Adaptive Chroma Prediction Based on Luma Difference for H.266/VVC

Junyan Huo, *Member, IEEE*, Danni Wang, Hui Yuan, *Senior Member, IEEE*,
Shuai Wan, *Member, IEEE*, and Fuzheng Yang, *Member, IEEE*

***Abstract*— Cross-component chroma prediction plays an important role in improving coding efficiency for H.266/VVC. We use the differences between reference samples and the predicted sample to design an attention model for chroma prediction, namely luma difference-based chroma prediction (LDCP). Specifically, the luma differences (LDs) between reference samples and the predicted sample are employed as the input of the attention model, which is designed as a softmax function to map LDs to chroma weights nonlinearly. Finally, a weighted chroma prediction is conducted based on the weights and chroma reference samples. To provide adaptive weights, the model parameter of the softmax function can be determined based on the template (T-LDCP) or offline learning (L-LDCP), respectively. Experimental results show that the T-LDCP achieves BD-rate reductions of 0.34%, 2.02%, and 2.34% for the Y, Cb, and Cr components, and the L-LDCP brings 0.32%, 2.06%, and 2.21% BD-rate savings for Y, Cb, and Cr components, respectively. The L-LDCP introduces slight encoding and decoding time increments, i.e., 2% and 1%, when integrated into the latest VVC test model version 18.0. Besides, the LDCP can be implemented by a pixel-level parallelization which is hardware-friendly.**

***Index Terms*— Weighted chroma prediction, cross-component prediction, versatile video coding, video coding, softmax function.**

I. Introduction

THE development of video services is to meet the growing demand for high-quality visual experience [1], which brings challenges for network bandwidth and storage capacity. Therefore, efficient compression of video content, especially high-quality video, is an essential research topic for industry and academia.

In the last two decades, MPEG and VCEG jointly published H.264/Advanced Video Coding (AVC) [2] and H.265/High Efficiency Video Coding (HEVC) [3], [4], which are widely used in the video industry. To further improve the coding efficiency, Joint Video Experts Team (JVET) developed the latest video coding standard, i.e., H.266/Versatile Video Coding (VVC) [5], [6], and finalized in 2020 [7]. The Audio Video coding Standard (AVS) workgroup of China published the 3rd generation of video coding standard (namely AVS3) [8] in 2021, bringing significant coding improvement compared with its predecessors. Alliance for Open Media (AOM) recently launched explorations for the next-generation coding standard beyond Alliance Video 1 (AV1) [9]. These latest video coding standards achieve excellent performance by combining many sophisticatedly designed coding algorithms [10], [11], [12].

In the prediction module [13], [14], one consensus of these standards is to design efficient chroma prediction by exploiting color component redundancies, e.g., cross-component linear model (CCLM) [15] in H.266/VVC, two-step cross-component prediction mode (TSCPM) [16] in AVS3, and chroma from luma (CFL) [17] in AV1. These algorithms establish the relationship between the luma and chroma components and then predict the chroma in a block by the reconstructed luma. Specifically, linear models (LMs) are adopted to describe the relationship between luma and chroma within a block. These cross-component prediction (CCP) algorithms provide significant performance improvement in addition to the conventional angular prediction [13].

Recently, refined CCP models have been investigated to exploit cross-component redundancies and improve coding efficiency. The prediction models can be deduced based on the multiple hypothesis assumptions (MHAs) [15], [16], [17], [18], [19], [20], [21], [22], [23] or the neural networks (NNs) [24], [25], [26], [27], [28], [29], [30]. Specifically, the MHA-based models improved the prediction by adjusting the model parameters or conducting the prediction model in a smaller range. These algorithms bring small additional coding gains with a slight complexity increase. With the rapid development of machine learning, extensive works have been explored on efficient video compression or quality improvement with NN [31], [32], [33] in which convolutional neural networks (CNN) are employed to establish the CCP models. Benefiting from the nonlinear operation, higher coding efficiency can be achieved, but the resulting high complexity hinders the practical applications.

In this paper, we introduce the differences between reference samples and the predicted sample to design the CCP model. Fig. 1 depicts a typical image block with its reference regions.

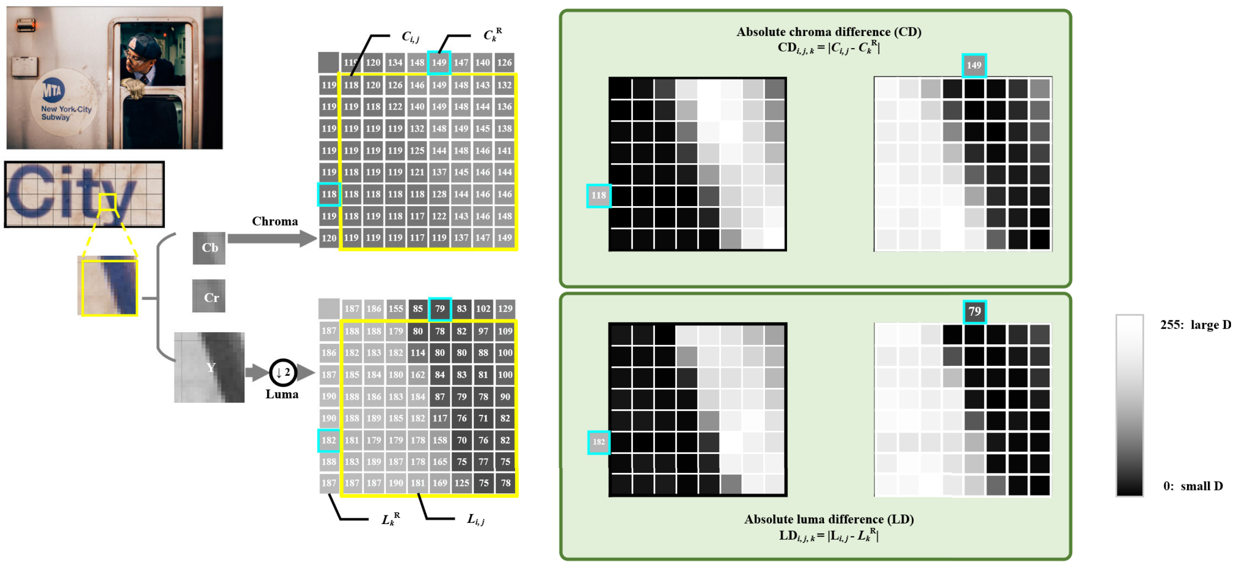
Manuscript received 11 December 2022; revised 29 May 2023; accepted 1 November 2023. Date of publication 13 November 2023; date of current version 20 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62222110, Grant 62172259, Grant 62101409, Grant 62171353 and Grant 62371358; and in part by the OPPO Research Fund. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chaker Larabi. *(Corresponding author: Junyan Huo.)*

Junyan Huo, Danni Wang, and Fuzheng Yang are with the State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: jyhuo@mail.xidian.edu.cn; danniwang_xidianu@163.com; fzhyang@mail.xidian.edu.cn).

Hui Yuan is with the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: huiyuan@sdu.edu.cn).

Shuai Wan is with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China, and also with the School of Engineering, Royal Melbourne Institute of Technology, Melbourne, VIC 3001, Australia (e-mail: swan@nwpu.edu.cn).

Digital Object Identifier 10.1109/TIP.2023.3330607


City
Cb
Cr
Y
Chroma
↓2
Luma
$C_{i,j}$
C_k^{R}
L_k^{R}
$L_{i,j}$
Absolute chroma difference (CD)
$\mathrm{CD}_{i,j,k} = |C_{i,j} - C_k^{\mathrm{R}}|$
149
118
79
182
Absolute luma difference (LD)
$\mathrm{LD}_{i,j,k} = |L_{i,j} - L_k^{\mathrm{R}}|$
255: large D
0: small D

	119	120	134	148	149	147	140	126
119	118	120	126	146	149	148	143	132
119	119	118	122	140	149	148	144	136
119	119	119	119	132	148	149	145	138
119	119	119	119	125	144	148	146	141
119	118	119	119	121	137	145	146	144
118	118	118	118	118	128	144	146	146
119	118	119	118	117	122	143	146	148
120	119	119	119	117	119	137	147	149

	187	186	155	85	79	83	102	129
187	188	188	179	80	78	82	97	109
186	182	183	182	114	80	80	88	100
187	185	184	180	162	84	83	81	100
190	188	186	183	184	87	79	78	90
190	188	189	185	182	117	76	71	82
182	181	179	179	178	158	70	76	82
188	183	189	187	178	165	75	77	75
187	187	187	190	181	169	125	75	78

Fig. 1. Absolute chroma differences (CDs) and absolute luma differences (LDs) of samples in the block with two representative reference samples. $C_{i,j}$ and $L_{i,j}$ are the chroma and luma intensities of the sample located at (i, j) of the block, C_k^{R} and L_k^{R} are the k-th elements of reference vector. CDs and LDs are rescaled to [0, 255] for better visualization. A darker intensity indicates a higher correlation between the sample in the block and the reference sample.

To align with the size of chroma, the luma component is downsampled by 2 for YCbCr 4:2:0 videos.

As shown in Fig. 1, we first calculate the chroma differences (CDs) between the two representative reference samples and samples in the block. It is observed that, if the chroma intensity of the reference sample is similar to those of samples in the block, small CDs can be derived. When a CD is with a large intensity, there will be a large difference between the sample in the block and the reference sample. Based on this observation, the CD can be used as an attention criterion to evaluate the contribution of reference samples during chroma prediction, i.e., a smaller CD corresponds to a larger weight.

In Fig. 1, we can also see that, when the CD is small, the corresponding LD usually has a small intensity, indicating a high correlation between the reference sample and the sample to be predicted. It is worth noting that the collocated luma block is available at both the encoder and decoder, and LD is easily obtained. Therefore, a refined chroma prediction, namely luma difference-based chroma prediction (LDCP), is proposed in this paper. We design a simple nonlinear attention model, take LDs as the input of the attention model to derive the weights, and then predict the chroma samples in the block using the chroma reference samples and the attention weights. Different from existing CCP algorithms based on a linear model of luma, we derive the predicted sample using a weighted model of the chroma reference samples, where the weights are derived by a LD-based nonlinear attention model. The contributions of this paper are listed as follows.

1) We use LDs to characterize the correlation between the reference and predicted samples. Compared to a complicated attention model based on NN, LD is a straightforward attention criterion.
2) With the LDs, an attention model is designed as a softmax function to map the LDs to the chroma weights in a nonlinear manner. The model parameter of the softmax function is tuneable based on a template region (namely T-LDCP) or offline learning (namely L-LDCP) to provide adaptive weight mapping.
3) A refined chroma prediction is proposed based on the attention weights and the chroma reference samples. Since the weights are derived based on the LDs between each predicted sample and the reference samples, the weights are customized for each sample in the block.
4) Compared with the latest reference software of H.266/VVC (VTM 18.0), the T-LDCP provides 0.34%, 2.02%, and 2.34% BD-rate savings for Y, Cb, and Cr components, and the L-LDCP provides 0.32%, 2.06%, and 2.21% BD-rate savings for Y, Cb, and Cr components, respectively. Moreover, the L-LDCP has a merit of low complexity, i.e., only 2% encoding time and 1% decoding time increments are induced. Besides, the LDCP can be implemented by a pixel-level parallelization which is hardware-friendly.

To the best of our knowledge, it is the first time that the differences between reference samples and the predicted samples are explicitly constructed and used to guide the attention model. In addition, only one softmax function is adopted as the attention model to provide a nonlinear feature, which is computationally simple yet efficient.

The remainder of this paper is organized as follows. Section II summarizes the state-of-the-art CCP algorithms. The proposed attention model for chroma weights is described in Section III. Section IV presents the principle of the LDCP and proposes two algorithms, i.e., T-LDCP and L-LDCP, to determine the model parameter in LDCP. Experimental results and analyses are given in Section V. Section VI concludes the paper.

II. Related Works

For chroma prediction, cross-component correlation is used to improve coding efficiency. This correlation was initially employed to compress videos in RGB format [18]. Recently, coding tools based on the color component correlation have been exploited in the latest video coding standards. A block adaptive CCP algorithm for RGB 4:4:4 format was proposed in [19] and adopted into the range extensions of H.265/HEVC. The CCLM solution of H.266/VVC employs an LM to predict the chroma using the collocated luma. The model parameters of LM are deduced according to the reference samples [15] to avoid signaling. In our prior work [20], four specific reference samples, based on the minimum geometry distance, were determined to derive the parameters. With the merit of unified operations for various coding units (CUs), it was adopted in H.266/VVC. In addition, Li et al. [16] proposed an enhanced LM derivation which has been integrated into H.266/VVC. In AVS3, the TSCPM derives its model parameters by the downsampled luma and the chroma of the reference samples [16]. In the CFL of AV1, the predicted chroma is decomposed to DC and AC parts [17]. The DC value is predicted from chroma reference samples while an LM based on the mean-removed luma derives the AC value.

Besides the algorithms adopted by existing video coding standards, there are also several other algorithms. We classify the state-of-the-art algorithms into two categories according to the model design strategies.

1) Prediction Algorithms Based on the MHAs: The CCP algorithms in [15], [16], [17] are established based on a simple LM assumption. Zhang et al. [21] proposed to improve the CCP by introducing three prediction solutions with multiple linear models (MMLM), multiple downsampling filters, and a combined CCLM-intra prediction. In MMLM, the top and left reference samples were classified into two groups, and a particular set of LM parameters was derived for each group. The luma samples in the block were also classified to predict the associated chroma samples with the corresponding LM. This algorithm could improve the prediction accuracy, especially for the blocks with complex textures. Lainema et al. [22] designed a series of LMs by adjusting the model parameters with offsets. In this way, the linear feature of the coding block can be characterized by finding an optimal offset. Convolutional cross-component model (CCCM) [23] is an effective cross-component prediction through modeling the relationship between luma and chroma flexibly using the neighboring template. Based on the CFL in AV1, we proposed an improved DC prediction based on the luma distribution [24].

2) Prediction Algorithms Based on NNs: Li et al. [25] proposed a hybrid NN chroma prediction, where the convolutional layers and the fully connected layers were employed to remove the spatial and color component redundancies. In [26], features for the luma and chroma were extracted independently by CNNs and combined via a fully connected layer to establish the cross-component chroma prediction. Zhu et al. [27] designed a chroma prediction based on CNN (CNNCP). For a coding tree unit (CTU), the predicted chroma was built according to its neighboring three CTUs. This design provides a high coding efficiency but requires large memory. An attention-based network was designed in [28] to establish the relationship between the reference and predicted samples. Based on [28], refined spatial information [29] was fed into the network as an additional input to improve the prediction capability of the network. In addition, Blanch et al. [30] analyzed the complexity of the attention-based intra-prediction [28] and proposed several simplifications while maintaining similar coding performance. Moreover, an NN-based CCP method was proposed in [31] to efficiently integrate the neighboring reconstructed samples and the collocated luma samples into a network.

Generally speaking, NN-based models bring more potential coding gains than MHA-based models. To introduce the nonlinearity in NN-based models, the convolution operation usually appends with a nonlinear activation function. This design can solve the problem of insufficient expression of linear models. Rectified linear unit (ReLU) and its variants, sigmoid, softmax, and tanh are the widely used activation functions. In particular, the softmax function has an excellent ability to generate the probability distribution. For the complexity of NN-based models, it was reported that their complexity is about nine times of that of MHA-based models [27], [30], [31]. Instead of a complicated NN model, we use a single softmax function as a lightweight attention model to map LDs to chroma weights nonlinearly. With the well-designed chroma weights, a simple linear prediction is conducted for chroma prediction. The proposed algorithm is characterized as a nonlinear CCP model with high prediction accuracy and extremely low complexity, which is easily implemented.

III. Proposed Attention Model for Chroma Weights

A. Lightweight Attention Model Based on the Luma Difference

In intra prediction, samples located at the left and top reference regions are employed to derive the predicted samples. In conventional angular prediction, given a specific angular, the predicted signal is derived based on partial reference samples. In NN-based models, all the reference samples are designed as input to predict samples in the block. In this way, the importance of each reference sample to each predicted sample needs to be determined. Blanch et al. [28] proposed to measure the importance (i.e., weight) by an attention network. However, the complexity of the attention model is high. For simplicity, we put forward to use LDs as a simple and efficient attention criterion.

Fig. 2 depicts a representative 4×4 chroma block with its reference samples. We denote the chroma reference samples as a vector $\mathbf{C}^{\mathrm{R}}$ and the k-th element as C_k^{R}. A sample's horizontal and vertical coordinates in the block are labeled as i and j, respectively. $C_{i,j}$ is the chroma intensity of the sample located at (i, j).

To investigate the importance of the reference samples during the chroma prediction, Fig. 3 visualizes the CDs between four samples (highlighted with yellow borders in Fig. 2) and the reference samples. The x-axis shows the indices of the individual reference samples in $\mathbf{C}^{\mathrm{R}}$ while the y-axis indicates the intensities of CD. In the curves, when $|C_{i,j} - C_k^{\mathrm{R}}|$ is small, $C_{i,j}$ and C_k^{R} corresponds to higher correlation and C_k^{R} has a stronger ability to predict $C_{i,j}$. Therefore, larger weights are preferred to assign for the reference samples with smaller CDs.

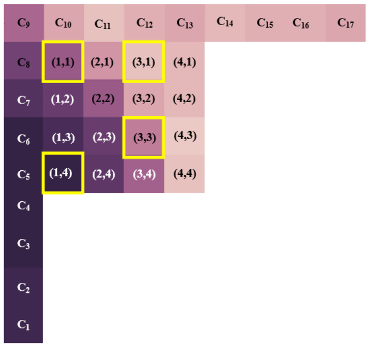


Fig. 2. A 4 × 4 chroma block with its reference samples. $C_{1,1}$, $C_{1,4}$, $C_{3,1}$, and $C_{3,3}$, highlighted with yellow borders, are four representative original chroma samples in the block. The chroma reference samples form a reference vector.

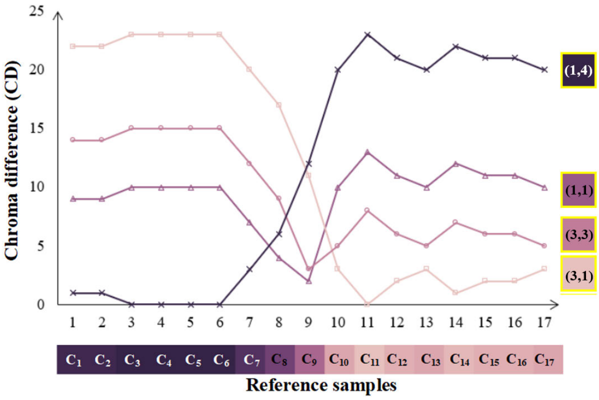


Fig. 3. The CDs between reference samples and four specific samples in the block. A smaller CD corresponds to a higher correlation.

In video coding, CDs are unavailable due to the absence of $C_{i,j}$. A predicted chroma sample can be derived as

$$C^{\mathrm{P}}_{i,j} = \mathrm{CCP}(\hat{Y}_{i,j}), \tag{1}$$

where $C^{\mathrm{P}}_{i,j}$ and $\hat{Y}_{i,j}$ are the predicted chroma sample and the collocated luma sample, respectively, $\mathrm{CCP}(\cdot)$ is the cross-component prediction model.

With the same model, the chroma reference sample can be approximately represented as

$$C^{\mathrm{R}}_{k} \approx \mathrm{CCP}(\hat{Y}^{\mathrm{R}}_{k}), \tag{2}$$

where C^{R}_{k} and $\hat{Y}^{\mathrm{R}}_{k}$ represent the chroma and luma of the k-th reference sample, respectively.

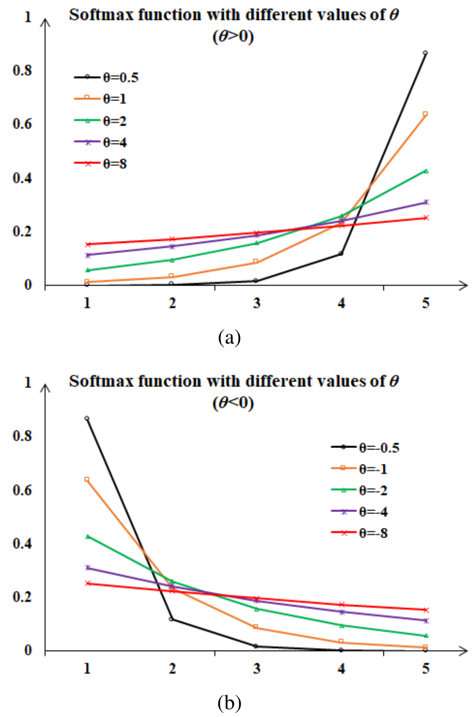
Consequently, a predicted CD, denoted as $|\Delta C^{\mathrm{P}}|$, can be derived as

$$|\Delta C^{\mathrm{P}}| = |C^{\mathrm{P}}_{i,j} - C^{\mathrm{R}}_{k}| \approx \mathrm{CCP}(|\Delta Y_{i,j,k}|), \tag{3}$$

where $|\Delta Y_{i,j,k}|$ represents the difference between $\hat{Y}_{i,j}$ and $\hat{Y}^{\mathrm{R}}_{k}$, i.e., LD. According to (3), the CD can be predicted from its corresponding LD. If the LD is small, the predicted CD will be small in most cases, indicating a high correlation between the reference and the predicted sample.

Based on the above analysis, a weighted chroma prediction is proposed as

$$C^{\mathrm{P}}_{i,j} = \mathbf{w}^{\top}_{i,j}\mathbf{C}^{\mathrm{R}}, \tag{4}$$



(a)

(b)

Fig. 4. Typical curves of softmax with different values of θ. (a) θ with positive values. (b) θ with negative values.

where $\mathbf{w}_{i,j}$ is the weight vector of reference samples for (i, j) and derived by

$$\mathbf{w}_{i,j} = \mathrm{F}(\mathbf{LD}_{i,j}), \tag{5}$$

where $\mathrm{F}(\cdot)$ is the attention model to map the LD vector to the weight vector.

B. Nonlinear Attention Model Using Softmax Function

The determination of the weight vector in (5) can be roughly regarded as a multi-classification [35] problem. A common solution is to apply a classifier to accomplish the task. One of the most popular classifiers is the softmax function which is defined as

$$\mathrm{softmax}(\mathbf{x})_k = \frac{\exp(\frac{x_k}{\theta})}{\sum_n \exp(\frac{x_n}{\theta})}, \tag{6}$$

where θ is the model parameter.

For an input vector, $\mathbf{x}$ with K elements, the softmax function transforms it into a new vector whose elements satisfy

$$\sum_{k=1}^{K} \mathrm{softmax}(x_k) = 1, \quad 0 \leq \mathrm{softmax}(x_k) \leq 1. \tag{7}$$

Since each element is within [0, 1] and the sum of elements is 1, the softmax function is suitable to represent the weights of reference samples. To better tackle the chroma prediction task, adaption of the softmax function is essential and discussed as follows.

1) Sign Adaption of θ: In the NN-based algorithms, θ in the softmax function is usually a constant [28]. To illustrate the characteristics of softmax, Fig. 4 provides several softmax

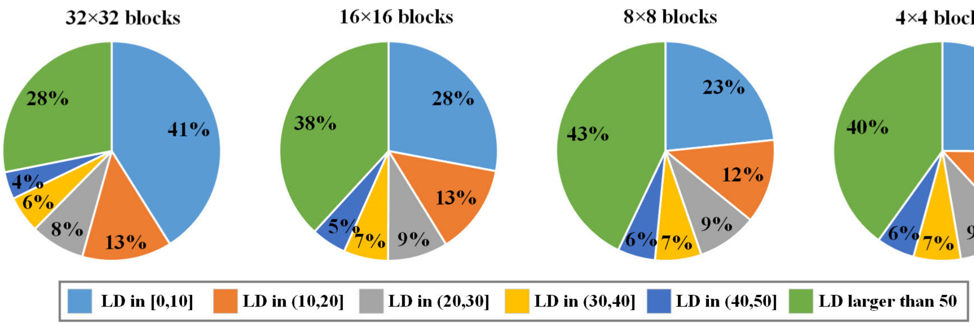


Fig. 5. The LD distributions of different block sizes.

curves with different values of θ, in which the values are set to be positive in Fig. 4 (a) and negative in Fig. 4 (b).

Taking the input vector $\mathbf{x}$ with five elements [0, 1, 2, 3, 4] as an example, the output vector is proportional to the input vector when θ is positive and inverse proportional to the input vector when θ is negative. In the attention model, reference samples with smaller LDs are assigned larger weights. Therefore, θ in the proposed LDCP is set to be a negative value to represent this relationship.

2) Softmax With Different Values of θ: Fig. 4 (a) and (b) provide several curves of softmax function under different values of θ. We can see that θ determines the nonlinear features of the softmax curves. When the absolute value of θ is small, the outputs vary sharply with the inputs.

As defined in (5), the input vector of the attention model is $\mathbf{LD}_{i,j}$ and the output vector is $\mathbf{w}_{i,j}$. In the following, the characteristic of the LD vector is analyzed first.

In general, the LD distribution is related to the texture complexity of the video content. It is expected to be simpler and dominated by smaller LDs for blocks with less texture and vice versa. To verify the description, a preliminary experiment was conducted. We investigated the LD distributions of blocks with different sizes after coding. In H.266/VVC, blocks with different textures can be flexibly split into CUs in different sizes. Four representative block sizes, including 32×32, 16×16, 8×8, and 4×4, were analyzed. The range of LDs was divided into six intervals, i.e., [0, 10], (10, 20], (20, 30], (30, 40], (40, 50], and (50, $+\infty$). Three test sequences, i.e., $Campfire$, $ParkRunning3$, and $DaylightRoad2$, were encoded. Here, four typical quantization parameters (QPs), including 22, 27, 32, and 37, were used and the average LD distributions were calculated.

Fig. 5 depicts the LD distributions under different block sizes. We can see that the LD distributions are different for blocks with different sizes. Specifically, for blocks coded with 32×32 (indicating they are flat and easily predicted), the portion of LDs within [0, 10] is 41% on average, and LDs larger than 50 occupy 28%. For blocks coded with smaller sizes, the portion of small LDs decreases, and that of large LDs increases. Taking 4×4 blocks as an example, the portion of LDs within [0, 10] is 25% on average, and LDs larger than 50 reach 40%. Therefore, we can conclude that LDs of larger blocks mainly fall into the small range due to the simple and flat texture, while LDs of smaller blocks are characterized by large intensities due to the complex texture.

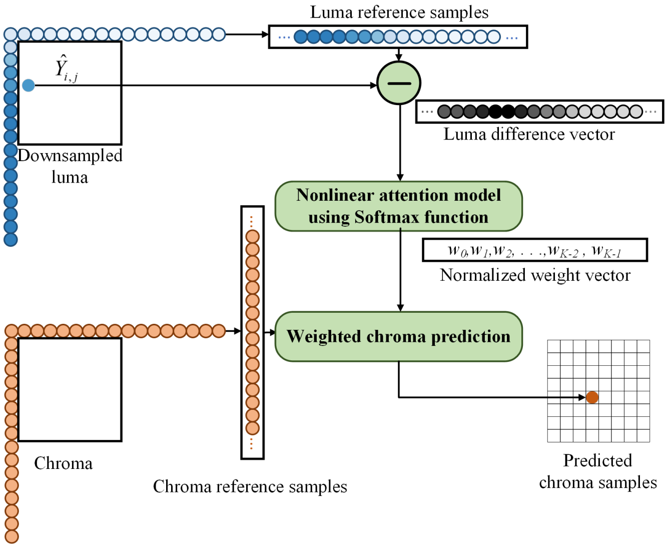


Fig. 6. The diagram of LDCP. LDCP takes the luma reference samples, the collocated luma in the block, and the chroma reference samples as the inputs and the predicted chroma samples as the output. For a block with W lines and H columns, K is equal to $2W + 2H + 1$.

Since the LD distributions for blocks with different sizes are different, the attention model needs to be designed according to the block sizes. To be specific, an adaptive attention model is designed for different block sizes through a customized model parameter of the softmax function.

IV. Proposed LD-Based Chroma Prediction

Fig. 6 provides the diagram of the proposed LDCP algorithm with the luma reference samples, chroma reference samples, and the collocated luma in the block as the inputs. Three modules are introduced to obtain predicted chroma samples, including the LD vector derivation, the attention model using softmax function, and weighted chroma prediction.

For a pixel at (i, j), the LD vector can be derived as

$$LD_{i,j,k} = |\Delta Y_{i,j,k}| = |\hat{Y}_{i,j} - \hat{Y}_k^{\mathrm{R}}|, \quad k \in [1, K], \tag{8}$$

where $LD_{i,j,k}$ is the k-th element, K is the number of reference samples.

Then, a nonlinear attention model is designed as a softmax function to map the LD vector to the normalized weight vector

$$w_{i,j,k} = \frac{\exp(-\frac{1}{|\theta|}|\Delta Y_{i,j,k}|)}{\sum_n \exp(-\frac{1}{|\theta|}|\Delta Y_{i,j,n}|)}, \tag{9}$$

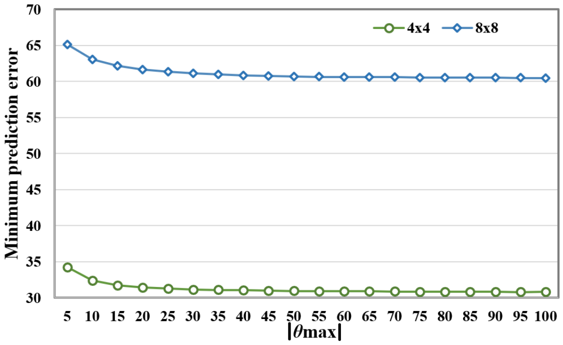


Fig. 7. The minimum prediction error with different ranges of Θ.

where $w_{i,j,k}$ is the k-th element of the weight vector.[1]

With the weight vector of (i, j), the predicted chroma sample can be obtained by a vector multiplication as

$$\begin{aligned} C^{\mathrm{P}}_{i,j} &= \mathbf{w}^{\top}_{i,j}\mathbf{C}^{\mathrm{R}} \\ &= w_{i,j,1}C^{\mathrm{R}}_{1} + w_{i,j,2}C^{\mathrm{R}}_{2} + \cdots + w_{i,j,K}C^{\mathrm{R}}_{K}. \end{aligned} \tag{10}$$

Eqs. (8), (9), and (10) are conducted for each pixel in the block to derive the predicted chroma block.

In video coding, efficient prediction minimizes the error between the original and predicted samples. In the proposed LDCP, the prediction error can be defined as

$$E = \mathrm{Error}(\mathbf{C}^{\mathrm{O}}, \mathbf{C}^{\mathrm{P}}), \tag{11}$$

where $\mathbf{C}^{\mathrm{O}}$ is the original chroma vector of the block, $\mathbf{C}^{\mathrm{P}}$ is the predicted chroma vector as

$$\mathbf{C}^{\mathrm{O}} = \begin{pmatrix} C_{1,1} \\ C_{1,2} \\ \vdots \\ C_{W,H} \end{pmatrix}, \mathbf{C}^{\mathrm{P}} = \begin{pmatrix} \mathbf{w}^{\top}_{1,1}\mathbf{C}^{\mathrm{R}} \\ \mathbf{w}^{\top}_{1,2}\mathbf{C}^{\mathrm{R}} \\ \vdots \\ \mathbf{w}^{\top}_{W,H}\mathbf{C}^{\mathrm{R}} \end{pmatrix}, \tag{12}$$

Error$(\cdot)$ is the error criterion that can be evaluated by mean square error (MSE) or sum of absolute transformed difference (SATD).

By assigning different values of $|\theta|$, Eq. (9) produces different attention models. Therefore, $\mathbf{C}^{\mathrm{P}}$ is a function of $|\theta|$. Given a set of $|\theta|$, the prediction error under different values of $|\theta|$ is evaluated and the optimal model parameter with the minimized prediction error can be determined as

$$|\theta|^{\mathrm{Opt}} = \underset{|\theta|\in\Theta}{\mathrm{argmin}}\, \mathrm{Error}(\mathbf{C}^{\mathrm{O}}, \mathbf{C}^{\mathrm{P}}(|\theta|)), \tag{13}$$

where $|\theta|^{\mathrm{Opt}}$ is the optimal model parameter, Θ is a set of $|\theta|$ and represented as $[|\theta_{\min}|, |\theta_{\max}|]$.

A. Range of $|\Theta|$

To analyze the influence of Θ on the prediction error, different ranges of $|\Theta|$ are discussed in this subsection. Without loss of generality, we assume the elements in Θ as integers and $|\theta_{\min}|$ is set to 1. Different ranges of $|\Theta|$ are implemented by assigning different values of $|\theta_{\max}|$.

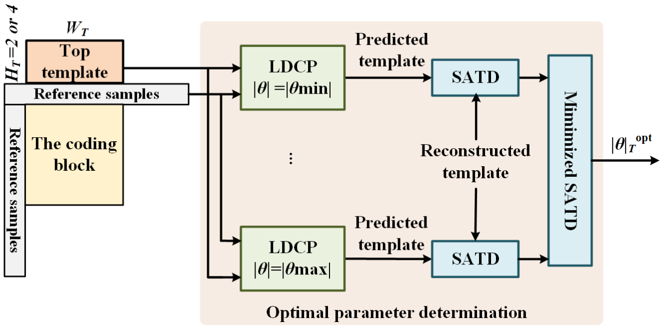


Fig. 8. The diagram of T-LDCP. SATD is calculated between the predicted chroma and the reconstructed chroma of the template. The model parameter with the minimized SATD is set to be the optimal parameter of the template.

In the experiment, test sequences were split into blocks of 4×4 or 8×8, respectively. Given a specific block, the predicted chroma was constructed using the proposed LDCP algorithm and the prediction error was measured by MSE. The minimum prediction error of Θ was determined according to Eq. (13).

Fig. 7 provides the minimum prediction error under different $|\Theta|$. The x-axis is the value of $|\theta_{\max}|$ and the y-axis is the minimum prediction error under different settings of $|\theta_{\max}|$. Here, $|\theta_{\max}|$ was set from 5 to 100 with an interval of 5. The two curves in Fig. 7 are the prediction error of 4×4 and 8×8 blocks, respectively. When we split test sequences into 4×4 blocks, the prediction error is smaller than that of 8×8 blocks. We can also see that the minimum prediction error decreases with the increment of $|\theta_{\max}|$. That means an extension of $|\theta|$'s range brings a smaller prediction error. However, the decrease of the prediction error becomes marginal when $|\theta_{\max}|$ is larger than 20. The data indicate that $[1, 20]$ is sufficient for Θ to find the optimal model parameter in the proposed LDCP.

As defined in (9), $|\theta|$ is a crucial parameter that decides the attention model. In the following two subsections, two algorithms are put forward to determine $|\theta|$ adaptively.

B. Template-Based LDCP (T-LDCP)

For each block, $|\theta|^{\mathrm{Opt}}$ can be determined according to (13). This information needs to be sent to the decoder at the expense of increased signaling overhead. Benefitting from a high spatial correlation between the neighboring regions and the block, we propose an LDCP scheme to derive $|\theta|^{\mathrm{Opt}}$ from a template, namely T-LDCP.

The diagram of T-LDCP is illustrated in Fig. 8. First, the template is extracted from the top and left neighboring regions. Taking the top template as an example, the size of the template depends on the size of the block. For a block of $W \times H$, the size of the template is $W \times 2$ when $H \leq 4$ and is $W \times 4$ for $H > 4$ cases.

With the template and reference samples, LDCP is employed to derive the predicted chroma of the template. In LDCP modules of Fig. 8, the model parameter, $|\theta|$, is set to $1, 2, \ldots, |\theta_{\max}|$, respectively. Here, $|\theta_{\max}|$ is set to 20 according to the analysis of Fig. 7. Then, the reconstructed chroma of the template is utilized to evaluate the performance of LDCP outputs. Specifically, SATDs between the LDCP

[1] To be consistent with the relationship in Fig. 4 (b) with a negative θ, (9) adds a negative sign in each exponential function and $|\theta|$ is discussed in the following part.

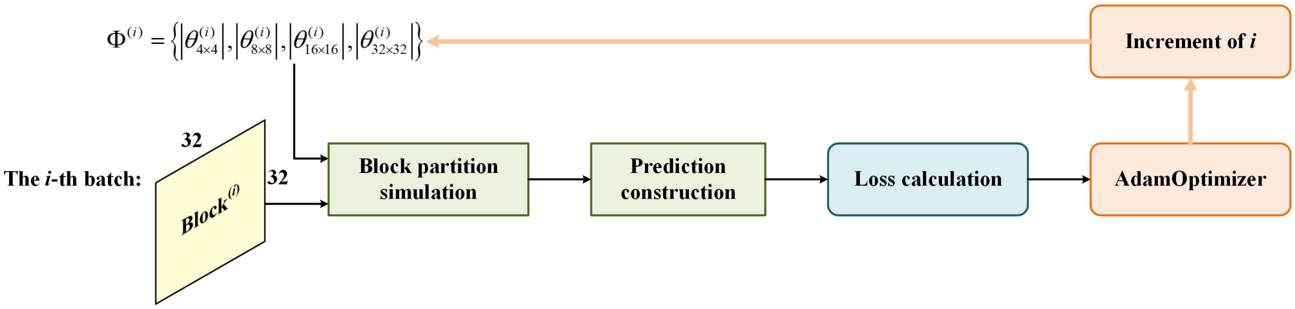

$\Phi^{(i)} = \{|\theta^{(i)}_{4\times4}|, |\theta^{(i)}_{8\times8}|, |\theta^{(i)}_{16\times16}|, |\theta^{(i)}_{32\times32}|\}$
Increment of *i*
The *i*-th batch:
32
32
Block$^{(i)}$
Block partition simulation
Prediction construction
Loss calculation
AdamOptimizer

Fig. 9. The training diagram of L-LDCP. For blocks of the i-th batch, $|\theta|$ of four typical sizes are used to launch the block partition simulation and prediction construction. With the prediction based on LDCP, the loss is calculated and $|\theta|$ is optimized using AdamOptimizer. Here, $|\theta|$ of four typical sizes are updated for the $(i+1)$-th batch.

TABLE I
CHROMA BLOCKS CLASSIFICATION IN L-LDCP

ClassIdx	Condition	Block size for training
1	$min(W, H) \leq 4$	4×4
2	$4 < min(W, H) \leq 16$	8×8, 16×16
3	$min(W, H) > 16$	32×32

outputs and the reconstructed template are calculated and $|\theta|^{\mathrm{Opt}}$ are determined according to (13). Finally, T-LDCP derives the predicted chroma in the block using the optimal attention model of the template.

T-LDCP provides flexible attention models for different coding blocks. Since the attention model from the template is available at both the encoder and decoder, the model parameter of T-LDCP can be implicitly derived without transmission. The concern of T-LDCP is the complexity which mostly comes from the optimal parameter determination.

C. Learning-Based LDCP (L-LDCP)

To perform the proposed algorithm with low complexity, we also propose a parameter determination through machine learning, namely L-LDCP. Based on the analysis of Section III-B, L-LDCP introduces different values of $|\theta|$ for different block sizes.

In H.266/VVC, flexible CU partition [36], [37] is designed where the width and height of CUs[2] can be any value of powers of two between 2 and 32. To facilitate training, it is widely used to group the diverse CU sizes into several classes [38], [39]. We classify chroma blocks into three classes, as shown in Table I. For the three classes, four typical blocks, including 4×4, 8×8, 16×16, and 32×32, are trained and the model parameter of each class is obtained according to the training results. The parameter to be trained is denoted as $\Phi = \{|\theta_{4\times4}|, |\theta_{8\times8}|, |\theta_{16\times16}|, |\theta_{32\times32}|\}$.

Fig. 9 provides the training diagram of L-LDCP. For blocks of the i-th batch, we employed $\Phi^{(i)}$ to launch the block partition simulation and LDCP prediction. With the predicted chroma of LDCP, the loss can be calculated and the model parameters are optimized through back-propagation to derive $\Phi^{(i+1)}$. AdamOptimizer was employed as the optimizer, and the learning rate and the batch size were 10^{-4} and 128, respectively.

As depicted in Fig. 9, three essential procedures are put forward in L-LDCP, including block partition simulation, prediction construction, and loss calculation. In the following, the procedures are introduced in detail.

1) Block Partition Simulation: To derive model parameters for different block sizes, a straightforward solution is to train $|\theta_{\mathrm{N}\times\mathrm{N}}|$ by splitting all the training data into $\mathrm{N}\times\mathrm{N}$ blocks, $\mathrm{N} = \{4, 8, 16, 32\}$. Besides, we should note that the CU partition in H.266/VVC is determined according to the rate-distortion optimization (RDO) criteria [40]. For example, flat content is characterized by a large block size, and a small block size is preferred for rich-texture content. To train $|\theta_{\mathrm{N}\times\mathrm{N}}|$ efficiently, a block partition simulation module is introduced in L-LDCP.

As depicted in Fig. 10, a 32×32 block is split into four typical block sizes. For a specific block of $\mathrm{N} \times \mathrm{N}$, the predicted chroma is constructed using the proposed LDCP based on $|\theta^{(i)}_{\mathrm{N}\times\mathrm{N}}|$. After that, the SATD of the LDCP output and the ground-truth chroma is calculated. To be close to the partition results in video coding, an additional penalty term, i.e., Partition cost_N $\times$ N, is designed to estimate the signaling cost. In this paper, partition costs are set to fix empirical values. Then, the cost for a $\mathrm{N} \times \mathrm{N}$ block is derived by accumulating SATD and Partition cost_N $\times$ N.

With the cost of different block sizes, the partition decision is determined as follows. For each 8×8 block, we compare the cost of one 8×8 block and the sum of costs with four 4×4 blocks, and the partition result is decided according to the minimized cost. We conduct the same procedure for each 16×16 block. Then, the minimized cost of four 16×16 blocks is compared with the cost predicted by one 32×32 block. Finally, the block partition with the minimized cost can be determined. Fig. 10 shows the partition decision for a 32×32 block of the i-th batch.

2) Prediction Construction: Fig. 11 provides a predicted chroma of 32×32 block, where the proposed LDCP is conducted for each specific block according to the block partition results. Taking an 8×8 block as an example, the corresponding predicted chroma is derived based on the reference samples, the collocated luma in the block, and $|\theta^{(i)}_{8\times8}|$.

It is valuable to note that $|\theta_{\mathrm{N}\times\mathrm{N}}|$ is employed for $\mathrm{N} \times \mathrm{N}$ blocks through block partition. This strategy helps to collect $\mathrm{N} \times \mathrm{N}$ blocks in a coding manner to train $|\theta_{\mathrm{N}\times\mathrm{N}}|$.

[2] The maximize size of a chroma CU in H.266/VVC is 32×32.

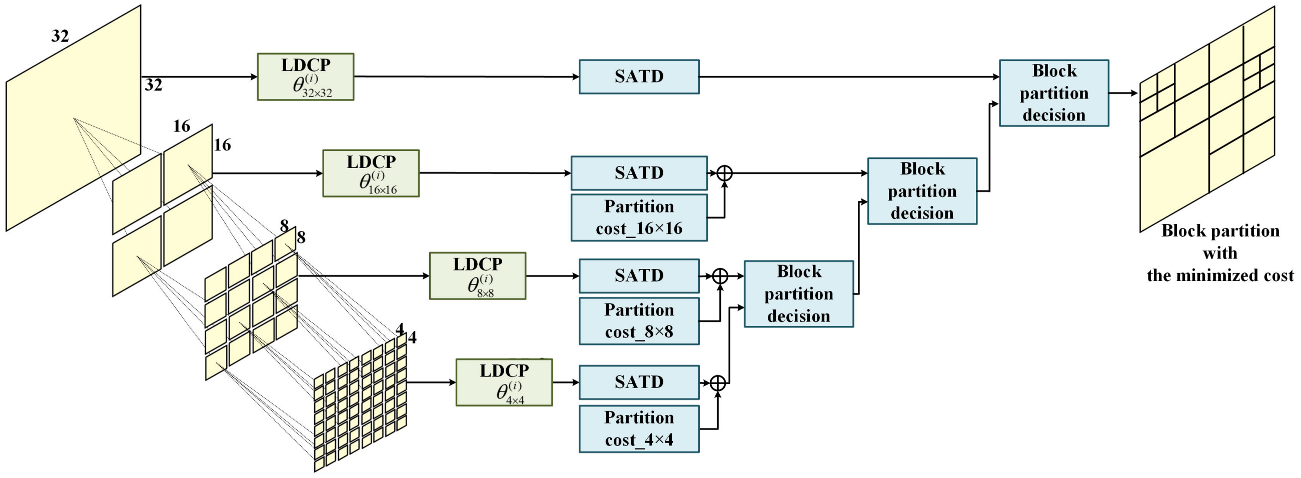

32
32
LDCP $\theta^{(i)}_{32\times32}$
SATD
Block partition decision
16
16
LDCP $\theta^{(i)}_{16\times16}$
SATD
Partition cost_16×16
Block partition decision
8
8
LDCP $\theta^{(i)}_{8\times8}$
SATD
Partition cost_8×8
Block partition decision
4
4
LDCP $\theta^{(i)}_{4\times4}$
SATD
Partition cost_4×4
Block partition with the minimized cost

Fig. 10. Block partition simulation for 32 × 32 blocks. The block partition is determined based on the minimized cost.

3) Loss Calculation: The objective of L-LDCP is to minimize the loss between the predicted chroma in Fig. 11 and the ground-truth chroma.

In video coding, the prediction residual is usually transformed into the frequency domain for efficient quantization and entropy coding [41]. To simulate the coding loss in video coding, the proposed L-LDCP designs the loss criterion according to the prediction residual after discrete Cosine transform (DCT). Specifically, the L1 norm of the transform coefficients is used as

$$\begin{aligned} Loss &= \left\| \mathrm{DCT}(\mathbf{C}^{\mathrm{P}} - \mathbf{C}^{\mathrm{O}}) \right\|_1 \\ &= \left\| \mathrm{DCT}(\mathbf{w}^{\top}\mathbf{C}^{\mathrm{R}} - \mathbf{C}^{\mathrm{O}}) \right\|_1, \end{aligned} \tag{14}$$

where the attention model is derived by

$$\mathbf{w} = \mathrm{Softmax}(\Phi, \mathbf{LD}). \tag{15}$$

In the proposed L-LDCP, the open DIV2K dataset [42] was adopted as the training set. 800 images in DIV2K were cropped into blocks of 65 × 65. Specifically, a block of 65×65 in the YCbCr 4:2:0 format is composed of the reference samples, a 64×64 luma block, and two 32×32 chroma blocks. The luma component was downsampled to align with the size of the chroma components. The proposed L-LDCP is based on the PyTorch deep learning framework, and the experimental environment is the 64-bit Windows 10 operating system. For each $|\theta|$ in Φ, the AdamOptimizer and the learning rate are configured independently.

Finally, the values of $|\theta|$ for three block classes[3] are designed as

$$|\theta| = \begin{cases} 8 & min(W,H) \le 4 \\ 12 & 4 < min(W,H) \le 16. \\ 16 & min(W,H) > 16 \end{cases} \tag{16}$$

In L-LDCP, $|\theta|$s for different block sizes are stored at the encoder and decoder in advance. Compared with the proposed T-LDCP, the attention model of L-LDCP is derived with low complexity.

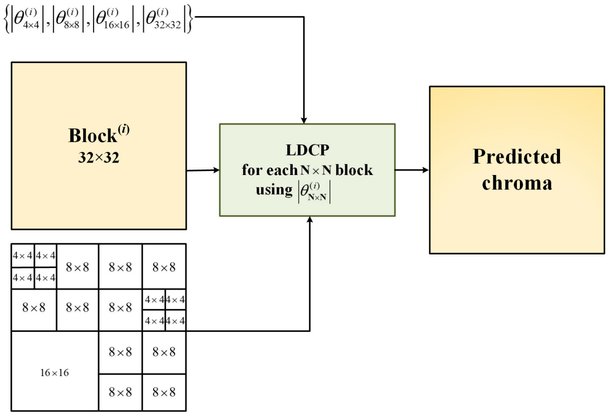

$\left\{ |\theta^{(i)}_{4\times4}|, |\theta^{(i)}_{8\times8}|, |\theta^{(i)}_{16\times16}|, |\theta^{(i)}_{32\times32}| \right\}$
Block$^{(i)}$
32×32
LDCP for each N×N block using $|\theta^{(i)}_{N\times N}|$
Predicted chroma
4×4
4×4
4×4
4×4
8×8
8×8
8×8
8×8
8×8
8×8
4×4
4×4
4×4
4×4
16×16
8×8
8×8
8×8
8×8

Fig. 11. The prediction construction of L-LDCP. Each block employs the proposed LDCP in which the model parameter is decided according to the block size.

D. Integration Into H.266/VVC

We integrate the proposed LDCP into H.266/VVC codec as a new chroma prediction mode to improve the coding performance. For L-LDCP, GPU is activated for model parameter training, and the encoding and decoding are employed based on the CPU.

On the encoder side, all the candidate modes are compared according to the RDO criteria [40], the mode with the smallest rate-distortion cost is selected to construct the predicted signal [43], and the index of the optimal mode is signaled in the bitstream. The optimal mode can be expressed as

$$M^{\mathrm{Opt}} = \underset{M_i \in \mathbf{M}}{\operatorname{argmin}}\, D(M_i) + \lambda R(M_i), \tag{17}$$

where M_i is the i-th candidate mode, $\mathbf{M}$ is the set of chroma prediction mode, $D(M_i)$ and $R(M_i)$ represent the coding distortion and coding bits overhead, respectively, λ is the Lagrangian factor which controls the rate-distortion trade-off.

On the decoder side, the optimal mode is parsed from the bitstream, and the corresponding predicted chroma is derived. By introducing a new coding mode, one additional binary flag is transmitted to indicate whether the coding block is coded as the proposed LDCP.

[3] The values of $|\theta|$ in training are float numbers. After training, we rounded the training results to the integers.

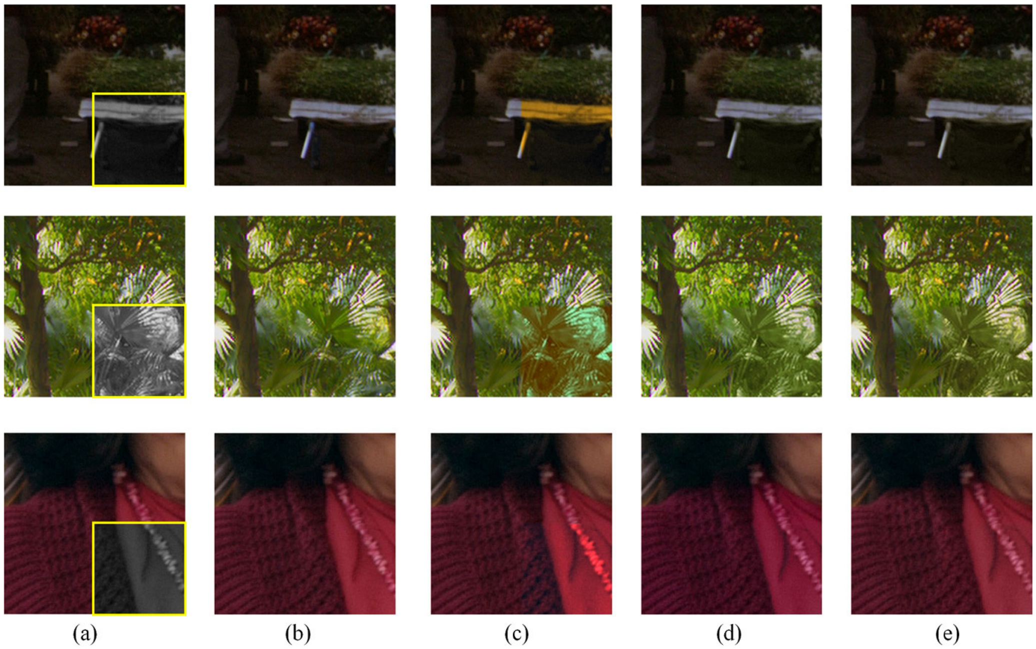


Fig. 12. Subjective comparison of the predicted chroma with different algorithms, the blocks located at the bottom-right, highlighted with yellow borders, are the blocks to be predicted. (a) Predicted block without chroma. (b) Predicted block with original chroma. (c) Predicted block by CCLM. (d) Predicted block by T-LDCP. (e) Predicted block by L-LDCP.

V. Experimental Results and Analyses

Extensive experiments were conducted to verify the performance of the proposed algorithm in terms of prediction accuracy, coding performance, and complexity analyses. According to the model parameter derivation, the proposed T-LDCP and L-LDCP are measured and analyzed, respectively.

A. Prediction Accuracy Comparison

In this subsection, an offline test was employed to split images into blocks and to derive the predicted chroma according to different prediction algorithms, including CCLM [7], T-LDCP, and L-LDCP. The peak signal-to-noise ratio (PSNR) of the predicted chroma with the original chroma was used to measure the prediction accuracy. 100 images in the verification set of DIV2K were tested. The corresponding PSNR results are listed in Table II.

Given a specific prediction algorithm, a higher PSNR can be obtained when images are split with smaller block sizes. With the increase of the block sizes, the prediction accuracy decreases. This phenomenon is because the correlation between reference samples and the predicted sample is higher for smaller blocks than that of larger blocks. Given specific block sizes, the proposed T-LDCP and L-LDCP provide higher PSNRs than CCLM, reaching 0.94-1.34dB improvement.

TABLE II

PSNR of the Prediction Algorithms With Different Block Sizes. (Unit: dB)

	4×4	8×8	16×16	32×32
CCLM	36.99	35.06	33.01	31.06
T-LDCP	37.93	36.23	34.34	32.40
L-LDCP	37.95	36.03	34.20	32.21

We also launched the offline chroma prediction for VVC-recommended sequences. The results of three test sequences [44], including $Marketplace$, $ParkRunning3$, and $Tango2$, are provided in Fig. 12 for subjective comparison. Here, the blocks at the bottom-right, highlighted with yellow borders, are the blocks to be predicted. In the first column, the predicted blocks are presented without chroma. Then, the blocks of the following columns are filled with the original chroma, the prediction of CCLM, T-LDCP, and L-LDCP, respectively. For the blocks with complex textures, the results of CCLM are not accurate enough. The proposed T-LDCP and L-LDCP construct the predictions which are closer to the original chroma. We believe the high prediction accuracy is benefitted from the adaptive attention model. The attention model based on the softmax function provides a nonlinear feature where the reference samples with small LDs are assigned large weights, and samples with large LDs are assigned small weights. More importantly, the weight vector of reference samples for each predicted sample is decided by the LD vector, which is pixel-by-pixel information. This design is more flexible than a mono-model in CCLM.

B. Coding Performance Comparison With Existing Cross-Component Prediction Algorithms

To verify the coding performance, the latest reference software of H.266/VVC, VVC test model (VTM) version 18.0 [45] was used as the anchor. The experiments were performed according to the common test condition (CTC) specified in [44].

Eighteen test sequences were coded to provide an overall evaluation of the proposed algorithm. Specifically, Class A1, A2, B, and C are video sequences with resolutions of 3840 × 2160, 3840 × 2160, 1920 × 1080, and 832 × 480,

TABLE III

BD-Rate Reductions of MMLM, Attention CCP, CCCM, T-LDCP, L-LDCP, and CCCM+L-LDCP for Each Class in AI Configuration. (Unit: %)

	MMLM [21]			**Attention CCP** [30]			**CCCM** [23]			**T-LDCP**			**L-LDCP**			**CCCM+L-LDCP**		
Class	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**
A1	-0.42	-1.36	-3.37	-0.28	-3.24	-2.00	-1.36	-4.53	-7.42	-0.84	-2.30	-4.59	-0.72	-2.33	-4.05	-1.62	-5.97	-9.34
A2	-0.06	-0.25	-0.31	-0.26	-2.90	-1.52	-5.49	-6.48	-9.33	-0.22	-1.97	-1.70	-0.22	-1.98	-1.69	-5.59	-7.59	-10.22
B	-0.11	-0.95	-0.85	-0.26	-2.51	-2.40	-0.70	-5.69	-3.21	-0.26	-2.33	-2.36	-0.27	-2.40	-2.36	-0.75	-6.74	-4.51
C	-0.17	-0.78	-0.99	-0.28	-1.95	-1.55	-0.27	-1.81	-1.60	-0.34	-1.86	-1.94	-0.35	-1.83	-1.70	-0.45	-3.09	-2.91
E	0.00	-0.12	-0.01	-0.14	-1.73	-1.35	-0.29	-3.01	-1.08	-0.07	-1.47	-1.24	-0.06	-1.60	-1.34	-0.32	-4.12	-2.38
Avg	**-0.15**	**-0.72**	**-1.07**	**-0.25**	**-2.44**	**-1.82**	**-1.45**	**-4.32**	**-4.22**	**-0.34**	**-2.02**	**-2.34**	**-0.32**	**-2.06**	**-2.21**	**-1.56**	**-5.50**	**-5.56**
YCbCr		**-0.25**			**-0.51**			**-1.87**			**-0.57**			**-0.54**			**-2.13**	

TABLE IV

BD-Rate Reductions of L-LDCP for Each Sequence in AI and RA Configurations. (Unit: %)

		All Intra			**Random Access**		
	Test Sequences	**Y**	**Cb**	**Cr**	**Y**	**Cb**	**Cr**
A1	*Tango2*	-0.73	-5.67	-5.84	-0.19	-1.15	-0.85
	FoodMarket4	-0.33	-1.75	-2.71	-0.16	-1.00	-0.87
	Campfire	-1.11	0.44	-3.59	-0.51	-0.24	-1.56
A2	*CatRobot*	-0.47	-2.51	-2.86	-0.16	-1.32	-1.04
	DaylightRoad2	-0.13	-3.02	-1.86	-0.04	-1.01	-0.63
	ParkRunning3	-0.06	-0.40	-0.35	-0.04	-0.10	0.04
B	*MarketPlace*	-0.58	-3.28	-1.76	-0.14	-2.09	-1.69
	RitualDance	-0.35	-2.82	-4.29	-0.15	-1.14	-2.17
	Cactus	-0.14	-1.34	-1.58	0.01	-1.04	-1.64
	BasketballDrive	-0.19	-2.48	-1.99	0.04	-0.91	-0.76
	BQTerrace	-0.07	-2.09	-2.18	-0.05	-1.82	-1.64
C	*BasketballDrill*	-0.86	-3.74	-3.13	-0.17	-0.75	-1.06
	BQMall	-0.24	-1.78	-1.77	-0.09	0.26	-0.45
	PartyScene	-0.11	-1.05	-0.97	-0.07	-1.09	-0.56
	RaceHorses	-0.19	-0.73	-0.92	0.03	-0.28	0.12
E	*FourPeople*	-0.05	-1.30	-0.96	-	-	-
	Johnny	-0.06	-1.68	-1.30	-	-	-
	KristenAndSara	-0.08	-1.83	-1.77	-	-	-
	Average	**-0.32**	**-2.06**	**-2.21**	**-0.11**	**-0.91**	**-0.98**

respectively. Class E is a set of typical video conference sequences with a resolution of 1280 × 720. The Bjϕntegaard-delta bitrate (BD-rate) [46] was used to evaluate the objective rate-distortion performance. A negative BD-rate value indicates a bitrate reduction compared with the anchor at the same reconstructed quality. Four QPs, i.e., 22, 27, 32, and 37, were used to obtain four rate points.

For comparison purposes, we integrated MMLM [21], Attention CCP [30], CCCM [23], T-LDCP, and L-LDCP on top of VTM 18.0. For MMLM [21] and CCCM [23], which are advanced chroma predictions in the state-of-the-art Enhanced Compression Model (ECM) [47], we employed the corresponding implementation developed by JVET and integrated it into VTM 18.0. For the attention CCP [30], which is an open source, we retrained the model, made sure the retrained performance was consistent with what reported in [30], and implemented the algorithm on VTM 18.0. The coding results of MMLM, Attention CCP, CCCM, T-LDCP, L-LDCP, and CCCM+L-LDCP are listed in Table III.

Compared with VTM 18.0, BD-rate reductions of 0.15%, 0.72%, and 1.07% can be achieved by MMLM for the Y, Cb, and Cr components, respectively. A higher coding improvement, i.e., 0.25%, 2.44%, and 1.82% BD-rate savings, can be provided by the Attention CCP. It is observed that CCCM brings 1.45%, 4.32%, and 4.22% bitrate savings for the Y, Cb, and Cr components over VTM 18.0. Apparently, CCCM is an advanced algorithm, and its coding gain over VTM is from efficient filtering using the co-located, top, bottom, left, and right luma samples. For T-LDCP, the BD-rate savings are 0.34%, 2.02%, and 2.34% for the Y, Cb, and Cr components, respectively. L-LDCP achieves 0.32%, 2.06%, and 2.21% bitrate savings for the Y, Cb, and Cr components, respectively. For these CCP algorithms, substantial BD-rate savings can be achieved for Cb and Cr components. We further integrated a combination scheme of CCCM and L-LDCP. A higher coding gain can be achieved, i.e., 1.56%, 5.50%, and 5.56% bitrate savings for Y, Cb, and Cr, respectively. That is, compared with CCCM alone, L-LDCP brings 0.11%, 1.18%, and 1.34% additional bitrate savings for the Y, Cb, and Cr components, respectively.

For better comparison, a YCbCr PSNR-based BD-rate [48] was also reported. As shown in Table III, 0.25%, 0.51%, 1.87%, 0.57%, 0.54%, and 2.13% BD-rate savings can be achieved by MMLM, Attention CCP, CCCM, T-LDCP, L-LDCP and CCCM+L-LDCP, respectively.

From Table VII, we can see that both T-LDCP and L-LDCP bring a higher coding improvement than the two state-of-the-art algorithms. Compared with L-LDCP, T-LDCP provides a slightly higher coding efficiency by looping the candidate $|\theta|$s and determining $|\theta|^{\text{Opt}}$ based on the template. However, since $|\theta|^{\text{Opt}}$ determination is needed for both the encoder and decoder, a high complexity is required for T-LDCP. In L-LDCP, $|\theta|$s of three classes are stored in both the encoder and decoder. In this way, the attention model can be derived with extremely low complexity. In terms of coding performance and complexity, L-LDCP is used in the following experiments.

Table IV further provides the detailed BD-rate results of L-LDCP under AI and random access (RA) configurations. For AI configuration, the coding improvement of L-LDCP has a good consistency for test sequences. Here, L-LDCP delivers significant coding gains for sequences in Class A1 with a resolution of 4K and rich textures. For Class E sequences dominated by flat backgrounds, the coding gain originating

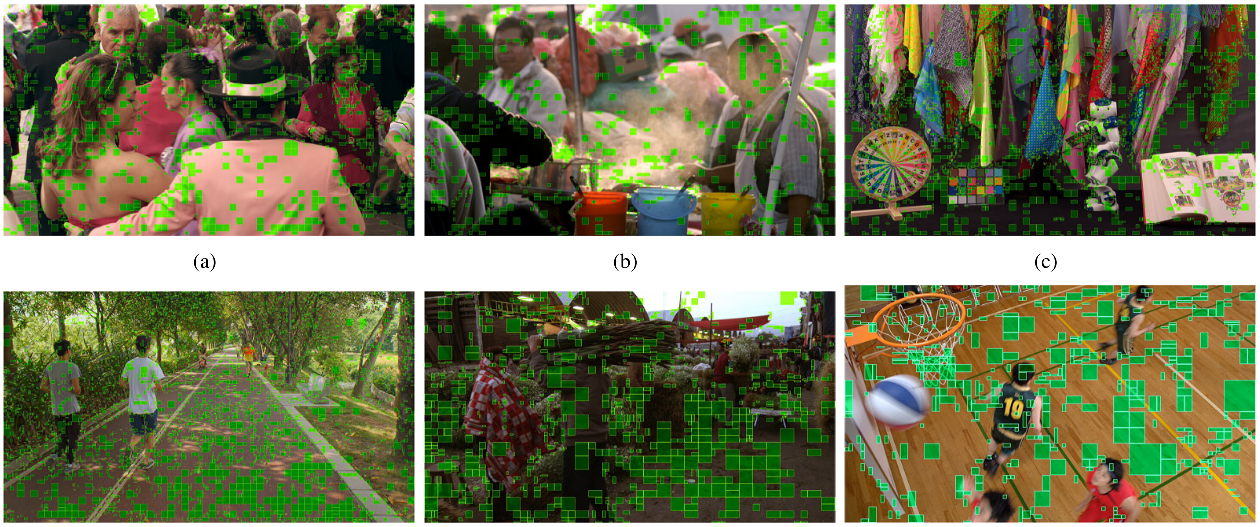


Fig. 13. Visualization of blocks coded with LDCP, highlighted with green color. (a) *Tango*. (b) *FoodMarket4*. (c) *CatRobot*. (d) *ParkRunning3*. (e) *Marketplace*. (f) *BasketballDrill*.

TABLE V
PERCENTAGES OF LDCP BLOCKS UNDER DIFFERENT QPS. (UNIT: %)

Class	Sequences	QP			
		22	**27**	**32**	**37**
A1	*Tango2*	20.96	24.68	29.18	30.78
	FoodMarket4	14.70	18.25	22.38	21.57
	Campfire	10.87	15.73	18.44	21.38
A2	*CatRobot*	19.28	17.90	19.71	21.97
	DaylightRoad2	14.00	12.07	13.02	12.57
	ParkRunning3	21.30	16.24	16.65	15.67
B	*MarketPlace*	25.85	29.32	31.07	31.86
	RitualDance	18.15	23.84	26.65	25.28
	Cactus	14.25	15.13	15.75	17.93
	BasketballDrive	10.13	10.96	13.05	12.87
	BQTerrace	15.16	17.36	17.04	15.97
C	*BasketballDrill*	21.97	28.03	36.94	37.74
	BQMall	13.91	14.98	15.64	14.67
	PartyScene	16.01	17.90	18.05	17.99
	RaceHorses	9.73	14.27	17.46	20.52
E	*FourPeople*	6.02	8.83	10.82	14.29
	Johnny	7.53	8.79	7.00	8.69
	KristenAndSara	9.06	9.64	9.86	11.96
Average		**14.94**	**16.88**	**18.82**	**19.65**

from L-LDCP is limited. For RA configuration, 0.11%, 0.91%, and 0.98% BD-rate reductions can be presented for Y, Cb, and Cr components, respectively. The performance of Class E was not evaluated in RA configuration [44] due to the tiny motion. It is observed that the coding efficiency can be improved by the proposed L-LDCP algorithm, especially for the Cb and Cr components.

C. LDCP Blocks Statistics

To further analyze the contribution of LDCP, the blocks coded with LDCP mode are counted. Table V provides the quantitative percentage of LDCP blocks for VVC test sequences. We can observe that 14.94%, 16.88%, 18.82%, and 19.65% pixels are coded with LDCP under four QP settings, respectively. The percentage of LDCP increases with the QP's increment.

Furthermore, blocks that choose LDCP as the optimal mode are vividly shown in Fig. 13, highlighted with green color. Specifically, LDCP blocks of the first frame of $Tango2$ (3840 × 2160), $FoodMarket4$ (3840 × 2160), $CatRobot$ (3840 × 2160), $ParkRunning3$ (3840 × 2160), $Maketplace$ (1920×1080), and $BasketballDrill$ (832×480) are provided. Here, QP is set to 22.

D. Coding Performance Comparison With Different Reference Regions

An experiment was conducted to use different numbers of reference samples to launch LDCP. In the experiment of Table III, one top line and one left column are used as the reference region of LDCP. This design is similar to the conventional intra prediction in H.266/VVC. In this experiment, we added two reference region options. Specifically, the reference region was set to a half line and a half column, denoted as LDCP-half, or set to 6 lines/columns, denoted as LDCP-6 lines.

The results of the LDCP-half and LDCP-6 lines are listed in Table VI. If the reference region is reduced to half, the YCbCr BD-rate saving is 0.44%. Compared with the performance of L-LDCP in Table III, less BD-rate saving is presented for LDCP-half. When we extend the reference region to 6 lines and 6 columns, a larger coding gain, i.e., 0.72% bitrate saving, can be achieved. The gain of LDCP turns larger using more reference lines, especially for Class A1 and A2 sequences. The reason is that a larger reference region will help these high-resolution sequences to find reference samples with higher correlation.

E. Coding Performance Comparison With Neural Network-Based Video Coding

To evaluate the performance of LDCP on top of more advanced codecs after H.266/VVC, we implemented the

TABLE VI

BD-Rate Reductions of LDCP-Half and LDCP-6 Lines Based on VTM 18.0 for Each Class in AI Configuration. (Unit: %)

	LDCP-Half				LDCP-6 lines			
Class	**Y**	**Cb**	**Cr**	**YCbCr**	**Y**	**Cb**	**Cr**	**YCbCr**
A1	-0.57	-2.08	-3.44	**-0.79**	-0.93	-3.42	-5.79	**-1.39**
A2	-0.16	-1.73	-1.40	**-0.36**	-0.32	-2.73	-2.11	**-0.64**
B	-0.21	-2.02	-2.00	**-0.42**	-0.34	-3.26	-3.19	**-0.69**
C	-0.27	-1.40	-1.44	**-0.43**	-0.39	-2.24	-2.44	**-0.66**
E	-0.05	-1.34	-1.14	**-0.20**	-0.08	-1.62	-1.54	**-0.27**
Avg	**-0.25**	**-1.73**	**-1.87**	**-0.44**	**-0.40**	**-2.70**	**-3.00**	**-0.72**

TABLE VII

BD-Rate Reductions of LDCP Based on NNVC4.0 for Each Class in AI Configuration. (Unit: %)

NNVC4.0+set0+intra+LDCP vs. NNVC4.0+set0+intra				
Class	**Y**	**Cb**	**Cr**	**YCbCr**
A1	-0.91	-0.23	-2.13	**-0.92**
A2	-0.19	-0.18	-0.52	**-0.21**
B	-0.23	-0.73	-0.41	**-0.29**
C	-0.29	-0.36	-0.24	**-0.29**
E	-0.06	-0.12	-0.21	**-0.08**
Avg	**-0.32**	**-0.37**	**-0.64**	**-0.35**

proposed LDCP on the latest NN-based video coding (NNVC) with the test model version 4.0. Several NN-based coding algorithms have been adopted in NNVC 4.0 [49], including two sets of NN-based in-loop filters [50], [51], NN-based intra prediction [52], super-resolution [53], and post-filters [54]. Here, we turned on the NN-based in-loop filter and NN-based intra prediction to get a high coding performance, and used this configuration as the anchor, denoted as NNVC4.0+set0+intra. We further integrated the proposed LDCP on top of the anchor, denoted as NNVC4.0+set0+intra+LDCP.

The coding result for each test class is listed in Table VI. Compared with the anchor, the proposed LDCP achieves 0.32%, 0.37%, and 0.64% bitrate savings for the Y, Cb, and Cr components, respectively. Compared with the coding performance on VTM 18.0 (i.e., 0.32%, 2.06%, and 2.21% bitrate savings), the coding gain in the luma component is maintained. The reason for that is the advanced coding algorithms in NNVC focus on efficient luma intra prediction and advanced in-loop filter, while LDCP is dedicated to improving the accuracy in chroma prediction and reducing the bitrate of chroma prediction residual. The BD-rate gain in the luma component comes from the decreased total bitrate. The coding gain of LDCP over NNVC in the chroma component is smaller than that over VTM 18.0, partially because the quality of chroma has been improved owing to the NN-based in-loop filter in NNVC. Nevertheless, LDCP brings further coding gains for NNVC in all the luma and chroma components.

F. Computational Complexity Analyses

Fig. 14 provides BD-rate savings and decoding time ratios of each algorithm for the overall evaluation. As a refined MHA algorithm, the encoding and decoding times of MMLM [21] are similar to that of the anchor, i.e., 100% and 100%. For the CCCM, 101% and 101% encoding and decoding times are presented. Compared with the anchor, the encoding and decoding times of T-LDCP is 120% and 580%, respectively. For the proposed L-LDCP, 102% and 101% of the encoding and decoding times are introduced, which is slightly higher than the anchor. The best coding performance in the MHA category is CCCM+L-LDCP, its encoding and decoding time is 102% and 101%, respectively. For the NN-based algorithms, the encoding and decoding time are higher than the MHA algorithms. It was reported that the attention CCP in [28] takes 212% and 2163% encoder and decoder times, while a simplified attention CCP in [30] reduce the encoding and decoding times to 164% and 1302%, respectively. For CNNCP [27], a higher BD-rate saving, i.e., 3.57%, can be achieved with an extremely large memory requirement, an encoding time of 116%, and a decoding time of 934%.

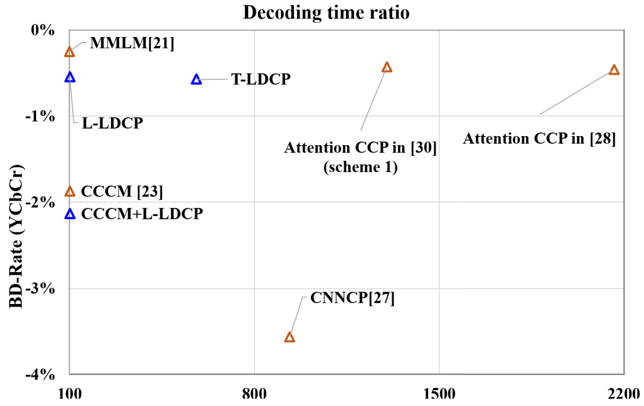


Fig. 14. BD-rate and decoding time comparison of CCP algorithms compared with VTM 18.0.

As illustrated in Fig. 14, the coding efficiency of the proposed LDCP is better than MMLM [21] and the attention algorithms [28], [30]. A combination of CCCM+LDCP can provide additional coding gain beyond CCCM. Apart from the good coding performance, the proposed algorithm has another merit of a low complexity. Here we employ the number of floating-point operations (FLOPs) as a metric to analyze the complexity. In [30], the FLOPs in the training stage are reported as 102859 and the FLOPs in the inference stage are 13770. In the proposed L-LDCP, according to (9) and (10), for each predicted pixel with the reference vector of K elements, the predicted chroma can be derived by 3K-3 additions and 2K+1 multiplications. For a 4 × 4 block, K is equal to 9. Apparently, the FLOPs of the proposed method are orders of magnitude smaller than those in [30]. Moreover, the proposed LDCP is independent of pixels in the coding block, which has the merit of pixel-level parallelization.

VI. Conclusion

We propose a chroma prediction algorithm based on a low-complexity attention model. The attention model takes the LDs of reference samples and the predicted sample as the input and constructs a nonlinear mapping from the LDs to the chroma weights. The attention model, designed as a softmax function, adaptively maps the weight according to the model parameter. Two model parameter determination methods, i.e., T-LDCP and L-LDCP, are proposed based on template derivation or offline learning. The T-LDCP achieves 0.34%, 2.02%, and 2.34% BD-rate savings for the Y, Cb, and Cr components, and 0.32%, 2.06%, and 2.21% BD-rate savings can be achieved by L-LDCP with low complexity. In future work, we will further investigate nonlinear prediction modeling to exploit video redundancies and improve coding efficiency.

Acknowledgment

The authors would like to thank Saiping Zhang, Hongqing Du, Zhenyao Zhang, and Wenzhong Liu for their valuable suggestions.

References

[1] Z. Wang et al., "Multi-memory convolutional neural network for video super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2530–2544, May 2019.

[2] *Advanced Video Coding for Generic Audiovisual Services*, Standard Rec. ITU-T H.264, ISO/IEC Standard 14496-10 AVC, May 2003.

[3] *High Efficiency Video Coding*, Standard Rec. ITU-T H.265, Version 1, ISO/IEC Standard 23008-2, Jan. 2013.

[4] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2225–2238, Jul. 2015.

[5] B. Bross et al., "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.

[6] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)," *Proc. IEEE*, vol. 109, no. 9, pp. 1463–1493, Sep. 2021.

[7] *Versatile Video Coding*, Standard Rec. ITU-T H.266, ISO/IEC Standard 23090-3 VVC, Aug. 2020.

[8] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma, and W. Gao, "Recent development of AVS video coding standard: AVS3," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, Nov. 2019, pp. 1–5.

[9] J. Han et al., "A technical overview of AV1," *Proc. IEEE*, vol. 109, no. 9, pp. 1435–1462, Sep. 2021.

[10] X. Meng, C. Jia, X. Zhang, S. Wang, and S. Ma, "Spatio-temporal correlation guided geometric partitioning for versatile video coding," *IEEE Trans. Image Process.*, vol. 31, pp. 30–42, 2022.

[11] M. Wang et al., "Low complexity trellis-coded quantization in versatile video coding," *IEEE Trans. Image Process.*, vol. 30, pp. 2378–2393, 2021.

[12] M. Lei, F. Luo, X. Zhang, S. Wang, and S. Ma, "Joint local and nonlocal progressive prediction for versatile video coding," *IEEE Trans. Image Process.*, vol. 31, pp. 2824–2838, 2022.

[13] J. Pfaff et al., "Intra prediction and mode coding in VVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3834–3847, Oct. 2021.

[14] D. Jin, J. Lei, B. Peng, W. Li, N. Ling, and Q. Huang, "Deep affine motion compensation network for inter prediction in VVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3923–3933, Jun. 2022.

[15] G. Laroche, J. Taquet, C. Gisquet, and P. Onno, *CE3: Cross-Component Linear Model Simplification (Test 5.1)*, document JVET-L0191, Oct. 2018.

[16] J. Li et al., "Sub-sampled cross-component prediction for emerging video coding standards," *IEEE Trans. Image Process.*, vol. 30, pp. 7305–7316, 2021.

[17] L. Trudeau, N. Egge, and D. Barr, "Predicting chroma from Luma in AV1," in *Proc. Data Compress. Conf.*, Mar. 2018, pp. 374–382.

[18] W.-S. Kim, D.-S. Cho, and H. M. Kim, "Inter-plane prediction for RGB video coding," in *Proc. Int. Conf. Image Process. (ICIP)*, 2004, pp. 785–788.

[19] W.-S. Kim et al., "Cross-component prediction in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1699–1708, Jun. 2020.

[20] J. Huo et al., "Unified cross-component linear model in VVC based on a subset of neighboring samples," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8654–8663, Dec. 2022.

[21] K. Zhang, J. Chen, L. Zhang, X. Li, and M. Karczewicz, "Enhanced cross-component linear model for chroma intra-prediction in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3983–3997, Aug. 2018.

[22] J. Lainema, A. Aminlou, P. Astola, and R. Youvalari, *AHG12: Slope Adjustment for CCLM*, document JVET-Y0055, Jan. 2022.

[23] P. Astola et al., *AHG12: Convolutional Cross-Component Model (CCCM) for Intra Prediction*, document JVET-Z0064, Apr. 2022.

[24] J. Huo, M. Zhang, W. Qiao, F. Yang, H. Su, and D. Mukherjee, "Improved chroma from Luma prediction in AV1 based on virtual chroma block generation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shenzhen, China, Jul. 2021, pp. 1–6.

[25] Y. Li et al., "A hybrid neural network for chroma intra prediction," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1797–1801.

[26] M. Meyer, J. Wiesner, J. Schneider, and C. Rohlfing, "Convolutional neural networks for video intra prediction using cross-component adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1607–1611.

[27] L. Zhu, Y. Zhang, S. Wang, S. Kwong, X. Jin, and Y. Qiao, "Deep learning-based chroma prediction for intra versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3168–3181, Aug. 2021.

[28] M. G. Blanch, S. Blasi, A. Smeaton, N. E. O'Connor, and M. Mrak, "Chroma intra prediction with attention-based CNN architectures," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 783–787.

[29] C. Zou, S. Wan, T. Ji, M. Mrak, M. G. Blanch, and L. Herranz, "Spatial information refinement for chroma intra prediction in video coding," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2021, pp. 1422–1427.

[30] M. G. Blanch, S. Blasi, A. F. Smeaton, N. E. O'Connor, and M. Mrak, "Attention-based neural networks for chroma intra prediction in video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 366–377, Feb. 2021.

[31] Y. Li, Y. Yi, D. Liu, L. Li, Z. Li, and H. Li, "Neural-network-based cross-channel intra prediction," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 3, pp. 1–23, Aug. 2021.

[32] Z. Huang, J. Sun, X. Guo, and M. Shang, "Adaptive deep reinforcement learning-based in-loop filter for VVC," *IEEE Trans. Image Process.*, vol. 30, pp. 5439–5451, 2021.

[33] C. Liu, H. Sun, J. Katto, X. Zeng, and Y. Fan, "QA-Filter: A QP-adaptive convolutional neural network filter for video coding," *IEEE Trans. Image Process.*, vol. 31, pp. 3032–3045, 2022.

[34] J. Lin, D. Liu, H. Li, and F. Wu, "M-LVC: Multiple frames prediction for learned video compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3543–3551.

[35] M. Singh, S. Nagpal, R. Singh, and M. Vatsa, "DeriveNet for (very) low resolution image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6569–6577, Oct. 2022.

[36] Y.-W. Huang et al., "A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1311–1325, May 2020.

[37] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, "Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1668–1682, Jun. 2020.

[38] J. Pfaff et al., "Video compression using generalized binary partitioning, trellis coded quantization, perceptually optimized encoding, and advanced prediction and transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1281–1295, May 2020.

[39] M. Koo, M. Salehifar, J. Lim, and S.-H. Kim, "Low frequency non-separable transform (LFNST)," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, Nov. 2019, pp. 1–5.

[40] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[41] X. Zhao et al., "Transform coding in the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3878–3890, Oct. 2021.

[42] R. Timofte et al., "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1110–1121.

[43] T. Li, M. Xu, R. Tang, Y. Chen, and Q. Xing, "DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC," *IEEE Trans. Image Process.*, vol. 30, pp. 5377–5390, 2021.

[44] F. Bossen et al., *JVET Common Test Conditions and Software Reference Configurations for SDR Video*, document JVET-N1010, Mar. 2019.

[45] *VVC Software VTM-18.0*. Accessed: Sep. 19, 2022. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-18.0

[46] G. Bjϕntegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33, Apr. 2001.

[47] V. Seregin et al., *JVET AHG Report: ECM Software Development (AHG6)*, document JVET-AD0006, Apr. 2023.

[48] W. Chien et al., *JVET AHG Report: Tool Reporting Procedure and Testing (AHG13)*, document JVET-T0013, Oct. 2020.

[49] S. Eadie et al., *JVET AHG Report: NNVC Software Development (AHG14)*, document JVET-AD0014, Apr. 2023.

[50] Y. Li, K. Zhang, and L. Zhang, *EE1-1.7: Deep In-Loop Filter with Additional Input Information*, document JVET-AC0177, Jan. 2023.

[51] R. Chang, L. Wang, X. Xu, and S. Liu, *EE1-1.1: More Refinements on NN Based In-Loop Filter with a Single Model*, document JVET-AC0194, Jan. 2023.

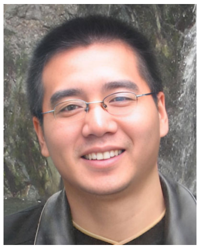
[52] T. Dumas, F. Galpin, and P. Bordes, *E1-3.2: Neural Network-Based Intra Prediction with Learned Mapping to VVC Intra Prediction Modes*, document JVET-AC0116, Jan. 2023.

[53] R. Chang et al., *EE1-2.2: GOP Level Adaptive Resampling with CNN-Based Super Resolution*, document JVET-AC0196, Jan. 2023.

[54] M. Santamaria et al., *EE1-1.11: Content-Adaptive Post-Filter*, document JVET-AC0055, Jan. 2023.

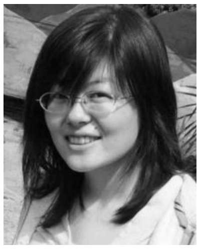
Junyan Huo (Member, IEEE) received the B.E. degree in telecommunication engineering and the M.E. and Ph.D. degrees in communication and information systems from Xidian University, Xi’an, China, in 2003, 2006, and 2008, respectively. From 2016 to 2017, she was a Visiting Scholar with the School of Computer Science and Engineering, Nanyang Technological University. She is currently an Associate Professor with Xidian University. Her current research interests include video compression, processing, and communication.

Danni Wang received the B.E. and M.E. degrees in communication engineering from Xidian University, Xi’an, China, in 2019 and 2022, respectively. Her current research interests include video coding and processing.

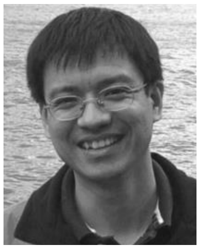


Hui Yuan (Senior Member, IEEE) received the B.E. and Ph.D. degrees in telecommunication engineering from Xidian University, Xi’an, China, in 2006 and 2011, respectively.

In April 2011, he joined Shandong University, Jinan, China, where he was a Lecturer from April 2011 to December 2014, an Associate Professor from January 2015 to August 2016, and a Professor in September 2016. From January 2013 to December 2014, he was a Postdoctoral Fellow (Granted by the Hong Kong Scholar Project) with the Department of Computer Science, City University of Hong Kong, where he was a Research Fellow from November 2017 to February 2018. From November 2020 to November 2021, he was a Marie Curie Fellow (Granted by the Marie Skłodowska-Curie Actions Individual Fellowship under Horizon2020 Europe) with the School of Engineering and Sustainable Development, De Montfort University, Leicester, U.K. From October 2021 to November 2021, he was a Visiting Researcher (secondment of the Marie Skłodowska-Curie Individual Fellowships) with the Computer Vision and Graphics Group, Fraunhofer Heinrich-Hertz-Institut (HHI), Germany. His current research interests include 3D visual media coding, processing, and communication. He is a member of the IEEE CTSoc Audio/Video Systems and Signal Processing Technical Committee (AVS TC), the IEEE CASSoc Visual Signal Processing and Communication Technical Committee (VSPC TC), and the APSIPA Image, Video, and Multimedia Technical Committee (IVM TC). He served as an Area Chair for IEEE ICME in 2023, ICME in 2022, ICME in 2021, IEEE ICME in 2020, and IEEE VCIP in 2020; and a Senior Area Chair for PRCV in 2023. He has been serving as an Associate Editor for *IET Image Processing* since 2023.



Shuai Wan (Member, IEEE) received the B.E. degree in telecommunication engineering and the M.E. degree in communication and information systems from Xidian University, Xi’an, China, in 2001 and 2004, respectively, and the Ph.D. degree in electronic engineering from the Queen Mary University of London in 2007. She is currently a Professor with Northwestern Polytechnical University, Xi’an. Her current research interests include scalable/multiview video coding, video quality assessment, and hyperspectral image compression.



Fuzheng Yang (Member, IEEE) received the B.E. degree in telecommunication engineering and the M.E. and Ph.D. degrees in communication and information systems from Xidian University, Xi’an, China, in 2000, 2003, and 2005, respectively. He was a Lecturer with Xidian University in 2005, where he was an Associate Professor in 2006. From 2006 to 2007, he was a Visiting Scholar and a Postdoctoral Researcher with the Department of Electronic Engineering, Queen Mary University of London. He has been a Professor of communications engineering with Xidian University since 2012. He is currently an Adjunct Professor with the School of Engineering, RMIT University. His current research interests include video quality assessment, video coding, and multimedia communication.