IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 May 2024, accepted 19 June 2024, date of publication 1 July 2024, date of current version 9 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3421346

RESEARCH ARTICLE

A Symmetric Graph Transformation Framework for Image Compression

ALESSANDRO GNUTTI, (Member, IEEE), FABRIZIO GUERRINI, (Member, IEEE), AND RICCARDO LEONARDI, (Fellow, IEEE)
Department of Information Engineering, CNIT, University of Brescia, 25123 Brescia, Italy
Corresponding author: Alessandro Gnutti (alessandro.gnutti@unibs.it)

ABSTRACT In this paper, a framework for image and video intra-frame coding able to effectively employ the multiple transform paradigm using Symmetry-Based Graph Fourier Transforms (SBGFTs) is proposed. As data representation relies heavily on the characteristics of signal classes in high dimensional spaces, over the years it has been understood that signal instances for a given class, e.g., images, typically lie on a manifold which is not a single linear subspace. Accordingly, standards for image/video compression have considered to introduce multiple representation models to encode the data, i.e., multiple linear block transforms. However, the advantages are currently limited typically due to implementation complexity and the high signaling cost of the representation mode. As a result, only a small set of alternative transforms is typically considered, which restricts the adaptation capabilities to the data. In this paper, we instead argue that it is feasible to incorporate into the image coding framework a large set of SBGFTs which can overcome the aforementioned drawbacks. Specifically, we demonstrate the ability to derive the block encoding transform index from the quantization pattern fingerprint using a Multilayer Perceptron (MLP) architecture, achieving prediction with high confidence, surpassing 80% for 4-top accuracy. This translates into a more efficient entropy coding strategy for the index, allowing to save on the average more than 3 bits per block. These experimental results highlight the significant benefits of SBGFTs for multiple transform coding, particularly when combined with the proposed MLP based index prediction, with only a limited increase in computational complexity.

INDEX TERMS Image coding, multiple transforms, symmetry-based graph Fourier transforms, H.266/VVC, mode prediction, quantization, neural network architectures, multilayer perceptron.

I. INTRODUCTION

Image and video compression standards have always relied on linear block transform coding as a key representation framework. The general underlying assumption is that, while the data samples in the original domain are strongly correlated, the corresponding coefficients in a suitable transform domain turn out statistically independent, or to the least uncorrelated, making it easier to encode them [1].

It is well known that the 2D Discrete Cosine Transform (DCT) approximates the optimal Karhunen-Loève Transform (KLT) under first-order Markov conditions [2], which is a rather appropriate statistical model for natural visual content [3]. For this reason, the DCT has been employed in many popular past and present block-based hybrid compression systems, with a few exceptions, e.g., the ISO JPEG2000 standard [4] which is based on the Discrete Wavelet Transform (DWT) [5]. In particular, the DCT has been used for video compression, from the ITU-T H.261 standard [6] to the most recent JVET H.266/VVC [7]. Similarly, in image compression, it has been adopted from the ISO JPEG standard [8] to BPG [9], which is based on the intra-frame coding of JVET H.265/HEVC [10].

However, it is arguable that a fixed transform does not accommodate well with the frequent statistical changes occurring in natural images [11]. Furthermore, in some of the more recent coding systems among those previously cited, the transform is not even applied on the original

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

visual data. Instead, it is most often applied on residual signals, the output of a prediction process, which possess different statistical features. So, a single transform is expected not to be as effective for producing uncorrelated coefficients in the context of representing a prediction residual signal.

Recently, to overcome these issues, several works have proposed compression schemes relying on an alternative coding paradigm based on *multiple transforms* [12], [13], [14]. In this context, the encoder and the decoder share a set of transforms. The encoder chooses the most appropriate transform for each block through a rate-distortion (RD) optimization, and sends the transform coefficients and the corresponding transform index to the decoder to identify which inverse transform to employ for reconstruction.

Ideally, a *large* family of transforms could be designed for adapting to the diversified statistical contents that characterize the visual data. For example, a representative image (or residual) block training set can be partitioned into a considerable number of clusters, then for each cluster a specific KLT, which is optimal for linear approximation, can be built. In the end, however, straight KLTs are not usually considered as feasible candidate transforms because they are non separable, thus computationally intensive, and the needed clustering process can be captious. Nonetheless, non-separable transforms have been proposed for multiple transform coding [15], and a non-separable transform is even used in VVC as an alternative mode.

Still, the rationale for using a large set of available optimized transforms, which is increasing the number of transforms to significantly reduce data redundancy, is in principle attractive. However, coding frameworks with *small* transform sets have been deployed so far, usually relying on computationally efficient variants of the type-2 DCT, i.e., DCT-II. The first visual compression standard in which a multiple transform idea has been adopted is HEVC, where the Discrete Sine Transform (DST) complements the DCT. Its successor VVC includes additional primary transforms besides the usual DCT-II, namely, the 4 combinations of the type-8 DCT (DCT-VIII) and the type-7 DST (DST-VII) applied in the horizontal and vertical directions.

In essence, the cardinality of the transform set plays a crucial role, and that is why smaller sets have been employed so far. While a large set of transforms would allow to choose the most efficient transform in a RD sense for each block, such gain is bound to be mitigated by an increasing cost to signal the transform index. In addition, the transform selection process becomes more demanding unless it is based on sub-optimal heuristics.

Symmetry-Based Graph Fourier Transforms (SBGFTs) have been originally proposed in [16]. They enjoy remarkable energy compaction properties, even superior to families of KLTs [17], and they also can be implemented very efficiently [18]. Their construction is derived from Graph Signal Processing (GSP) [19], [20], in which the graphs are adopted as models that represent the correlation among data samples by determining edge weights between nodes.

In this paper, we argue that in a framework for image and video intra-frame coding it is possible to adopt a large SBGFTs set that is able to avoid the aforementioned drawbacks. More specifically, a feasible approach to reduce the average number of overhead bits required to signal the transform index is derived, a mandatory feature for allowing a large set of efficient transforms.

The transform index coding approach involves estimating the likelihood of each index from the map of each Quantized Transform Coefficients Block (QTCB). To this end, a Multilayer Perceptron (MLP) estimates a confidence vector, where each element represents the probability that a given set of quantized coefficients is derived from each transform. This vector is then used to design a conditionally dependent entropy code. The codeword that is sent to the decoder is associated with the true transform index by using the code derived from the estimated index probabilities produced by the QTCB map, which is thus on average a more compact code.

To prove the flexibility of the MLP approach, extensive experiments have been carried out on both pixel domain and residual blocks datasets, the latter derived from the prediction modes integrated in VVC. The proposed technique allows to significantly reduce the average bit-length per block needed to represent the transform index.

To properly highlight the potential benefits of using the SBGFTs in the intra-frame coding phase of a video compression scheme, significant RD gains are demonstrated with respect to DCT variants in a VVC-like codec using the Explicit Multiple Transform Selection (MTS) method [7]. In the experiments, since the investigated SBGFTs are specifically crafted to operate on blocks with dimensions of 8×8, they have been only applied to 8×8 residual blocks that result from the VVC block partitioning and intra-prediction, leaving other sized blocks using the VVC primary transforms as usual.

The rest of the paper is organized as follows. Sec. II recalls background knowledge on SBGFTs and cast them in a multiple transform coding framework. In Sec. III, we discuss the proposed solution for transform index coding, thus providing the underlying motivation. The experimental setup and the obtained results for the standalone transform index coding technique are presented in Sec. IV. Then, Sec. V shows the experimental results in terms of RD performance obtained by applying the SBGFTs set and transform index coding instead of the VVC primary transforms on 8×8 residual blocks. Finally, Sec. VI concludes the paper.

The source code implementing the proposed framework is publicly available at [21].

II. SBGFT BASED MULTIPLE TRANSFORM FRAMEWORK

In this Section, we recall background information and introduce the experimental multiple transform coding framework

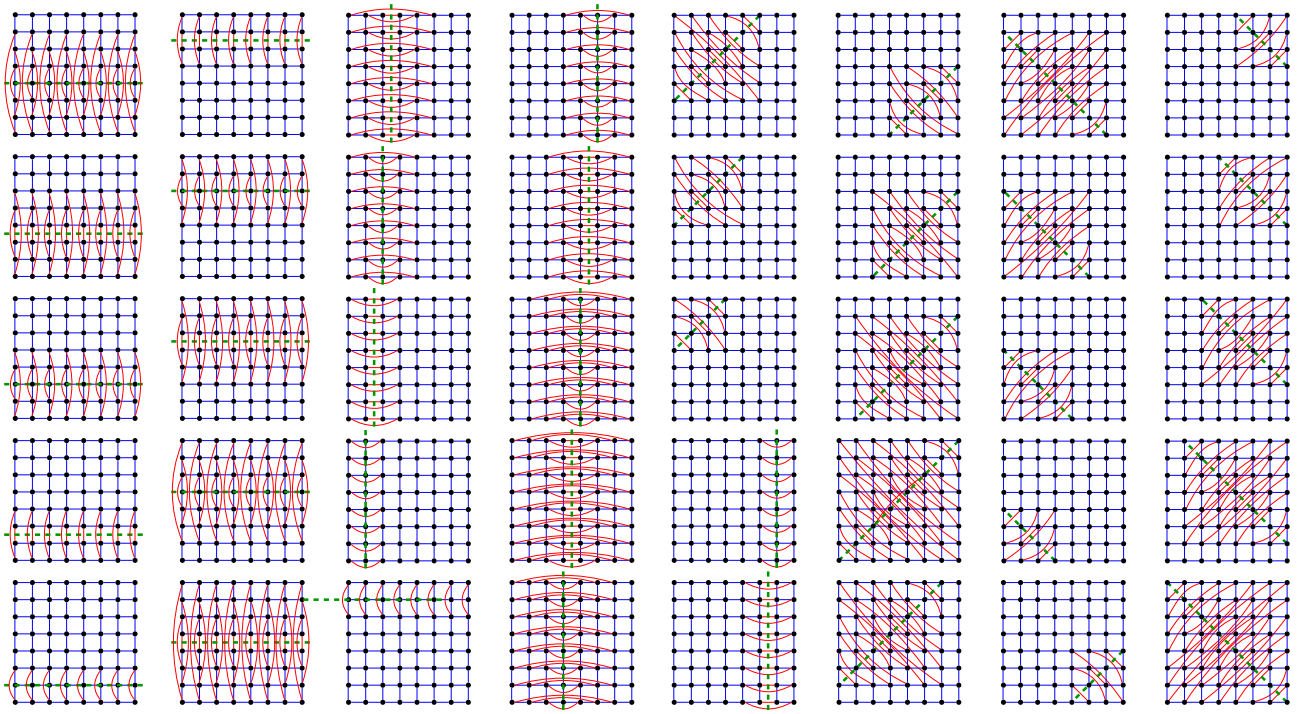


FIGURE 1. The set of 40 SBGFTs. Blue: edges forming the starting 2D grids. Dashed green: the symmetry axes. Red: additional edges representing symmetric connections.

using the SBGFTs. The transform set is briefly described in Sec. II-A. The framework for assessing the effectiveness of SBGFTs in an image coding scheme is presented in Sec. II-B.

A. SYMMETRY-BASED GRAPH FOURIER TRANSFORMS

In Graph Signal Processing (GSP) [20], graph signals are placed on the set of vertices of an undirected, connected, and weighted graph. A spectral representation of graph signals is defined by employing the Laplacian matrix eigenvalue decomposition. Since the Laplacian is self-adjoint, its eigenvectors form an orthonormal basis for the so-called Graph Fourier Transform (GFT) [19]. Therefore, the GFT coefficients are obtained by computing the inner product between the graph signal and each eigenvector.

The SBGFTs are a family of GFTs that have been proposed in [16], [17], and [22] for efficiently representing 8×8 blocks of both image and residual data. Each block is modeled as a graph, the nodes of which are its pixel/residual values. Accordingly, graphs connect pixels in specular position with respect to a predefined symmetry axis. Considering different specular positions, a set of 40 SBGFTs can be derived, each associated with a different symmetric graph determined by the location and orientation of the considered symmetry axis. Fig. 1 provides a visual representation of the graphs associated with the SBGFTs.

There are two main reasons why the design of these graphs pursue symmetries. First, symmetries are often present in both real-world data [23], [24] and residual signals [18]. For example, in [18], the authors worked with graphs modeled from the correlation between coefficients of residual blocks obtained through HEVC prediction. The considered blocks showed various symmetry types, therefore the authors concluded that there exist efficient 2D GFTs built from symmetric grids. As a matter of fact, the SBGFTs possess remarkable energy compaction properties, with promising results for non-linear approximation or coding tasks. A second important reason is that all the symmetric graphs are associated with a GFT that admits a fast transform implementation due to their natural symmetry [18]. Lower computational complexity of transforms is of course a necessity for practical image/video compression systems.

B. MULTIPLE TRANSFORMS FRAMEWORK

The framework for evaluating the efficiency of SBGFTs within a multiple transform compression scheme is proposed in Fig. 2. Specifically, the objective is to investigate the potential benefits of utilizing the previously described SBGFTs instead of the VVC primary transforms for 8×8 residual blocks.

To achieve this, a video frame F is processed to generate a family of residuals E obtained from VVC intra-coding. Of course, the resulting residual blocks can vary in size, depending on the output of the VVC partitioning process. In particular, F can be partitioned into blocks ranging from 64×64 to 4×4. More details on VVC partitioning and intra-coding can be found in [7].

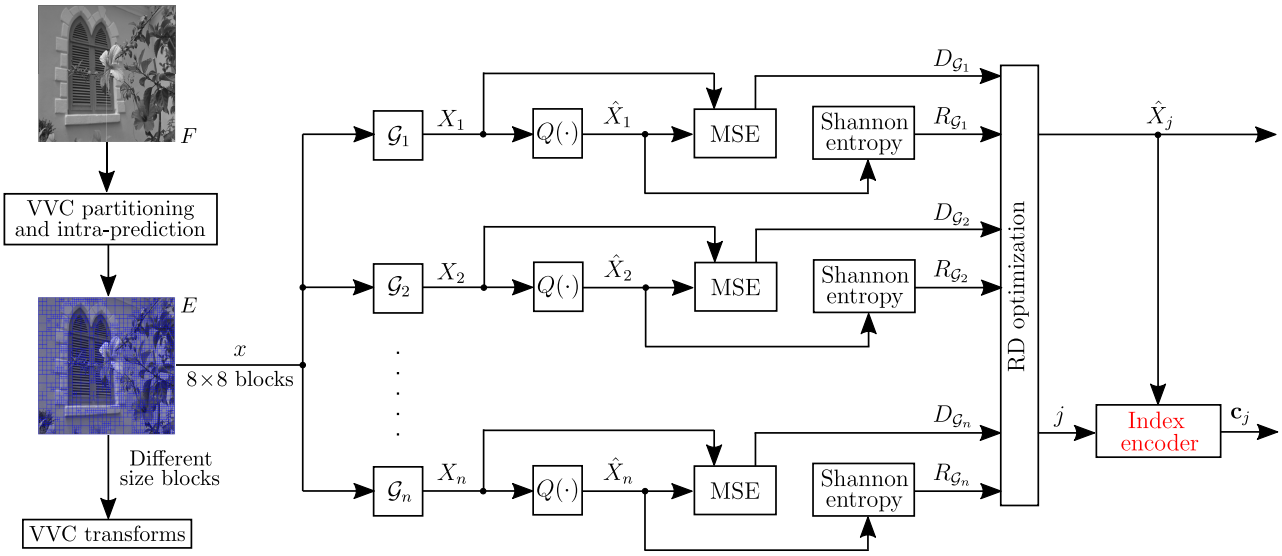


FIGURE 2. Framework for evaluating the performance of the 8×8 SBGFTs in a multiple transform compression scheme. The frame *F* undergoes partitioning through the VVC partitioning process, and residual blocks *E* are generated during the VVC intra-coding phase. All blocks, excluding those of size 8×8, undergo processing with the DCT variants transforms employed in VVC intra-coding. The 8×8 SBGFTs, instead, are specifically applied to 8×8 blocks. The optimal graph is chosen by minimizing the Lagrangian cost function given in Eq. (1). The index *j* of the optimal graph is represented using the code built from the output of the proposed MLP (refer to Fig. 3).

At this point, the SBGFTs set is applied to transform 8×8 residual blocks only, while blocks of a different size follow the standard VVC process that adopts the DCT variants. Experimentally, 8×8 blocks cover approximately 40% of the frame area on average, across a wide range of Quantization Parameters (*QP*s). An example of block partitioning generating the residuals E is depicted in Fig. 2.

Let us now consider a length-64 column vector x representing one of the unwrapped residual 8×8 blocks of E. Let $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n\}$ be the set of $n = 40$ SBGFTs, represented by 64×64 matrices where the columns form the correspondent orthonormal basis. To perform the transform selection driven by RD optimization for x, all the SBGFTs are applied to x by computing n column vectors in the transform domain, thus $X_i = \mathcal{G}_i^T \cdot x$, $i = 1, \ldots, n$. The X_i vectors are then quantized with a given *QP*, obtaining $\hat{X}_i$. Finally, the inverse GFTs are applied to $\hat{X}_i$, so reconstructing n signals $\tilde{x}_i = \mathcal{G}_i \cdot \hat{X}_i$.

Next, an unconstrained RD optimization problem is solved to choose the optimal graph transform for each block. Therefore, for each $\mathcal{G}_i$ a pair $(R_{\mathcal{G}_i}, D_{\mathcal{G}i})$ is computed first. The distortion $D_{\mathcal{G}_i}$ is obtained as the Mean Squared Error (MSE) between X_i and $\hat{X}_i$, which is the same as the MSE between x and $\tilde{x}_i$, due to Parseval's theorem. Then, $R_{\mathcal{G}_i}$ approximates the bitrate needed to represent the transform coefficients $\hat{X}_i$, using the quantized coefficients estimated Shannon entropy.

Finally, the optimal GFT $\mathcal{G}_j$ for x is identified through the minimization of the following Lagrangian cost function:

$$j = \arg\min_i D_{\mathcal{G}i} + \lambda R_{\mathcal{G}i} \tag{1}$$

with the Lagrangian multiplier $\lambda \geq 0$ determining the RD working point. After selecting the optimal GFT, the quantized coefficients are encoded and transmitted to the decoder, along with the corresponding index j.

Specifically, this index j is separately encoded into a codeword $\mathbf{c}_j$. This ensures that the correct inverse transformation for the QTCB $\hat{X}_j$ can be applied at the decoder. How to best design the codetable for the codewords $\mathbf{c}_i$ is described in the next Section.

III. TRANSFORM INDEX REPRESENTATION

In this Section, we discuss strategies for building an effective codeword $\mathbf{c}_j$ for the transform index j.

As a straightforward baseline solution, the optimal transform index j can be encoded using fixed-length codewords at most $\lceil \log_2 l \rceil$ bits long, no matter the chosen transform. Of course, a more effective code could be built, were the probability distribution of the transforms known, so that the encoder and decoder may share a codetable for encoding and decoding j. For example, such probabilities may be learned offline over a large set of data. However, the latter strategy requires an adequately representative dataset, which is not a practical solution.

The authors of [25] instead proposed a Convolutional Neural Network (CNN) to predict the chosen transform index from the QTCB, to gain in representation efficiency. The general idea is as follows: a neural network takes a QTCB as input and returns a confidence vector, giving the probabilities that the examined QTCB is derived from each candidate transform. Such probabilities are then used to design an efficient codetable, and the codeword associated with the true transform index is then sent to the decoder. Through the same

process, the decoder can build on its own the same codetable from the QTCB, and so it is able to decode the correct index from the received codeword.

We remark that Artificial Intelligence (AI) techniques in general are commonly found nowadays in many different computer vision tasks. Specifically, the seminal work [26] kick-started a renewed interest in image compression, particularly introducing AI-powered end-to-end compression. Notably, for video intra-frame compression, neural architectures have been either employed in the same end-to-end fashion, or employed for specific tasks still within a traditional hybrid compression framework [27]. The work in [25] falls into the latter category, where low complexity is paramount as the neural architecture represents only a fraction of the entire hybrid coding system.

Likewise, in this work a low complexity neural network architecture is employed for the specific job of transform index prediction inside a hybrid image coding system using multiple transforms. The proposed network takes a QTCB as input to estimate the probability of each transform being the true encoding transform from which the QTCB has been built among the sizable set of possible SBGFTs. It then uses these probabilities to build a code where each candidate transform is encoded using a variable-length codeword.

Nonetheless, many differences exist between the method proposed in this paper and [25]. Therein, a CNN was used to estimate the probabilities. However, learning cannot be achieved efficiently given the sparse structure of the input data. Moreover, a simple truncated unary code was employed to build the associated codewords. From a validation perspective, importantly, the method was only applied to 4×4 residual blocks generated from 4 HEVC intra-prediction modes, and the results showed a small gain with respect to a fixed-length code to represent just 4 candidate transforms. These embody several drawbacks to achieve good RD performance in more complex and realistic scenarios, as shown by the experiments reported in Secs. V and IV.

A. DISCUSSION

The problem of transform index prediction could be interpreted as a soft multi-class classification problem, where the classifier is employed to estimate the confidence level of each class (the candidate transform) being the correct one, and the QTCB represents a block to be classified. To solve this type of problems with images, CNN architectures are usually appropriate.

In a CNN, the neurons in a convolutional layer are only connected to nearby neurons from the preceding layer, and the number of connections depends on the size of the convolutional kernel. So, these neurons determine a narrower range of features. In other words, the activation of a single neuron is insensitive to the activation of the majority of the neurons from the previous layer. This is particularly efficient when local information is important, as in images. Employing a CNN is especially advantageous when the input data are characterized by local patterns with strong spatial correlation.

On the contrary, coefficients in the transform domain are typically uncorrelated, and the energy of the original signal is distributed into very few significant components. Quantization can only further increase the sparsity of the transform coefficients. As a consequence, adopting local convolutional filters in the transform domain may be ineffective, particularly when trying to identify which transformation has originated a given set of quantized coefficients.

As an example, consider the following two QTCBs:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

They are nearly identical, except for position (6, 7). However, these QTCBs have been actually generated by a pair of different transforms applied to two separate residual blocks. Thus, even a slight change in the position (or value) of a single transform coefficient may indicate, at least on average, that the entire block has been originated by distinct transforms.

Thus, we conclude that the implementation of small convolutional kernels is not suitable for this type of data, since there is no local structural information that could be learned. Conversely, to perform a better prediction, the absence of any special assumption in the structure of the data supports the adoption of fully-connected layers instead.

B. ALGORITHM

Let us recall that, in the multiple transform scenario illustrated in Fig. 2, a QTCB ($\hat{X}_j$) at the encoder is generated by transforming and quantizing a residual block. The adopted SBGFT, say $\mathcal{G}_j$, is selected through RD optimization among the set of n available transforms $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n$, with in our case $n = 40$.

Building upon the insights provided in the previous Section, an MLP is adopted. The process is described in Fig. 3. The MLP takes the QTCB as input. On both sides, the encoder and decoder modules use the same trained MLP. The MLP output is a probability vector $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, the elements of which indicate the estimated probabilities, i.e.,

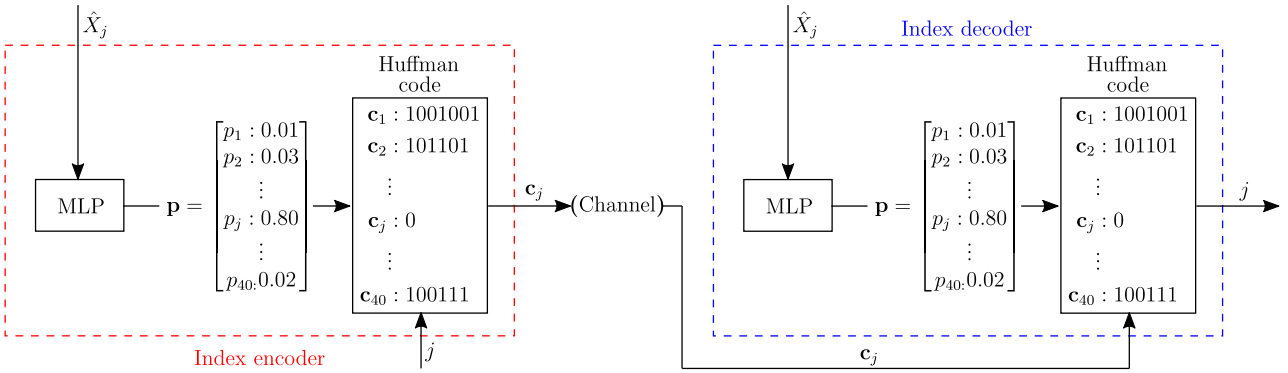


FIGURE 3. Proposed graph index representation with a variable-length code. This example showcases a favorable instance in which the MLP produces a remarkably high confidence value (probability) for the true GFT $\mathcal{G}_j$ that the encoder selected for this block through RD optimization. As a result, the corresponding codeword $\mathbf{c}_j$ is merely one bit in length, leading to an exceptionally efficient encoding. Note that the decoder is able to replicate the codetable by simply inputting the received $\hat{\boldsymbol{x}}_j$ into its own built-in MLP, which is the same as the encoder one.

confidence values, that the considered QTCB derives from transforms $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n$, respectively. Such a vector is used for constructing a Huffman code, which generates n uniquely decodable, variable-length codewords $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n$, each associated to one of the n possible transforms.

At this point, the encoder sends the codeword $\mathbf{c}_j$ associated with the true transform to the decoder, which may or may not be the codeword associated with the highest probability output by the MLP. Of course, the code is effective when the MLP confidence values are precise. In fact, if $p_k < p_h$ then $l(\mathbf{c}_k) \geq l(\mathbf{c}_h)$, where $l(\cdot)$ indicates the codeword length. Thus, the higher p_j is, namely, the more the MLP accurately identifies the originating transform from which the QTCB derives, the shorter $l(\mathbf{c}_j)$ is, and vice-versa. As an example, Fig. 3 presents a best case scenario where the MLP outputs a high probability for the optimal SBGFT $\mathcal{G}_j$ and, as a consequence, the corresponding codeword $\mathbf{c}_j$ is just a single bit long.

The same process is replicated at the decoder end using the QTCB received. With $\hat{X}_j$, the decoder designs the identical Huffman code through the same MLP network. Then, the decoder can retrieve the correct transform index j through Huffman decoding of the received $\mathbf{c}_j$. We remark that this is possible since j is not needed during the decoding process of the bit-stream associated with $\hat{X}_j$. In fact, in our experiments, the entropy coding method employed to represent the quantized transform coefficients $\hat{X}_j$ remains consistent across all transforms. Therefore, it is independent of the index j, which is solely necessary for selecting the accurate inverse transform.

IV. EXPERIMENTAL RESULTS–MLP PERFORMANCE EVALUATION

In this Section, an experimental evaluation on the proposed MLP architecture for transform index coding is conducted. Sec. IV-A first describes the setup. Then, Sec. IV-B discusses the network training phase, and Sec. IV-C provides the obtained results on the test datasets. Finally, a brief experimental analysis aimed at showing how we reached the architecture of the proposed MLP is provided in Sec. IV-D.

A. SETUP OF THE EXPERIMENTS

In the experiments, two distinct data sources have been considered. The first source comprises various still image types, including *aerial*, *textures*, and *miscellaneous*, in both standard and high definition [28], [29]. We have divided these images into 8×8 blocks, thus assembling for this first collection a total of 1,362,033 pixel blocks. The second source is composed of four standard video signals: *BQMall*, *BasketballDrill*, *Mobcal*, and *Shields*. The 8×8 residual blocks as obtained from intra-frame prediction using the prediction modes in H.266/VVC have been collected. This second set comprises 483,067 blocks.

As described in Sec. II, to select the best graph for each block on the two data sources, an exhaustive search is employed, testing all 40 SBGFTs for each block. The resulting transform coefficients in the graph domain are then quantized using a given QP, and the induced MSE is calculated for each transform. In the experiments, the following quantization parameters $QP = \{25, 30, 35, 40\}$ have been set, resulting in the quantization step sizes $Q_{\text{step}} = \{12, 20, 36, 64\}$ given the relation $Q_{\text{step}} = 2^{(QP-4)/6}$.

To be able to appreciate the generality of the proposed solution, the QTCB associated with the smallest distortion is directly chosen instead of performing the usual RD optimization, that is, the optimal transform is selected by minimizing Eq. (1) when $\lambda = 0$. In fact, ignoring the rate in the optimization avoids restricting the analysis to a predetermined, specific entropy coding scheme. At the same time, this choice does not hamper the experimental conclusions on transform index representation, as we show how the solution is able to handle different λ values as well (see Sec. V).

In the end, 8 distinct datasets of QTCBs and optimal SBGFT labels are created (2 data sources and 4 Q_{step} values).

Each dataset is then randomly split into training and test sets in a 4 : 1 ratio.

B. MLP TRAINING PHASE

In order to choose the best MLP architecture for each training set, a k-fold ($k = 5$), cross-validated grid-search is constructed over an extensive parameter grid, testing different combinations of number of neurons, activation functions, optimizers, batch sizes, and strategies for over-fitting prevention. A clear constrain on the architecture optimization is to keep the network at a tractable computational cost.

The configuration giving the best results is composed by 4 hidden fully-connected layers, respectively with n, $n/2$, $n/4$, and $n/8$ neurons, with $n = 512$ for all the pixel datasets and $n = 1024$ for all the residual datasets. We refer to Sec. IV-D for a brief sample of the experimental analysis conducted to validate these architectural choices.

All the layers of the proposed architectures adopt a ReLu activation function, except for the last layer of both that implements a Softmax activation function. For training, the Nadam optimizer [30] with learning rate set to 0.001 and batch size equal to 100 consistently reported the best performance. Cross-entropy was used as the loss function, and the number of epochs was set to 100. To reduce overfitting, we found that it was beneficial to use drop-out layers which randomly set input units to 0, with a concurrent constraint on the weights for each hidden layer, bounding their maximum norm with a threshold.

The 8 datasets and the associated trained models can also be found in [21] alongside the source code for the experiments.

C. MLP TESTING PHASE

After training, the MLP architectures are evaluated on the test datasets by computing the average bit-length per block needed to represent the transform index. A performance comparison is conducted with respect to both fixed-length coding and the transform entropy $H = -\sum_{i=1}^{40(=n)} p(\mathcal{G}_i) \log_2 p(\mathcal{G}_i)$, where $p(\mathcal{G}_i)$ is the *a priori* probability that the i-th graph is the correct transform for a given residual block. Furthermore, we compare with the results obtained by the CNN proposed in [25], of course retrained with the datasets used here.

Fig. 4 presents the results for all datasets with the pixel (Fig. 4a) and residual (Fig. 4b) domains data, respectively. The average bit-length per block of the proposed fully-connected network shows a strong prediction ability, saving more than 50% with respect to both the fixed-length coding and the transform entropy. Note how the latter is always close to its theoretical maximum $\log_2 40 \approx 5.32$ bits. This is due to the almost uniform *a priori* probability distribution of the adopted graphs, which is a common characteristics for every Q_{step}.

The method also tops [25], whose performance tend to worsen for large quantization steps, especially for residual blocks, and become even worse than fixed-length coding. In fact, the corresponding QTCBs are extremely sparse, confirming that the convolutional filters are not able to learn and correctly estimate the different transforms, as explained in Sec. III-A.

The achieved transform index prediction accuracy is satisfactory, considering how challenging the problem is. For example, Fig. 5 illustrates the performance for the pixel blocks when $Q_{\text{step}} = 20$. Fig. 5a shows the confusion matrix, which is in fact almost diagonal. Moreover, Fig. 5b depicts the k-top accuracy, i.e., the probability of ranking the true graph index among the first k spots. Notably, the straight accuracy of the model ($k = 1$) is around 50%, and already surpassing 80% when $k = 4$.

As an additional remark, it is a fact that learning capabilities of a neural network architecture are surely bound to improve when input blocks sharing potential common features or characteristics can be identified through an additional input. Such input is naturally present when a residual intra-coding is performed, that is, the Prediction Mode (PM). In fact, it is reasonable to assume that the residuals deriving from a same PM can be described by a statistical model which can be simpler than that of the single model representing all the PMs at the same time.

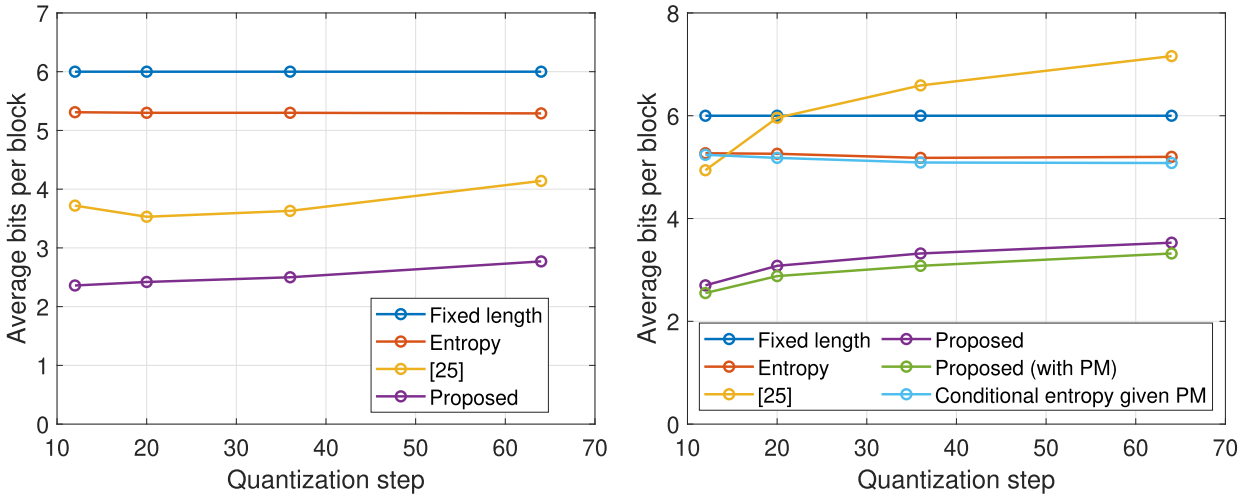
Therefore, an additional experiment has been conducted, for residuals derived datasets only, by appending the PM information to the input QTCBs. The MLP has been thus trained anew taking as input the quantized transform coefficients concatenated to the PM from which that block derives. The input vector size is thus now 65 long, the rest of the architecture being unchanged. Note that, in this scenario, the overall rate would not increase, since the PM is already encoded in the bit-stream.

The resulting performance is shown in Fig. 4b. The graph entropy conditional to the given prediction mode is also reported for a thorough comparison. Interestingly, it is observable a small but clear improvement (green curve) with respect to the original model (purple curve). This confirms that incorporating additional relevant information such as the PM as input to the MLP can enhance the overall performance of the system.

D. MLP ARCHITECTURE SELECTION

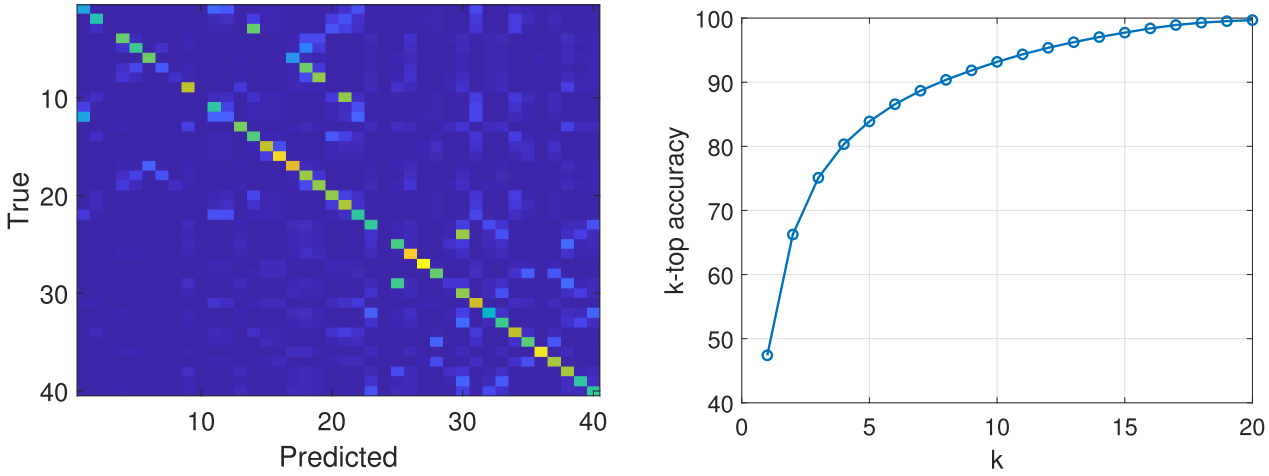
For the sake of completeness, this Section provides evidence that the MLP architecture proposed in Sec. IV-B outperforms other potential configurations. We focus on experiments conducted on residuals, as the same conclusions can be drawn in the pixel domain.

First, we compare the proposed structure, which consists of 4 hidden fully-connected layers with 1024, 512, 256, and 128 nodes, against two other structures: (i) an architecture with 3 hidden fully-connected layers with 1024, 512, and 256 nodes, and (ii) an architecture with 5 hidden fully-connected layers with 1024, 1024, 512, 256 and 128 nodes. The performance comparison is depicted in Fig. 6. The results



(a) Pixel blocks datasets (b) Residual blocks datasets

FIGURE 4. **Performance comparison of the proposed MLP for each data source.**



(a) Confusion matrix (b) k-top accuracy for $k \leq 20$

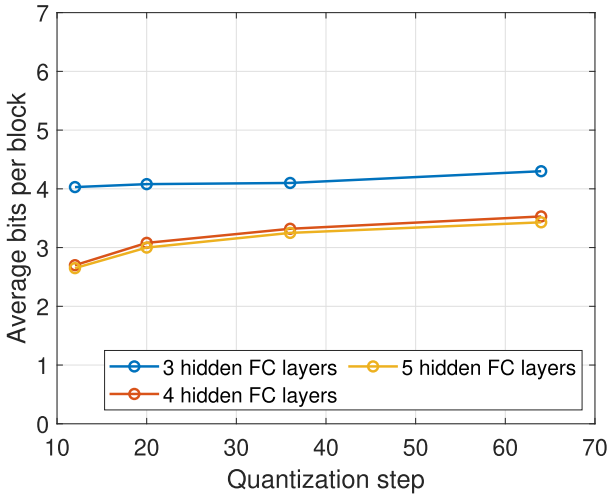
FIGURE 5. **Performance on the pixel blocks datasets - $Q_{step} = 20$.**

demonstrate that using fewer hidden fully-connected layers (i.e., 3 layers) results in a significant drop in performance. Conversely, adding an extra layer (making it 5 layers) provides only a marginal performance gain. Therefore, we opted for the configuration with 4 hidden fully connected layers, as it offers the best trade-off between performance and complexity.

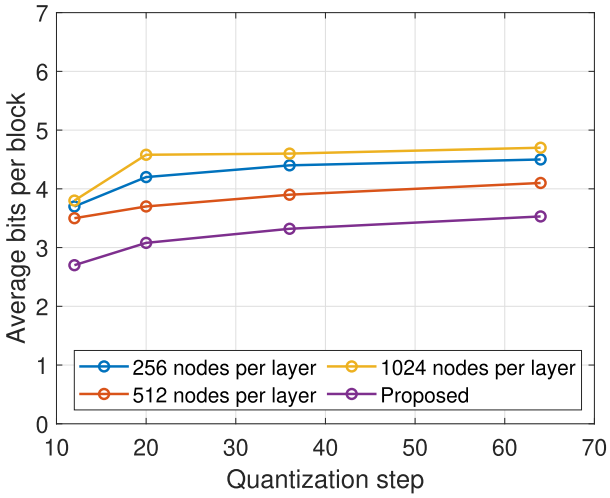
Next, we investigate the optimal number of nodes. In this experimental run, we compare three configurations: (i) 4 hidden layers with 256 nodes each, (ii) 4 hidden layers with 512 nodes each, and (iii) 4 hidden layers with 1024 nodes each. The results, shown in Fig. 6b, clearly indicate that using 512 nodes per layer provides the best performance. Interestingly, increasing the number of nodes from 512 to 1024 results in a performance drop even larger than using 256 nodes per layer, suggesting that the structure becomes too complex for this task. Based on these observations, we opted for the proposed MLP architecture, which employs a decreasing number of nodes from 1024 to 128, providing a comparable complexity than the one composed of 512 nodes per layer. The proposed MLP architecture outperforms all other configurations, including the latter with similar complexity, as shown by the purple curve.

V. EXPERIMENTAL RESULTS–CODING PERFORMANCE

In the last set of experiments, the objective is to demonstrate how advantageous it is to employ the SBGFTs in the intra-frame prediction residuals coding of a video compression scheme, both independently and coupled with the proposed MLP. In Sec. V-A, the comparison in terms of RD performance is given, while in Sec. V-B complexity considerations are provided.

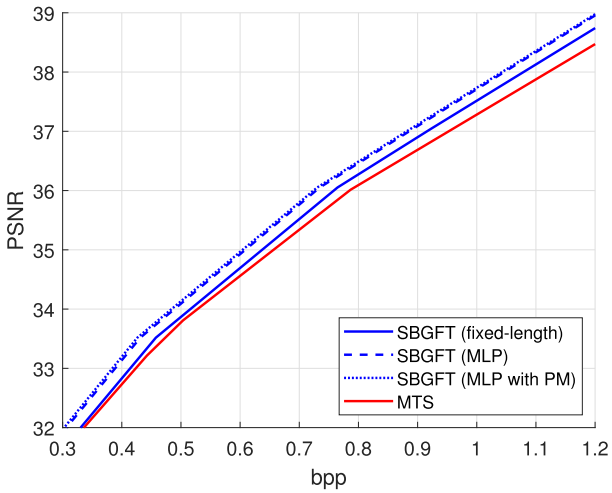


(a) Comparison on the number of hidden layers

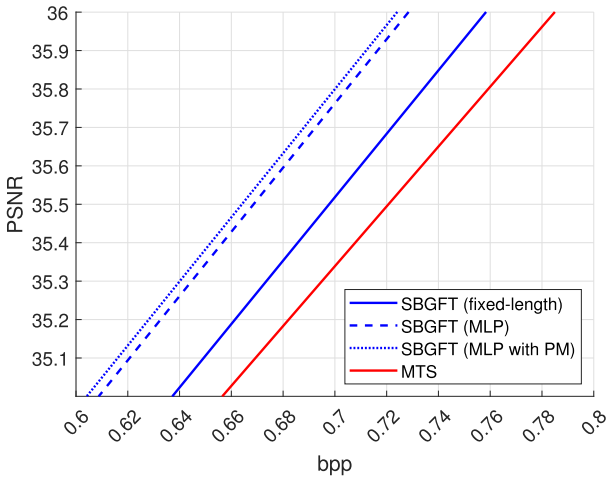


(b) Comparison on the number of nodes per layer

FIGURE 6. **Performance comparison between the proposed MLP architecture and other configurations.**



(a) RD curves obtained on the residuals dataset.



(b) Zoomed portion of (a).

FIGURE 7. **Comparison performance between the VVC's Explicit MTS method and the families of SBGFTs with different strategies to signal the graph index.**

A. RD EXPERIMENTS

To simulate how SBGFTs would perform within the VVC framework, we follow the processing steps outlined in Fig. 2. Therefore, we first take a set of images and video frames, not the same used in the aforementioned MLP training phase, and partition them into variable-sized residual blocks using VVC partitioning and intra-prediction.

Then, let us begin by describing the baseline method, which serves as the reference method for our comparison. In this case, transform coding of any residual blocks, regardless of their size, relies solely on the Explicit MTS method of VVC, using the same exhaustive approach as in Fig. 2, ensuring that the best transform in a RD sense is selected. To recap, the Explicit MTS method includes four additional transforms, in addition to the standard DCT-II. These extra transforms are the DCT-VIII and the DST-VII taken in the 4 combinations of horizontal and vertical directions. Thus, for each block, we select the transform that yields the smallest Lagrangian cost function as in Eq. (1), with:

$$\lambda = 0.57 \cdot 2^{(Q_{\text{step}}-12)/3} \tag{2}$$

which is identical to the VVC rate-control function.

TABLE 1. ***BD*** **rate associated with Fig. 7. The simplified VVC-like codec is taken as baseline.**

	Fixed-length	MLP	MLP with PM
Δ PSNR	0.15	0.39	0.43
Δ rate	3.69%	7.21%	7.92%

Regarding quantization, the previously considered QP parameters have been employed. The quantized coefficients are then entropy coded through arithmetic coding. The indices of the optimal transforms for each block are represented with the corresponding MTS flags adopted in the VVC standard.

The above scheme is compared with the same coding framework, but using the SBGFTs as the primary transforms core for 8×8 blocks, while keeping the process unchanged for blocks of other sizes where the Explicit MTS method is still employed. Rate control is still driven by Eq. (2). In this case, for the SBGFTs that are applied on 8×8 blocks, the optimal transform indices are coded through the proposed MLP predictive approach.

Fig. 7 depicts the rate-distortion curves associated with the VVC Explicit MTS method (red line) and the SBGFTs (blue line). In particular, the solid line is a control line indicating the performance when the graph indices are represented simply through fixed-length coding. The dashed line shows the performance when the MLP is used instead to build the variable-length codewords per block by taking the quantized transform coefficients as input.

Even when employing fixed-length encoding, the 40 SBGFTs already outperform the VVC Explicit MTS method, reporting an average gain of 0.15dB and requiring -3.69% of bit rate on average based on the Bjøntegaard's metric (*BD* rate), as presented in Table 1.

By constructing the graph index code exploiting the quantized transform coefficients with the MLP allows to even increase such gain, reaching an average gain of 0.39dB and requiring -7.21% of bit rate on average. We recall that the models have been trained on QTCBs selected using Eq. (1) with $\lambda = 0$. This underscores how well the trained models generalize to other λ values, in other words, they are not rigidly tied to the specific entropy coding scheme adopted.

Furthermore, the dotted line in Fig. 7 shows the resulting performance when the graph index is predicted by the MLP exploiting the PM as additional input. This modification leads to a further slight performance gain, which is better highlighted in Fig. 7b. Overall, in this case an average gain of 0.43dB is achieved, while saving 7.92% of bit rate on average, as reported in Table 1.

TABLE 2. Percentage increase in time complexity for encoding ($\Delta T_{\mathcal{E}}$) and decoding ($\Delta T_{\mathcal{D}}$) when incorporating the SBGFTs with/without MLP into a VVC-like framework. For decoding, the values on the left pertain to the computed computational times in the simplified VVC-like framework, while those on the right represent an estimate of the time complexity increase in the actual VVC framework.

	$\Delta T_{\mathcal{E}}$	$\Delta T_{\mathcal{D}}$	
		Computed	Estimated
SBGFT	2.5%	0%	0%
SBGFT with MLP	3.6%	120%	10%

B. COMPLEXITY

In this Section, we provide an analysis of the computational complexity linked to the SBGFTs and the proposed MLP for signaling the graph transform index.

Table 2 shows the time complexity relative increase using the VVC-like codec as the baseline. In the encoding stage, the time comparison is easy to perform since for the VVC-like coder it is just needed to measure the overall frame encoding time when the block coding transform core and the associated rate-distortion optimization is replaced. In contrast, only a simplified version of the decoding process has been implemented, namely, the inverse transforms and quantization. To allow estimating the impact of increased computational complexity on overall decoding time, we refer to the VVC complexity breakdown analyses found in [31] and [32], which are also leveraged here to validate the encoding times.

Without resorting to the MLP-based transform index prediction, the encoding time increases by a slight $\Delta T_{\mathcal{E}} = 2.5\%$ margin, due to the exhaustive search on 40 fast transforms instead of just 5 as in VVC. This figure is consistent with the fact that this process is applied on about 40% of the frame area occupied by 8×8 blocks. Additionally, it corresponds to the average time spent by the encoding process performing block coding transforms as opposed to other tasks such as intra-prediction and block partitioning, which also may involve RD optimization and transform computations (see [31, Fig. 3], [32, Fig. 1]).

Of course, it is obvious that an exhaustive search among the potential transforms is computationally inefficient and it causes huge encoding computational times in VVC as well. This issue is currently the subject of a lot of research, e.g., [33]. We remark that when employing SBGFTs too, recently proposed methods leveraging graph representation such as [34] could be used to reduce the computational burden of transform selection.

However, this computational issue does not impact the decoding stage, where the complexity remains low. In the examined case of using a SBGFT instead of a VVC primary transform, when the MLP is not employed there is no noticeable penalty in decoding time $\Delta T_{\mathcal{D}}$, as expected, because a single fast inverse transform is applied in either case.

On the other hand, when MLP-based transform index coding is used, the encoding time increases by 3.6%, indicating that the MLP carries more or less half the complexity of the SBGFTs selection process. Consequently, MLP usage more than doubles the 8×8 block decoding times of the simplified implementation, adding $\Delta T_{\mathcal{D}} = 120\%$ on average.

However, by again considering that such blocks cover approximately 40% of the frame area and concluding from [31, Table 7], [32, Fig. 5] that about 12% of the decoding time computation on average is dedicated to the block inverse transformation, we estimate that the VVC intra

frame decoding time should increase by about only 10%, which is a fair price to pay for the RD gains previously shown.

VI. CONCLUSION

Image and video compression standards have explored the introduction of multiple representation models, such as multiple linear block transforms. However, limitations arise due to complexity reasons and the high signaling cost associated with this representation mode, leading to consideration of only a small set of alternative transforms. This restriction hampers adaptation capabilities to the data.

In this paper, we have proposed incorporating a large set of Symmetry-Based Graph Fourier Transforms (SBGFTs) into a framework for image and video intra-frame coding to address these challenges. Specifically, we have introduced a Multilayer Perceptron (MLP) architecture to derive the transform index with high confidence from the quantization pattern fingerprint, thereby reducing the average number of overhead bits required to signal the transform index.

Experimental results demonstrate an average bitrate reduction of -3.69% when using SBGFTs compared to the DCT variants used in VVC. This reduction increases to -7.92% when combined with the MLP system, while maintaining a manageable increase in computational complexity. These results show how it is indeed possible to enlarge the set of available transforms for image and intra-frame video coding with noticeable performance gain.

There are still challenges that need to be tackled for considering future inclusion of multiple transform coding into practical image and video compression algorithms. The extension of the proposed transforms set to variable size blocks is of course an evident need, and it is actually the matter of our current research, as modern compression standards rely on complex partitioning processes, as we mentioned. Also, as JPEG AI, the first AI-based end-to-end image compression standard is to be released soon, experimental comparisons within such pipeline can also provide interesting insights especially in terms of acceptable complexity.

REFERENCES

[1] D. Salomon, *Data Compression: The Complete Reference*, 2nd ed. London, U.K.: Springer, 2004.

[2] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. Berlin, Germany: Springer-Verlag, 1975.

[3] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[4] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Aug. 2000.

[5] A. Gnutti, F. Guerrini, N. Adami, P. Migliorati, and R. Leonardi, "A wavelet filter comparison on multiple datasets for signal compression and denoising," *Multidimensional Syst. Signal Process.*, vol. 32, no. 2, pp. 791–820, Apr. 2021.

[6] V. Bhaskaran and K. Konstantinides, *The H.261 Video Coding Standard*. Boston, MA, USA: Springer, 1995, pp. 195–206.

[7] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[8] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.

[9] *BPG Image Format*. Accessed: Mar. 5, 2024. [Online]. Available: https://bellard.org/bpg/

[10] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[11] R. D. Dony and S. Haykin, "Optimally adaptive transform coding," *IEEE Trans. Image Process.*, vol. 4, no. 10, pp. 1358–1370, Sep. 1995.

[12] X. Zhao, L. Zhang, S. Ma, and W. Gao, "Rate-distortion optimized transform for intra-frame coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 1414–1417.

[13] T. Biatek, V. Lorcy, P. Philippe, "Transform competition for temporal prediction in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 815–826, May 2018.

[14] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin, "Joint separable and non-separable transforms for next-generation video coding," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2514–2525, May 2018.

[15] A. Arrufat, P. Philippe, and O. Déforges, "Non-separable mode dependent transforms for intra coding in HEVC," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 61–64.

[16] A. Gnutti, F. Guerrini, R. Leonardi, and A. Ortega, "Coding of image intra prediction residuals using symmetric graphs," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 131–135.

[17] A. Gnutti, F. Guerrini, R. Leonardi, and A. Ortega, "Symmetry-based graph Fourier transforms: Are they optimal for image compression?" in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 1594–1598.

[18] K.-S. Lu and A. Ortega, "Fast implementation for symmetric non-separable transforms based on grids," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4109–4113.

[19] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[20] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[21] *Transform Index Prediction Based on MLP*. Accessed: Mar. 5, 2024. [Online]. Available: https://github.com/AlessandroGnutti/Transform-Index-Prediction-based-on-MLP

[22] A. Gnutti, F. Guerrini, R. Leonardi, and A. Ortega, "Symmetry-based graph Fourier transforms for image representation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2575–2579.

[23] F. Guerrini, A. Gnutti, and R. Leonardi, "InnerSpec: Technical report," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1–4.

[24] A. Gnutti, F. Guerrini, and R. Leonardi, "Combining appearance and gradient information for image symmetry detection," *IEEE Trans. Image Process.*, vol. 30, pp. 5708–5723, 2021.

[25] S. Puri, S. Lasserre, and P. Le Callet, "CNN-based transform index prediction in multiple transforms framework to assist entropy coding," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 798–802.

[26] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. ICLR*, 2017, pp. 1–27.

[27] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

[28] *The USC-SIPI Image Database*. Accessed: Nov. 15, 2022. [Online]. Available: http://sipi.usc.edu/database/

[29] *Image Compression Set*. Accessed: Nov. 15, 2022. [Online]. Available: http://imagecompression.info

[30] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–4.

[31] F. Bossen, K. Sühring, A. Wieckowski, and S. Liu, "VVC complexity and software implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3765–3778, Oct. 2021.
[32] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj, and M. R. Hashemi, "Complexity analysis of next-generation VVC encoding and decoding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3134–3138.
[33] A. Wieckowski, B. Bross, and D. Marpe, "Fast partitioning strategies for VVC and their implementation in an open optimized encoder," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2021, pp. 1–5.
[34] K.-S. Lu, A. Ortega, D. Mukherjee, and Y. Chen, "Efficient rate-distortion approximation and transform type selection using Laplacian operators," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 76–80.

ALESSANDRO GNUTTI (Member, IEEE) received the M.S. degree (cum laude) in telecommunications engineering and the Ph.D. degree in information engineering from the University of Brescia, Italy, in 2014 and 2017, respectively. In 2017, he was a Visiting Fellow with the University of Southern California, where he worked on graph signal processing. In 2024, he was a Visiting Fellow with National Yang Ming Chiao Tung University, with a focus on end-to-end image and video compression. He is currently a Tenure-Track Assistant Professor with the Department of Information Engineering, University of Brescia. His primary research interests include signal and image representation, transform and graph analysis for image and video compression, end-to-end image and video compression, and coding for machines.

FABRIZIO GUERRINI (Member, IEEE) received the M.S. degree (cum laude) in electronic engineering and the Ph.D. degree in information engineering from the University of Brescia, Italy, in 2004 and 2008, respectively. He is currently with the Department of Information Engineering, University of Brescia. His main research interests include image and video processing and applications, including transform coding and feature extraction, image security and watermarking, and pattern analysis.

RICCARDO LEONARDI (Fellow, IEEE) received the Diploma and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 1984 and 1987, respectively. He was a Researcher with UC Santa Barbara and Bell Laboratories, from 1987 to 1991. Since 1992, he has been with the University of Brescia, Italy, where he established the Signal, Imaging, Networking and Communications (SINC) Group. He conducts research in signal and image representation for visual communications and visual content protection. He is an Expert in machine learning tools with application to multimedia content analysis, and medical imaging. He is an Expert Evaluator for the European Commission. He is currently the President of the Italian Telecommunication and Information Technology Association (GTTI).

Open Access funding provided by 'Università degli Studi di Brescia' within the CRUI CARE Agreement