving a 15 times higher throughput than BP on a GPU. Furthermore, our codec is integrated into learned Wyner-Ziv frameworks, providing the first constructive learned WZ systems, as earlier works assumed a non-constructive, ideal SW codec. Additionally, we incorporate our SW decoder into the more realistic and challenging scenario of distributed image coding. At low data rates, our method reduces the rate by up to 19% compared to the state-of-the-art, with minimal loss in image quality. This work shows the viability of SW coding in practical neural distributed coding architectures, translating information theory into real-world systems. Nevertheless, several research directions remain open for exploration. The architecture can be made more efficient, and its robustness on different source statistics and correlation model mismatches could be further investigated. Especially for the stereo image coding scenario, more advanced correlation models and disparity compensation algorithms may be explored, as well as adaptive mode decision to switch dynamically between SW coding and entropy coding for each bitplane.

ACKNOWLEDGMENT

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

- [1] X. Zhang, J. Shao, and J. Zhang, "LDMIC: Learning-based distributed multi-view image coding," in *Proc. Int. Conf. Learn. Representations*, 2023, p. 16.

- [2] B. Liu, X. Sheng, L. Li, D. Liu, and H. Li, "Distributed deep point cloud feature compression for vehicle-to-vehicle cooperative perception," *IEEE Trans. Multimedia*, early access, Jan. 19, 2026, doi: 10.1109/TMM.2026.3655479.
- [3] L. Abrahamyan, Y. Chen, G. Bekoulis, and N. Deligiannis, "Learned gradient compression for distributed deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7330–7344, Dec. 2022.
- [4] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
- [5] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
- [6] S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.
- [7] N. Deligiannis, A. Sechelea, A. Munteanu, and S. Cheng, "The no-rate-loss property of Wyner-Ziv coding in the Z-channel correlation case," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1675–1678, Oct. 2014.
- [8] Y. Steinberg and N. Merhav, "On successive refinement for the Wyner-Ziv problem," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1636–1654, Aug. 2004.
- [9] A. Liveris, Z. Xiong, and C. Georgiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 440–442, Oct. 2002.
- [10] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. Data Compression Conf.*, 2002, pp. 252–261.
- [11] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. Data Compression Conf.*, 1999, pp. 158–167.
- [12] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1250–1276, Jun. 2002.
- [13] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.
- [14] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, "Wyner-Ziv coding based on TCQ and LDPC codes," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 2003, vol. 1, pp. 825–829.
- [15] S. Cheng and Z. Xiong, "Successive refinement for the Wyner-Ziv problem and layered code design," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3269–3281, Aug. 2005.
- [16] D. Rowitch and L. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 948–959, Jun. 2000.
- [17] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Process.*, vol. 86, no. 11, pp. 3123–3130, 2006.
- [18] F. Verbist, N. Deligiannis, S. M. Satti, P. Schelkens, and A. Munteanu, "Encoder-driven rate control and mode decision for distributed video coding," *EURASIP J. Adv. Signal Process.*, vol. 2013, no. 156, p. 15, 2013.
- [19] J. Slowack, J. Skorupa, N. Deligiannis, P. Lambert, A. Munteanu, and R. Van de Walle, "Distributed video coding with feedback channel constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 7, pp. 1014–1026, Jul. 2012.
- [20] S. Wang, L. Cui, S. Cheng, L. Stankovic, and V. Stankovic, "Onboard low-complexity compression of solar stereo images," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3114–3118, Jun. 2012.
- [21] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 67–76, Sep. 2007.
- [22] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Representations*, 2017, p. 27.
- [23] H. Wu, Y. Shao, C. Bian, K. Mikolajczyk, and D. Gündüz, "Deep joint source-channel coding for adaptive image transmission over MIMO channels," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 15002–15017, Oct. 2024.
- [24] K. Yang, S. Wang, J. Dai, X. Qin, K. Niu, and P. Zhang, "SwinJSCC: Taming swin transformer for deep joint source-channel coding," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 1, pp. 90–104, Feb. 2025.

- [25] Y. Shi, S. Shao, Y. Wu, W. Zhang, and M. Debbah, "RWZC: A model-driven approach for learning-based robust Wyner-Ziv coding," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 7, pp. 2364–2378, Jul. 2025.
- [26] J.-J. Xiao and Z.-Q. Luo, "Multiterminal source-channel communication over an orthogonal multiple-access channel," *IEEE Trans. Inf. Theory*, vol. 53, no. 9, pp. 3255–3264, Sep. 2007.
- [27] V. M. Prabhakaran, R. Puri, and K. Ramchandran, "Hybrid digital-analog codes for source-channel broadcast of Gaussian sources over Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4573–4588, Jul. 2011.
- [28] N. Mital, E. Özyilkan, A. Garjani, and D. Gündüz, "Neural distributed image compression with cross-attention feature alignment," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 2498–2507.
- [29] J. Whang, A. Nagle, A. Acharya, H. Kim, and A. G. Dimakis, "Neural distributed source coding," *IEEE J. Sel. Areas Inf. Theory*, vol. 5, pp. 493–508, 2024.
- [30] E. Özyilkan, J. Ballé, and E. Erkip, "Learned Wyner-Ziv compressors recover binning," in *Proc. IEEE Int. Symp. Inf. Theory*, 2023, pp. 701–706.
- [31] B. Joukovsky, B. De Weerd, and N. Deligiannis, "Learned layered coding for successive refinement in the Wyner-Ziv problem," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2024, pp. 6020–6024.
- [32] Y. He, G. Yu, and Y. Cai, "Rate-adaptive coding mechanism for semantic communications with multi-modal data," *IEEE Trans. Commun.*, vol. 72, no. 3, pp. 1385–1400, Mar. 2024.
- [33] J. Chen, D.-K. He, and A. Jagmohan, "The equivalence between Slepian-Wolf coding and channel coding under density evolution," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2534–2540, Sep. 2009.
- [34] Y. Choukroun and L. Wolf, "Error correction code transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 38695–38705.
- [35] T. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 226–228, Mar. 1975.
- [36] S. Chen, M. Effros, and V. Kostina, "Lossless source coding in the point-to-point, multiple access, and random access scenarios," *IEEE Trans. Inf. Theory*, vol. 66, no. 11, pp. 6688–6722, Nov. 2020.
- [37] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inf. Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.
- [38] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong, "Slepian-Wolf coded nested lattice quantization for Wyner-Ziv coding: High-rate performance analysis and code design," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4358–4379, Oct. 2006.
- [39] J. Chou, S. Pradhan, and K. Ramchandran, "Turbo and trellis-based constructions for source coding with side information," in *Proc. Data Compression Conf.*, 2003, pp. 33–42.
- [40] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Representations*, 2018, p. 47.
- [41] M. Lian, F. Carpi, C. Häger, and H. D. Pfister, "Learned belief-propagation decoding with simple scaling and SNR adaptation," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 161–165.
- [42] A. Bennatan, Y. Choukroun, and P. Kisilev, "Deep learning for decoding of linear codes - A syndrome-based approach," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 1595–1599.
- [43] E. Nachmani and L. Wolf, "Hyper-graph-network decoders for block codes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, p. 11, 2019.
- [44] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [45] X. He, L. Zhou, and J. Du, "PEG-like design of binary QC-LDPC codes based on detecting and avoiding generating small cycles," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 1845–1858, May 2018.
- [46] C. Yu and G. Sharma, "Improved low-density parity check accumulate (LDPCA) codes," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3590–3599, Sep. 2013.
- [47] J. Duda, K. Tahboub, N. J. Gadgil, and E. J. Delp, "The use of asymmetric numeral systems as an accurate replacement for huffman coding," in *Proc. Picture Coding Symp.*, 2015, pp. 65–69.
- [48] N. Deligiannis, A. Munteanu, S. Wang, S. Cheng, and P. Schelkens, "Maximum likelihood Laplacian correlation channel estimation in layered Wyner-Ziv coding," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 892–904, Feb. 2014.
- [49] X. Artigas et al., "The discover codec: Architecture, techniques and evaluation," in *Proc. Picture Coding Symp.*, Lisboa, Portugal, 2007, pp. 1103–1120.
- [50] R. P. Westerlaken, S. Borchert, R. K. Gunnewiek, and R. L. Lagendijk, "Analyzing symbol and bit plane-based LDPC in distributed video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, 2007, pp. II-17–II-20.
- [51] S. Ayzik and S. Avidan, "Deep image compression using decoder side information," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 699–714.
- [52] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [53] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3061–3070.
- [54] C. Brites, J. Ascenso, and F. Pereira, "Feedback channel in pixel domain Wyner-Ziv video coding: Myths and realities," in *Proc. Eur. Signal Process. Conf.*, 2006, pp. 1–5.
- [55] J. Slowack et al., "Rate-distortion driven decoder-side bitplane mode decision for distributed video coding," *Signal Process.: Image Commun.*, vol. 25, pp. 660–673, 2010.
- [56] D. Chen, D. Varodayan, M. Flierl, and B. Girod, "Distributed stereo image coding with improved disparity and noise estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 1137–1140.
- [57] M. Levy, Y. Choukroun, and L. Wolf, "Accelerating error correction code transformers," 2024, *arXiv:2410.05911*.
- [58] S.-J. Park, H.-Y. Kwak, S.-H. Kim, Y. Kim, and J.-S. No, "CrossMPT: Cross-attention message-passing transformer for error correcting codes," in *Proc. Int. Conf. Learn. Representations*, 2024, p. 19.
- [59] N. Mital, E. Özyilkan, A. Garjani, and D. Gündüz, "Neural distributed image compression using common information," in *Proc. Data Compression Conf.*, 2022, pp. 182–191.



Strategic Basic Research from the Research Foundation Flanders (FWO), Belgium, in 2022.



Strategic Basic Research from the Research Foundation Flanders (FWO), Belgium, in 2022.

BRENT DE WEERDT (Graduate Student Member, IEEE) received the master's degree in electrical engineering from Vrije Universiteit Brussel, Ixelles, Belgium, and Université Libre de Bruxelles, Brussels, Belgium, in 2021. He is currently working toward the Ph.D. degree with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel. His research interests include distributed source coding, interpretable deep learning, image and video processing, and deep unfolding. He was the recipient of the Ph.D. Fellowship

NIKOS DELIGIANNIS (Senior Member, IEEE) received the Diploma in electrical and computer engineering from the University of Patras, Patras, Greece, in 2006, and the Ph.D. degree in engineering sciences from Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 2012. He is currently a Professor with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), the holder from 2024 to 2025 Francqui Research Professorship on Trustworthy AI with VUB, and a Principal Investigator at imec, Belgium. His current research focuses on interpretable and explainable AI, multidimensional signal processing, computer vision, and distributed/federated AI for automotive and healthcare applications. He was the recipient of the 2024 ERC Consolidator Grant to conduct research with the intersection between interpretable and explainable AI and multiterminal compression for autonomous vehicles. From 2013 to 2015, he was a Senior Researcher with the Department of Electronic and Electrical Engineering, University College London, London, U.K. He is a member of the EURASIP and was the Chair of the EURASIP Technical Area Committee on Signal and Data Analytics for Machine Learning from 2021 to 2023. He was an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and has been a Guest Editor for special issues with the *EURASIP Journal on Advances in Signal Processing* and the *Signal Processing journal*.