IEEE Open Journal of
Circuits and Systems

Received 29 June 2021; revised 30 September 2021; accepted 26 October 2021. Date of current version 22 November 2021.

Digital Object Identifier 10.1109/OJCAS.2021.3125730

Context-Adaptive Inverse Quantization for Inter-Frame Coding

KANG LIU (Graduate Student Member, IEEE), DONG LIU (Senior Member, IEEE), LI LI (Member, IEEE), AND HOUQIANG LI (Fellow, IEEE)

CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China

This article was recommended by Guest Editor M. Cagnazzo.

CORRESPONDING AUTHOR: D. LIU (e-mail: dongeliu@ustc.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 62036005, Grant 62022075, and Grant 62021001, and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2019-025.

ABSTRACT In the hybrid video coding framework, quantization is the key technique to achieve lossy compression. The information loss caused by the quantization may be reduced to improve compression efficiency, by using either encoder-side rate-distortion optimized quantization or decoder-side filtering. Nonetheless, the existing studies did not extensively use the already encoded information, i.e., context, to reduce the quantization loss. We address this issue and propose a context-adaptive inverse quantization method, namely, CAIQ. Specifically, for inter-frame coding, we analyze the correlation between the prediction signal (generated by motion compensated prediction) and the residual signal, as well as the correlation within the residual signal itself. We then present linear as well as nonlinear yet lightweight models to exploit the observed correlations in the frequency domain. Our models provide an optional inverse quantization mode by referring to the prediction signal, which is available at the decoder side. Next, block-level mode selection regarding the CAIQ method is used at the encoder side. We integrate the proposed CAIQ method into the reference software of Versatile Video Coding. We perform an extensive study of the models and analyze their resulting compression efficiency gain and encoding/decoding complexity. Experimental results show that our CAIQ method improves compression performance especially for high-resolution videos and at high bit rates.

INDEX TERMS Context-adaptive, correlation analysis, inter-frame coding, inverse quantization, transform coefficients.

I. INTRODUCTION

MODERN video coding standards, including H.264 [1], H.265 [2], and the latest H.266/VVC [3], all adopt a hybrid coding framework including prediction, transform, quantization, loop-filter, etc. In this framework, each frame is first divided into multiple blocks. Predictive coding refers to intra-frame or inter-frame prediction based on coded blocks to remove spatiotemporal correlation. Transform coding is followed closely, where the residual signal is transformed, quantized, and then coded into the bitstream in order. In the signal reconstruction process, the quantized level is scaled by the de-quantizer to achieve the frequency-domain coefficients, and then the inverse transform is conducted to reconstruct the residual signal in the spatial domain (with certain distortion). When all blocks of the current frame have been processed, in-loop filters will be applied to further reduce coding distortion.

In lossy coding, quantization is the root cause of coding distortion. It maps continuous signals into multiple discrete amplitudes, making the coefficients discrete, sparse, and easy to code. The remaining few representative coefficients are used for reconstruction by inverse quantization and inverse transform. Considering that transform and inverse transform do not introduce signal distortion, inverse quantization is the key to estimating coefficients and compensating for information loss.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Studies on inverse quantization mainly focus on transmitting the quantization step (or offset) to the decoder, thereby achieving context-adaptive coefficient reconstruction. However, these strategies are based on either a frame-level quantization matrix [4] or a global uniform quantization matrix [5], [6], making it infeasible to achieve flexible coefficient-level adaptation. Another strategy [7] is to design a signal-dependent inverse transform so that coefficient reconstruction takes into account the influence of quantization to reduce quantization distortion. Under this strategy, blocks with uniform size still need to share the same inverse transform kernels. To avoid the cost of transmitting parameters, some studies [8]–[10] have proposed adaptive quantization, which indirectly achieves the purpose of reducing quantization distortion by dynamically adjusting the quantization level at the encoder, but the decoder still follows a unified inverse quantization rule without having the ability of adaptive compensation. The in-loop filters [2] further reduce the quantization distortion by using the pixel correlations. However, the frame-level filter is not implemented in the rate-distortion optimization-based block partition, making it hard to achieve the optimal solution from the perspective of joint optimization.

In the video coding scheme, the block-based prediction mode and lossy reference area make it difficult to accurately describe the pixel-level motion of the signal. Hence, the prediction accuracy is limited, resulting in redundancy between the prediction signal and the residual signal. Furthermore, the discrete cosine transform (DCT) [11] is widely used in residual coding to remove the linear correlation between coefficients. However, DCT cannot ideally eliminate the linear correlations among pixels. Even worse, it cannot remove nonlinear correlations. Note that the prediction signal obtained by the predictive coding module does not help improve the efficiency of the residual coding module. Therefore, mining the correlation between signals still has great potential for improving compression efficiency.

Inspired by the above characteristics, in this paper, we propose a block-level context-adaptive inverse quantization (CAIQ) method and treat it as an optional inverse quantization mode. The already encoded information and the correlations among multiple signals are utilized as the context to improve the coding performance. We design linear and nonlinear models to establish a coefficient-level mapping to adaptively compensate for quantization distortion. Our specific contributions are summarized as follows.

- First, we identify the linear and nonlinear correlations between the prediction signal and the residual signal, especially in the frequency domain, and design linear and lightweight nonlinear models under the guidance of the frequency-domain correlations.
- Second, we combine the coefficient-level mapping model with the de-quantizer, and integrate it into the reference software of versatile video coding as a TU-level optional inverse quantization mode.
- Third, we explore the impact of context usage and model complexity on the performance, and analyze the impact of coding optimization on CAIQ.

The remainder of this article is organized as follows. Related work is presented in Section II. We identify the intercorrelations and intracorrelations in Section III. In Section IV, we introduce linear and nonlinear context-adaptive inv-quantization models and the whole framework. In Section V, we show the experimental settings and results about BD-rate performance and decoding complexity, followed by detailed analyses on the potential of CAIQ. Section VI concludes the paper.

II. RELATED WORK

In video coding, the information loss caused by quantization is often difficult to recover. In this section, we introduce the related work of adaptive quantization and quantization distortion compensation in detail.

A. ADAPTIVE QUANTIZATION

In mainstream codecs, scalar quantization is widely used due to its simplicity and ease of use. This is a typical hard-decision quantization method, that is, the correlation between coefficients is not considered. To improve the quantization efficiency, rate-distortion optimization quantization [2], [12] takes into account the context relationship among coefficients by jointly optimizing the bit rate and distortion. Dependent quantization [13], [14] maintains multiple alternative paths through state machines and dynamic programming and achieves more fine-grained quantization by switching between two quantizers. In addition, it is generally believed that transform coefficients with the same amplitude have different perceptual importance [15], [16]. Inspired by that, the quantization matrix [5], [17] can be utilized to adjust the coefficient-level quantization scaling in consideration of the sensitivity of the human eye to different frequency components.

To achieve more flexible content adaptation, in [4], Wedi and Wittmann encode the quantization offsets to the bitstream. HoangVan [18] studied the statistical relationship of the quantization parameter (QP) and rate-distortion performance and designed a fourth-order polynomial function to adaptively estimate frame-level QP. Yan *et al.* [19] utilized spatial and temporal characteristics to establish a spatiotemporal perception-aware model to adjust the CTU-level QP offset. In [10], [20]–[22], the information loss and compression artifacts introduced by quantization can be effectively reduced by accurately modeling the distribution of DCT coefficients. In [8], [9], [23], the temporal adaptive quantization methods were used to reduce the inter-frame dependency and achieve a significant global optimization by using a group of neighboring frames.

The adaptive quantization methods proposed above effectively reduce the quantization loss from the perspective of signal distribution and coefficient dependencies. However,

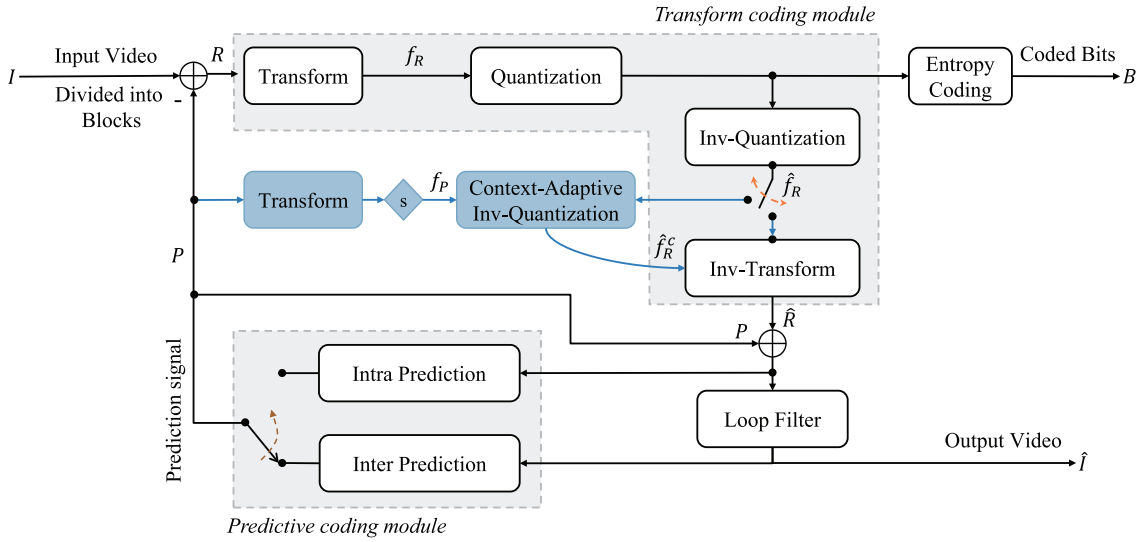


FIGURE 1. Illustration of the overall scheme integrated with the proposed context-adaptive inverse quantization method. The *P* and *R* represents the prediction signal and residual signal, respectively. "Inv-" is short for "Inverse."

once the quantization level is obtained, the inverse quantization process follows the fixed reconstruction relationship. In other words, adaptive reconstruction is not considered during inverse quantization.

B. QUANTIZATION DISTORTION COMPENSATION

Generally, transforms and inverse transforms are often designed in pairs without considering quantization distortion. Wang *et al.* [7] learn a new inverse transform to reduce the influence of quantization loss based on linear regression, where the corresponding inverse transform kernels can be transmitted offline to the decoder.

Chen *et al.* [24] pointed out that filtering the transform coefficients is a more direct way to compensate for the quantization loss, and it is helpful to consider the consistency with the human visual system. Studies in [25]–[27] show that it is feasible to use deep neural networks to process DCT-domain coefficients and may even accelerate convergence. Sun *et al.* [28] proposed a DCT-domain convolutional neural network in JPEG to learn the association between the reconstructed image and the original image, which effectively compensates for high-frequency information, thereby protecting the edge of the image.

Kim *et al.* [29] and Kang *et al.* [30] proposed that quantization distortion is not random and still has structural information. In [31], a second-order residual prediction technology is utilized based on vector quantization adapted to each intra-prediction mode. Yeh *et al.* studied the redundancy among multiple residual frames and proposed an inter-frame second-order residual prediction method in [32]. The residual motion vectors are transmitted to reduce the coefficients, thereby improving the coding efficiency. Besides, the impact of the prediction signal on residual reconstruction is explored in [33].

III. CORRELATION ANALYSIS

There is a large amount of spatiotemporally redundant information in videos. In the block-based video compression standard, intra-frame and inter-frame predictive coding are used to establish the relationship between the current block and the historical coded blocks, thereby effectively removing the intracorrelation and intercorrelation. As shown in Fig. 1, the predictive coding module represents the input signal I as a prediction signal P and a residual signal R. Note that block-based prediction has difficultly accurately measuring pixel-level motion. In addition, the quality of the reference frame further limits the prediction efficiency. Intuitively, there exists a correlation between the prediction signal P and the residual signal R.

In the transform coding module, the DCT transform is applied to R to remove the linear correlations, thereby improving the efficiency of entropy coding. The DCT-domain coefficients have removed certain linear correlations, which may reduce the difficulty of correlation analysis caused by inter-pixel dependence. Here, we select 40 sequences from the Consumer Digital Video Library (CDVL)[1] and obtain enormous blocks based on the original and reconstructed signals.

A. CORRELATION BETWEEN PREDICTION AND RESIDUAL

We consider two types of inter-signal correlations in the frequency domain. One is the correlation between the prediction signal f_P and the original prediction residual f_R. Another is the correlation between the reconstructed residual $\hat{f}_R$ and f_R.

Recording the number of blocks as N. Each block contains 64 (for 8×8 block) coefficients. Let $f_P^{(i,j)}$ represent the

1. Available at https://cdvl.org/.

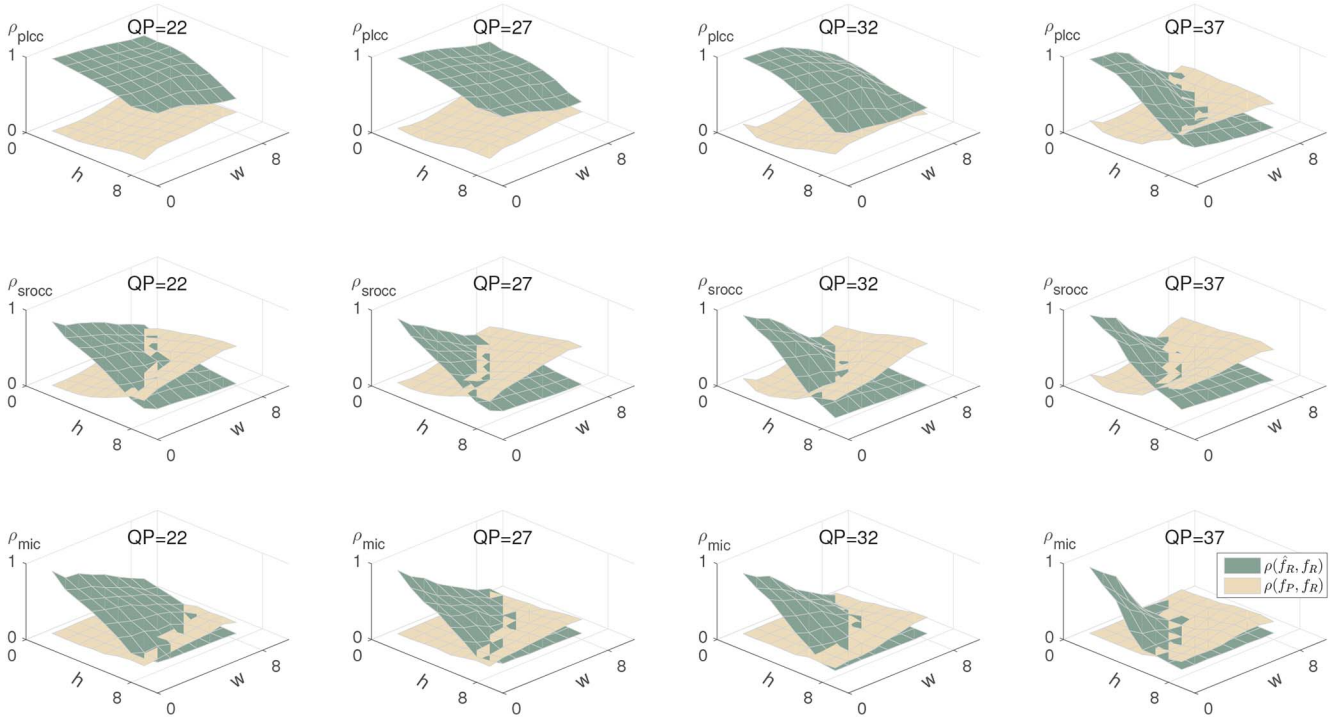


FIGURE 2. Visualization of correlations between signals. The three correlation metrics ρ_{plcc}, ρ_{srocc}, and ρ_{mic} are calculated based on each position. Furthermore, the correlation coefficients at all positions are fitted together to form a continuous plane. $\rho(f_P, f_R)$ describes the correlation between the prediction signal f_P and the original residual f_R (indicated by the yellow surface). $\rho(\hat{f}_R, f_R)$ describes the correlation between the reconstructed residual $\hat{f}_R$ and f_R (indicated by the green surface).

prediction coefficient at position (i, j). Based on similar representation rules, we can establish two kinds of data pairs, namely $\{f_P^{(i,j)}, f_R^{(i,j)}\}$ and $\{\hat{f}_R^{(i,j)}, f_R^{(i,j)}\}$.

Three correlation metrics are used to evaluate the correlation between signals. The first is the Pearson linear correlation coefficient (PLCC). Assume that X represents $f_P^{(i,j)}$ and Y represents $\hat{f}_R^{(i,j)}$. As shown in (1), where $cov(\cdot, \cdot)$ is the covariance function, σ_X and σ_Y are the standard deviations. ρ_{plcc} is widely used in the measurement of correlation, which reflects the linear relationship between variables and the direction of correlation. The sign of the coefficient indicates the correlation direction, and its range is $[-1, 1]$. For example, a coefficient of 1 indicates a strong positive correlation. When ignoring the direction, ρ_{plcc} can be transferred to $[0, 1]$.

$$\rho_{plcc} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{1}$$

In addition, we chose two other metrics for nonlinear measurement. The Spearman's rank-order correlation coefficient (SROCC) uses a monotonic equation to measure the correlation of two statistical variables. Compared with ρ_{plcc}, ρ_{srocc} classifies the variable data, thereby avoiding the influence of absolute values, while the same calculation strategy as ρ_{plcc} is adopted. The third metric is the maximum information coefficient (MIC) [34], which is based on the theory of mutual information and joint probability. ρ_{mic} quantifies the connection between two variables in a two-dimensional space. As shown in (2), n is the amount of data. $B(n)$ represents the number of divided grids and is set to $n^{0.6}$. I_G is denoted as the mutual information of the probability distribution included on the certain grid. $log(min(X, Y))$ is the normalization factor. As a result, MIC achieves a measure of dependence for two-variable relationships by maximizing mutual information.

$$\rho_{mic} = \max_{|X||Y|<B(n)} \frac{\max\{I_G\}}{\log(\min(X, Y))} \tag{2}$$

We define the correlation between f_R and $\hat{f}_R$ as $\rho(\hat{f}_R, f_R)$, and define the correlation between f_R and f_P as $\rho(f_P, f_R)$. Fig. 2 shows the correlation coefficient distribution characteristics of each location. The results show that the reconstructed residual has a strong linear and nonlinear correlation with the prediction residual in the low-frequency region, but the $\rho(\hat{f}_R, f_R)$ is extremely low in the high-frequency region. This phenomenon can be explained from the perspective of the quantizer; that is, a large number of high-frequency coefficients are quantized to zero and cannot be effectively recovered.

The correlation coefficients in $\rho(f_P, f_R)$ show an opposite trend. On the one hand, the linear correlation between the two is weak overall and is affected by the coefficient position and value of the quantization parameter (QP). As the coefficient position becomes closer to the high-frequency region or the QP increases, $\rho(f_P, f_R)$ gradually increases. On the other hand, from a nonlinear point of view, as the coefficient position transitions from the low-frequency region to the high-frequency region, $\rho(f_P, f_R)$

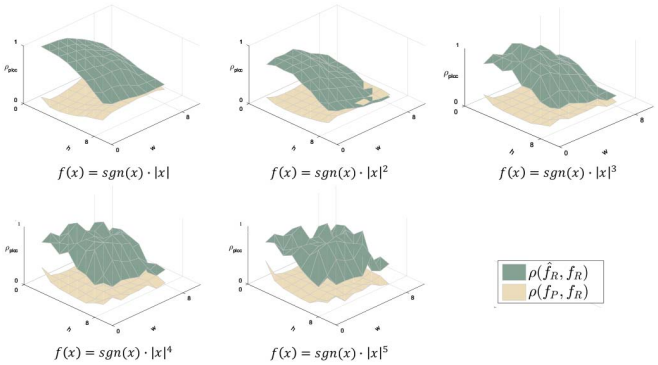


FIGURE 3. Influence of monotonic nonlinear mapping on correlation analysis. The coefficients of f_P and $\hat{f}_R$ at each position will be mapped by the function of $f(x)$.

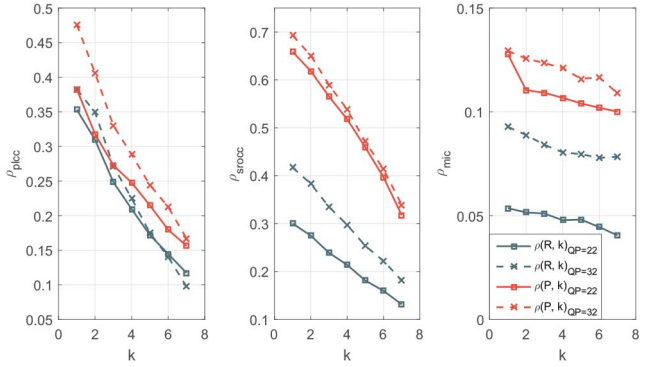


FIGURE 4. Intra-signal correlation with various distances, where k is defined as the pixel distance in the horizontal or vertical direction.

gradually increases and eventually surpasses $\rho(\hat{f}_R, f_R)$, which indicates that the prediction signal retains relatively rich high-frequency information and is suitable for the recovery of high-frequency residual coefficients. Note that the nonlinear correlation between signals is less affected by the QP, thus it shows a consistent changing trend under various bit rates.

To reduce the influence of the linear correlation when calculating the nonlinear metric, we further introduce multiple nonlinear mapping functions of different orders on f_P and $\hat{f}_R$. Figure 3 shows the changes in $\rho(f_P, f_R)$ and $\rho(\hat{f}_R, f_R)$ under different monotonic nonlinear mapping conditions. With an increasing nonlinear order, the gap between $\rho(f_P, f_R)$ and $\rho(\hat{f}_R, f_R)$ gradually increases, indicating that nonlinear mapping effectively reduces the linear correlation. However, the monotonic nonlinear mapping does not change the rank of the data, so the rank-based ρ_{srocc} does not change with the nonlinear order. Similarly, the impact of nonlinear mapping on ρ_{mic} is that the cells in the grid division process are scaled, but this scaling cannot affect the data distribution under different division structures, which can also maintain the unchanged correlation coefficient. From this perspective, we argue that this is strong proof of the nonlinear correlation among multiple signals.

B. CORRELATION WITHIN PREDICTION OR RESIDUAL

The Karhunen-Loeve Transform (KLT) is the theoretically optimal decorrelation transform, which can completely filter out the linear correlation of the signal. Considering the limitation of complexity, the DCT transform with a performance close to KLT is widely used in the mainstream compression framework. However, the decorrelation performance of DCT is directly related to the block size [6], meaning that its decorrelation ability is limited. In addition, it is difficult for DCT to effectively remove nonlinear correlations. In this section, we analyze the linear and nonlinear correlations within the signal. The DCT uses a fast algorithm based on parity decomposition. To avoid the sign difference caused by the parity position, the correlation analysis is based on the absolute value of the transform coefficients.

We define the distance in the horizontal or vertical direction as k. Therefore, the coefficient pair of f_P can be defined as a combination of $\{(f_P^{(i,j)}, \{(f_P^{(i+k,j)}\}$ and $\{(f_P^{(i,j)}, \{(f_P^{(i,j+k)}\}$. Similarly, the coefficient pair of f_R can be defined according to the same rules. Here we still take 8x8 blocks as an example to analyze intracorrelation, so the value of k is limited to no greater than 7.

The same three correlation metrics are utilized, and the results are fitted into a smooth curve, as shown in Fig. 4, where the solid and dashed lines correspond to the configurations of QP=22 and QP=32, respectively. First, the correlation of the three indicators within the signal shows the same changing trend as the distance k increases; that is, the greater the distance is, the lower the correlation. In addition, the intracorrelation of the prediction signal f_P is significantly higher than that of the residual signal f_R.

The correlation of ρ_{plcc} indicates that there is still a certain linear correlation between the coefficients of the transform domain, and the correlation decreases with increasing pixel distance. Under the linear index, the intracorrelation difference of different signals is small, and the discrimination of the same type of signals under different bit rates is also small. In contrast, the discrimination of the nonlinear correlations is improved, indicating that the nonlinearity may have a stronger potential to describe the correlation within the signal.

IV. METHOD

In residual coding, the transform concentrates the energy of the prediction residual to the low-frequency region as much as possible, which makes a large number of sparse high-frequency coefficients quantized to zero during quantization. Quantization defines a many-to-one mapping rule, while inverse quantization can only achieve one-to-one inverse mapping based on quantized levels. To overcome this problem, this paper proposes a context-adaptive inverse quantization method based on coefficient-level correlations. Furthermore, we combine CAIQ with the de-quantizer, thereby constructing a new optional inverse quantization mode.

A. METHOD OVERVIEW

Inspired by the analysis in Section III, we propose a coefficient mapping model $\Phi(\cdot)$ and combine it with the de-quantizer, and the detailed framework is shown in Fig. 1. The prediction signal is utilized to compensate for the reconstructed residual coefficients.

Assume that the de-quantized residual coefficients of the current transform unit (TU) are $\hat{f}_R$. Then, the distortion of the current block can be defined as:

$$D_{tu} = D(T^{-1}(\hat{f}_R) + P, R + P) \tag{3}$$

where T^{-1} corresponds to the inv-transform. P and R represent the prediction signal and original residual signal in the pixel domain, respectively. $D(\cdot,\cdot)$ represents the distortion caused by quantization. Note that the reconstructed block has been clipped to integers ranging from 0 to 255.

When the de-quantizer is combined with the coefficient mapping model, we build a new optional inverse quantization mode, that is, context-adaptive inv-quantization. First, an integer transform integrated with the baseline codec is utilized to transfer the prediction signal into transform coefficients. Considering that the integer transform brings a scaling effect that is bound to the block size, we introduce a scaling factor s to ensure that the transform coefficients of various block sizes have a consistent scaling scale. Therefore, we can achieve the scaled prediction coefficients f_P.

Second, as the key operation of the inverse quantization module, the coefficient mapping model fits the correlation characteristics among multiple signals, including $\hat{f}_R$, f_p, and f_R. For each residual coefficient, instead of being filtered directly by itself, the corresponding prediction coefficient can provide richer information. As a result, the compensated distortion is:

$$D^c_{tu} = D\Big(T^{-1}\Big(\Phi\Big(\hat{f}_R, f_P\Big)\Big) + P, R + P\Big) \tag{4}$$

Third, considering that the effect of the proposed inverse quantization method depends on the correlation between multiple signals, the limited expressive power of the mapping model and the diversity of the block content may lead to limited compensation effects and even side effects. Therefore, we adopt a rate-distortion optimization strategy to select the optimal mode between the context-adaptive inverse quantization mode and the original inverse quantization mode. A TU-level enable flag will be written into the bitstream. We have established a separate entropy coding model for the flag and designed the initialization parameters by statistically selecting the probability.

$$\hat{R} = \begin{cases} T^{-1}\Big(\Phi(\hat{f}_R, f_P)\Big), & D^c_{tu} < D_{tu} \\ T^{-1}\Big(\hat{f}_R\Big), & else \end{cases} \tag{5}$$

Furthermore, cosidering that our proposed CAIQ is based on the correlation between multiple signals, we believe that the correlation between the prediction signal and the residual signal is high when the efficiency of the predictive coding module is low so that the method has a higher potential. In the video coding framework, skip mode is utilized when the current block is very close to the reference area, and simultaneously the residual coding can be skipped. In addition, the code block flag (Cbf) is used to indicate whether its encoding result contains nonzero residuals. When skip mode is selected, or Cbf is equal to 0, the residual coefficients of the current block do not need to be coded; that is, the coefficients are all 0. In this case, from the perspective of computational complexity and the cost of the flag, the CAIQ mode can be skipped directly.

B. LINEAR REGRESSION MODEL

The linear correlation coefficients ρ_{plcc} shown in Fig. 2 indicate that there exists a clear linear relationship between the prediction signal and the residual signal, especially in the high-frequency region. Inspired by that, we first establish a coefficient-level mapping method based on a simple first-order linear model. Suppose that the frequency-domain prediction coefficient and reconstructed residual coefficient of the current block at position (i, j) are $f_{P(i,j)}$ and $\hat{f}_{R(i,j)}$, respectively, and the compensated residual coefficient $\hat{f}^c_{R(i,j)}$ is:

$$\begin{aligned} \hat{f}^c_{R(i,j)} &= \phi_{ij}\Big(\hat{f}_{R(i,j)}, f_{P(i,j)}\Big) \\ &= \alpha_{ij} \cdot f_{P(i,j)} + \beta_{i,j} \cdot \hat{f}_{R(i,j)} + \gamma_{ij} \end{aligned} \tag{6}$$

where ϕ_{ij} is the mapping model for achieving $\hat{f}^c_{R(i,j)}$. Note that α_{ij} and β_{ij} represent the weights and γ_{ij} represents the bias. Considering that $\hat{f}_{R(i,j)}$ is directly quantized by f_R, we set β_{ij} to 1 by default. As a result, the linear regression model is further simplified to:

$$\hat{f}^c_{R(i,j)} = \alpha_{ij} \cdot f_{P(i,j)} + \hat{f}_{R(i,j)} + \gamma_{ij} \tag{7}$$

Here, we optimize the parameters α_{ij} and γ_{ij} by minimizing the mean square error loss between $\hat{f}^c_{R(i,j)}$ and $f_{R(i,j)}$.

$$\arg\min_{\alpha_{ij},\gamma_{ij}} \epsilon^2_{ij} = \arg\min_{\alpha_{ij},\gamma_{ij}} \Big(f_{R(i,j)} - \hat{f}^c_{R(i,j)}\Big)^2 \tag{8}$$

We adopt a strategy of coefficient-level multi-round regression optimization. Specifically, for each candidate coefficient position, we first construct the input data pair $\{\hat{f}_{R(i,j)}, f_{P(i,j)}\}$ and the label data $f_{R(i,j)}$, and then conduct the least-squares algorithm according to (7). After finishing the first-round optimization, the dataset can be divided into a positive sample set and a negative sample set based on the latest linear model. Note that the positive samples refer to the samples whose compensation loss is within the threshold range. Furthermore, the next-round optimization is followed based on the positive sample set only. The above process was repeated, and the threshold was continuously adjusted. After the regression process is stable, we can achieve an optimal linear model. When all candidate coefficient positions have been modeled, the block-level compensated coefficients can be represented as:

$$\begin{aligned} \hat{f}^c_R = \Phi\Big(\hat{f}_R, f_P\Big) &= \mathbb{L}(f_P) + \hat{f}_R \\ &= A \cdot f_P + G + \hat{f}_R \end{aligned} \tag{9}$$

where A and G represent the weight matrix and the bias matrix, respectively, and the dimension is equal to $M \times N_{M,N\in\{4,8,16,32\}}$.

$$A = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1N} \\ \alpha_{21} & \cdots & \alpha_{2N} \\ \vdots & \ddots & \vdots \\ \alpha_{M1} & \cdots & \alpha_{MN} \end{bmatrix}, G = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1N} \\ \gamma_{21} & \cdots & \gamma_{2N} \\ \vdots & \ddots & \vdots \\ \gamma_{M1} & \cdots & \gamma_{MN} \end{bmatrix}. \quad (10)$$

C. NONLINEAR REGRESSION MODEL

Although the proposed linear model introduces the prediction signal into the residual inverse quantization process, it is difficult for the linear model to effectively capture the nonlinear relationship between the signals. Therefore, on the basis of the linear model, we make use of the convolutional neural network to further extend it into a nonlinear solution.

$$\hat{f}_R^c = \Phi(\hat{f}_R, f_P) = \mathbb{F}(f_P) + \hat{f}_R \quad (11)$$

As shown in (11), $\mathbb{F}$ represents the nonlinear model, and the specific network structure is shown in Fig. 5(a), where the structure of residual connection [35] is adopted. Different from simply using weights and biases to map f_P, here, we conduct multiple stacked convolution layers and rectified linear units (ReLU) to achieve nonlinear mapping from f_P to F_P. In addition, we infer a position-level-based mask matrix with the same dimension as F_P, namely, $Mask_P$, based on multi-layer feature fusion. When considering the use of richer nonlinear information, we can further introduce $\hat{f}_R$ into $\mathbb{F}(\cdot)$ to add a new branch that is dual to the branch of processing f_P, thereby constructing $\mathbb{F}(\hat{f}_R, f_P)$. The specific network structure is shown in Fig. 5(b).

$$\hat{f}_R^c = \Phi\left(\hat{f}_R, f_P\right) = \mathbb{F}\left(\hat{f}_R, f_P\right) + \hat{f}_R \quad (12)$$

The advantage of frequency-domain mapping lies in the fact that each frequency coefficient is associated with all the spatial-domain pixels, meaning that even when the kernel size is limited to 1×1, the receptive field of the network can still be considered as the entire block. In the network, the size of the convolution kernel and the number of layers together affect the complexity of the model. Note that k can be set as 1, 3, and 5, where the first configuration represents only the intercorrelation between signals is explored, while the latter two configurations further utilize the intra-signal correlation. To make full use of the extracted features and achieve the information interaction between the branches, all the extracted features are cascaded together, and the corresponding coefficient-level masks are available. The compensated coefficients can then be represented as a sum of masked branches and $\hat{f}_R$.

Different from the operation of optimizing linear models position by position, for nonlinear models, all coefficients share the same parameters, so we can update the Θ through a single-round regression optimization process.

$$\arg\min_{\Theta} \epsilon^2 = \arg\min_{\Theta} \mathcal{L}\left(f_R, \mathbb{F}(\hat{f}_R, f_P|\Theta) + \hat{f}_R\right) \quad (13)$$

where $\mathcal{L}(\cdot)$ represents the mean squared error loss.

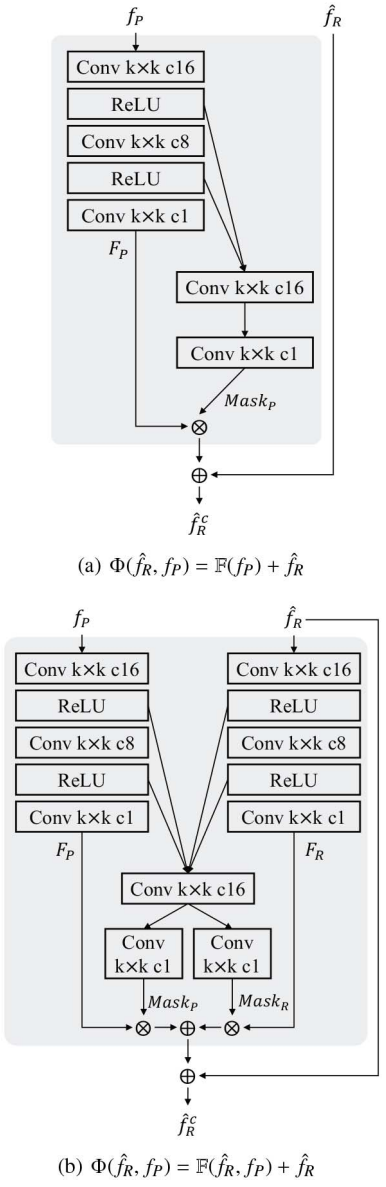


FIGURE 5. Structure of nonlinear convolutional neural network based on residual connection. "$k \times k$ $c16$" represents the kernel size and output feature channels of the current convolution layer. (a) Single-branch network. (b) Dual-branch network. (a) $\Phi(\hat{f}_R, f_P) = \mathbb{F}(f_P) + \hat{f}_R$ (b) $\Phi(\hat{f}_R, f_P) = \mathbb{F}(\hat{f}_R, f_P) + \hat{f}_R$.

V. EXPERIMENTS AND ANALYSES

A. EXPERIMENTAL SETTINGS

1) TEST CONFIGURATION

We integrate the proposed context-adaptive inverse quantization mode, including the linear models (implemented by C++) and nonlinear models (implemented by PyTorch), into VTM-1.0.[2] Compared with the high-efficiency video coding (HEVC) framework, VTM-1.0 expands the block division from the quadtree (QT) to the multi-type tree (MTT), resulting in various block sizes [36]. The proposed CAIQ mode is an optional mode based on TU blocks, therefore,

2. https://vcgit.hhi.fraunhofer.de/chujoh/VVCSoftware_BMS.

a TU-level enable flag needs to be written into the bitstream. Furthermore, we integrated the proposed method into VTM-6.0[3] and VTM-13.0[4] for research. Compared with VTM-1.0, the coding efficiencies of VTM-6.0 and VTM-13.0 are improved by more than 20% and 24% on average, respectively.

We set the QPs to {22, 27, 32, 37} as the common test condition and additionally test high bit rates, that is, we set the QPs to {17, 22, 27, 32}. Three coding configurations, including low delay P (LDP), low delay B (LDB), and random access (RA), are all tested. All sequences are utilized to evaluate the performance of our proposed method. The overall BD-rate performance is calculated based on only five classes consisting of classes A1, A2, B, C, and E. Note that all experiments and time complexity analyses are finished on the CPUs.

2) MODEL TRAINING

We randomly select 40 sequences from the CDVL as the training sequences. All sequences are first compressed with various QPs, including {17, 22, 27, 32, 37}, and the resulting reconstructed video has multiple frames (65 frames in total) with different qualities. Next, according to the QP of each frame, we cluster the frames ranging from $(QP-2)$ to $(QP+2)$, to produce the training sets under multiple QPs. Third, we take the TU as the basic unit and obtain blocks with various sizes based on the block partition results, where all blocks will be pre-transferred to the frequency domain by the DCT. Note that the dataset includes the prediction signal, the reconstructed residual, and the original residual (as the training label).

For the linear model, the position-level parameter regression strategy determines that we should train weight matrix A and bias matrix G for various block sizes. Considering that the proportion of blocks with sizes exceeding 32×32 is relatively low, we optimize parameters only for block sizes ranging from 4×4 to 32×32. All the training experiments are finished on the CPUs.

For the nonlinear model, the shareable parameters make it easy to perform joint optimization on all coefficients without considering the block sizes. Note that at high bit rates, the encoder tends to divide the image into smaller TUs, while at low bit rates, it is exactly relevant. Instead of training networks for each block size, we set 8×8 and 16×16 blocks as the training dataset when the QP is equal to {17, 22} and {27, 32, 37}, respectively. In addition, we train different models for multiple coding configurations. Considering that the gap between the input and the labels, under low bit rates, is large, we first train models under the high bit rates and then initialize the models under the low bit rates with pre-trained models. The Adam algorithm [37] is utilized for stochastic optimization. We set the learning rate (lr) to 0.01. All the training experiments are finished on the GPUs.

3. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-6.0.

4. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-13.0.

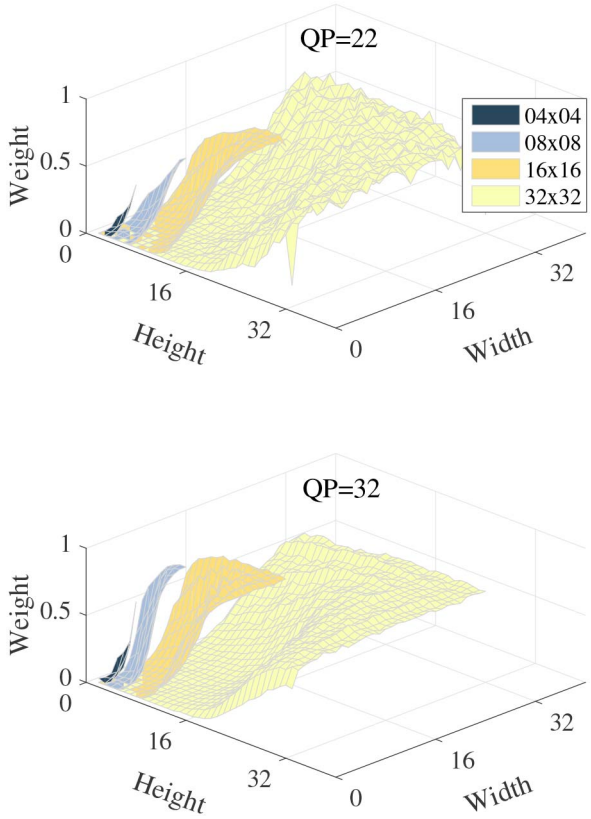


FIGURE 6. Visualization of weight matrices of the linear model for different block sizes.

B. EXPERIMENTAL RESULTS

1) PERFORMANCE OF LINEAR MODEL

Based on the function defined in (9), we trained a total of 16 sets of parameter matrices suitable for various block sizes (that is, block sizes ranging from 4×4 to 32×32). Fig. 6 shows the visualization results of the weight matrices $A_{QP=22}$ and $A_{QP=32}$, where the three-dimensional coordinates represent the width, height, and weight, respectively. Note that the weight of each coefficient in $\hat{f}_R$ is fixed to 1, so A plays a role in adjusting the weight of f_P. An obvious conclusion is that the weights in A are always limited in the range of [0, 1], most of which do not exceed 0.5. Since CAIQ is conducted in the frequency domain, the position near the coordinate (0, 0) indicates a lower frequency. In contrast, the coefficients far away from this position correspond to a higher frequency. The weights of f_P at the low frequency are close to 0, and the weights gradually increase with increasing frequency. This observation is in line with the intuitive impression and the results shown in Fig. 2, that is, the $\hat{f}_R$ at the low frequency can better reflect the energy distribution of the original residual, while the corresponding f_P may be useless. As the frequency increases, the correlation between the predicted signal and the residual signal becomes

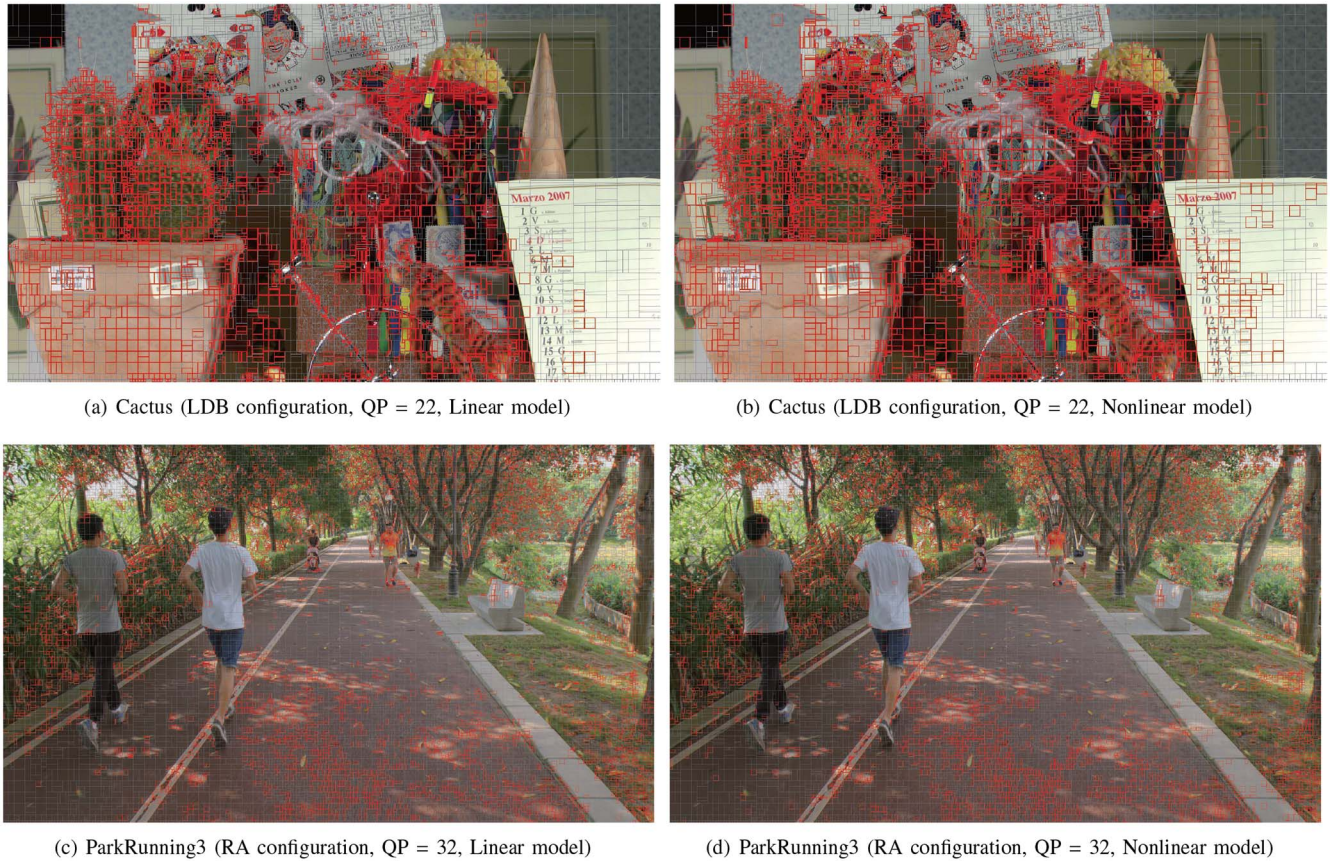
TABLE 1. BD-rate results of proposed linear model ($\hat{f}_R^C = \mathbb{L}(f_P) + \hat{f}_R$) on VTM-1.0.

QP	Class	Sequence	Low Delay P			Low Delay B			Random Access		
			Y (%)	U (%)	V (%)	Y (%)	U (%)	V (%)	Y (%)	U (%)	V (%)
{17, 22, 27, 32}	Class A1	Tango2	−1.81	−1.10	−0.31	−0.91	0.28	−0.68	−0.70	1.07	−1.07
		FoodMarket4	−1.46	0.70	1.05	−1.15	0.26	0.88	−0.86	−0.13	0.03
		CampfireParty2	−0.56	−0.79	−0.37	−0.21	−0.29	−0.22	−0.75	−0.71	−0.40
	Class A2	CatRobot1	−4.90	−1.56	−1.03	−1.65	−0.37	0.10	−2.14	−0.37	0.14
		DaylightRoad2	−1.73	−2.96	−1.88	−0.72	−0.92	−1.08	−1.13	−1.40	−0.66
		ParkRunning3	−4.94	0.30	0.78	−1.80	0.45	0.75	−0.94	0.43	0.65
	Class B	MarketPlace	−3.16	0.26	0.99	−1.01	0.67	0.43	−1.57	0.51	0.49
		RitualDance	−0.51	0.73	0.26	0.12	0.61	0.77	0.05	0.07	0.27
		Cactus	−3.10	−1.44	−1.62	−1.02	−0.47	−0.23	−1.40	−0.02	−0.30
		BasketballDrive	−1.95	−0.80	−0.36	−0.61	−0.40	−0.94	−0.89	0.83	−0.77
		BQTerrace	−1.03	−0.18	−0.14	−0.28	0.20	0.15	−0.44	−0.42	−0.01
	Class C	BasketballDrill	0.29	0.43	0.66	0.25	0.55	0.04	0.18	0.52	0.31
		BQMall	−0.11	−0.10	−0.33	0.44	0.89	0.63	0.20	0.55	0.48
		PartyScene	0.31	0.35	0.02	0.48	0.59	0.51	0.38	0.45	0.36
		RaceHorsesC	−0.62	0.21	−0.33	0.25	0.90	0.77	−0.09	−0.11	−0.32
	Class D	BasketballPass	0.28	0.88	0.99	0.45	1.47	1.28	0.09	0.56	0.03
		BQSquare	0.55	0.42	0.40	0.45	0.20	0.23	0.43	0.79	0.60
		BlowingBubbles	0.58	0.15	−0.56	0.54	0.49	0.19	0.43	0.69	−0.32
		RaceHorses	−0.03	0.67	0.92	0.43	−0.06	0.09	0.23	0.65	0.64
	Class E	FourPeople	−1.46	0.06	0.10	−0.49	0.37	−0.03	−0.61	0.60	0.23
		Johnny	−1.88	0.13	−0.16	−0.19	0.72	0.16	−0.49	0.75	0.25
		KristenAndSara	−1.17	−0.05	−0.64	−0.44	0.14	−0.10	−0.42	0.05	0.22
	Class F	BasketballDrillText	0.15	0.37	0.46	0.32	0.37	0.35	0.16	0.35	0.78
		ChinaSpeed	0.44	0.60	0.46	0.65	0.66	−0.03	0.61	−0.09	−0.18
		SlideEditing	−0.06	−0.15	−0.16	0.03	0.06	−0.03	0.22	0.12	0.04
		SlideShow	−0.09	−1.17	0.68	−1.28	−1.02	−1.87	0.47	−0.12	1.56
	Summary	Class A1	−1.28	−0.40	0.12	−0.76	0.08	−0.01	−0.77	0.08	−0.48
		Class A2	−3.86	−1.40	−0.71	−1.39	−0.28	−0.08	−1.40	−0.44	0.05
		Class B	−1.95	−0.29	−0.17	−0.56	0.12	0.04	−0.85	0.19	−0.06
		Class C	−0.03	0.22	−0.00	0.35	0.73	0.48	0.16	0.41	0.37
		Class E	−1.50	0.04	−0.23	−0.37	0.41	0.01	0.12	0.68	0.29
		Overall	**−1.66**	**−0.32**	**−0.18**	**−0.50**	**0.23**	**0.11**	**−0.65**	**0.16**	**0.03**
		Class D	0.35	0.53	0.44	0.47	0.53	0.45	0.29	0.67	0.24
		Class F	0.11	−0.09	0.36	−0.07	0.23	0.11	0.37	0.07	0.55
	Complexity	Enc. Time		113%			106%			105%	
		Dec. Time		103%			103%			102%	
{22, 27, 32, 37}	Summary	Class A1	−0.53	−0.12	0.33	−0.01	0.20	0.33	−0.19	0.52	−0.09
		Class A2	−1.97	−0.12	0.21	−1.39	0.62	0.45	−0.45	0.09	0.32
		Class B	−0.81	−0.23	0.37	−0.56	0.34	0.24	−0.15	0.29	0.14
		Class C	0.02	0.41	−0.01	0.39	0.91	0.57	0.18	0.19	0.39
		Class E	−0.30	−0.12	−0.15	0.11	0.49	0.01	0.19	0.51	−0.03
		Overall	**−0.69**	**0.09**	**0.17**	**0.04**	**0.51**	**0.32**	**−0.11**	**0.27**	**0.19**
		Class D	0.32	0.62	0.47	0.47	0.33	0.72	0.19	0.51	−0.03
		Class F	0.19	−0.24	0.37	0.13	0.00	−0.43	0.50	−0.08	0.55
	Complexity	Enc. Time		110%			105%			102%	
		Dec. Time		111%			105%			104%	

higher, so that the weight becomes larger to provide more useful information. The weight matrices corresponding to different block sizes have a consistent changing trend; that is, the weight for the high-frequency coefficients is significantly larger than that of the low-frequency coefficients. At the same time, the weight matrices under different bit rates also have the same distribution characteristics.

Table 1 shows the BD-rate performance. Under the condition of high bit rates, the proposed linear model-based CAIQ mode achieves an average of 1.66%, 0.50%, and 0.65% improvement in the LDP, LDB, and RA configurations, respectively. Simultaneously, this mode has no significant impact on the performance of the UV component. Note that the performance in the LDP mode is the highest. We think this may be related its poor predictive decorrelation ability under a single reference frame configuration. The weak predictive decorrelation ability makes the CAIQ mode, based on the correlation between the prediction and the residual, have greater potential.

In addition, the BD-rate performance at low bit rates is significantly lower than that at high bit rates, indicating that our proposed method has a larger potential at high bit rates. We think that the loss of high-frequency information caused by quantization makes the high-frequency coefficients in the prediction signal relatively small at low bit rates; therefore, the strategy of relying on the frequency correlation for information compensation may not be effective. Note that our proposed CAIQ is an RDO-based TU-level inv-quantization mode, so in the decoding process, only blocks hitting this mode will utilize CAIQ. Considering that the optimized first-order linear model is very simple, our method does not increase the decoding complexity.



(a) Cactus (LDB configuration, QP = 22, Linear model)

(b) Cactus (LDB configuration, QP = 22, Nonlinear model)

(c) ParkRunning3 (RA configuration, QP = 32, Linear model)

(d) ParkRunning3 (RA configuration, QP = 32, Nonlinear model)

FIGURE 7. Block-level visualization of hitting CAIQ mode. The red and gray block boundaries indicate the TUs that have hit and missed the CAIQ mode, respectively. In (a) and (c), the utilized linear model is "$\mathbb{L}(f_P) + \hat{f}_R$". In (b) and (d), the utilized nonlinear model is "$\mathbb{F}_{3\times3}(f_P) + \hat{f}_R$".

In Figs. 7(a) and 7(c), we visualize the hitting blocks of two frames. Note that in areas with rich textures, object edges or moving areas are more likely to be selected, while static or flat areas are less likely to be selected.

2) PERFORMANCE OF NONLINEAR MODEL

Based on the linear model, we further extend the nonlinear CNN-based mapping model. In this section, we use a single-branch network structure, that is, replacing $A \cdot f_P$ with $\mathbb{F}(f_P)$. Since our network structure is a pure convolution operation, the network model can be applied to all TU blocks of various sizes.

Table 2 shows the specific results. Compared with the linear model, the nonlinear model exhibits a more powerful compensation capability. Note that the nonlinear model has achieved a performance improvement of 3 times that of the linear model. In addition, a similar conclusion is that the nonlinear model-based CAIQ mode shows greater potential at high bit rates. Taking the RA configuration as an example, the average BD-rate performance of the high bit rates is 2.31%, which is more than twice the performance at the low bit rates. While the powerful nonlinear modeling ability brings significant performance gain, it also inevitably brings an increase in the encoding and decoding complexity. At the high bit rates of the RA configuration, the CAIQ solution integrating the nonlinear models increases the decoding complexity to 426% (in the CPU environment). Here, we provide a reference GPU decoding time (decoding directly on the CPU-based bitstream, just for reference). Compared to CPUs, GPUs can only provide approximately 21% of time savings. We think this may be caused by frequent and noncontinuous memory exchange.

In Table 3, we observe that the hitting ratio of the nonlinear model is slightly higher than that of the linear model, but the overall difference between the two is relatively small. In Figs. 7(b) and 7(d), we visualize the hitting blocks based on the nonlinear models. An interesting phenomenon is that the blocks hit by the linear model and the nonlinear model are similar. This means that our proposed context-adaptive inverse quantization is always helpful for blocks with texture, motion, edges, etc. Note that the nonlinear model can achieve a finer-grained representation in the inverse quantization process through complex nonlinear operations compared with the linear operation.

C. ANALYSES

1) IMPACT ON PARTITION RESULTS

The proposed CAIQ mode is a TU-level technology. In VVC, the maximum block size allowed by the transform

TABLE 2. BD-rate results of proposed nonlinear model ($\hat{f}_R^C = \mathbb{F}_{3\times3}(f_P) + \hat{f}_R$) on VTM-1.0.

QP	Class	Sequence	LDP Y (%)	LDB Y (%)	RA Y (%)
{17, 22, 27, 32}	Summary	Class A1	−8.67	−3.98	−4.19
		Class A2	−6.89	−2.56	−2.67
		Class B	−5.18	−1.97	−2.67
		Class C	−1.17	−0.23	−0.35
		Class E	−4.67	−1.80	0.05
		Overall	**−5.07**	**−1.98**	**−2.31**
		Class D	−0.33	0.08	0.05
		Class F	−0.14	−0.09	0.34
	Complexity	Enc. Time (CPU)	691%	579%	506%
		Dec. Time (CPU)	630%	576%	426%
		Dec. Time (GPU)	501%	449%	335%
{22, 27, 32, 37}	Class A1	Tango2	−6.47	−2.04	−2.18
		FoodMarket4	−8.50	−2.82	−2.90
		CampfireParty2	−0.63	−0.25	−0.66
	Class A2	CatRobot1	−3.27	−0.70	−1.26
		DaylightRoad2	−4.03	−1.43	−1.82
		ParkRunning3	−4.63	−1.34	−0.85
	Class B	MarketPlace	−3.15	−0.39	−1.08
		RitualDance	−1.26	−0.06	−0.23
		Cactus	−2.63	−0.77	−0.95
		BasketballDrive	−4.55	−1.17	−1.59
		BQTerrace	−2.22	−0.92	−1.12
	Class C	BasketballDrill	−0.34	0.29	0.09
		BQMall	−0.62	−0.00	0.03
		PartyScene	0.07	0.43	0.27
		RaceHorsesC	−1.43	−0.25	−0.38
	Class D	BasketballPass	−0.15	0.13	0.19
		BQSquare	0.60	0.23	0.24
		BlowingBubbles	0.12	0.36	−0.04
		RaceHorses	−0.45	−0.26	0.14
	Class E	FourPeople	−1.04	−0.29	−0.45
		Johnny	−2.49	−0.35	−0.55
		KristenAndSara	−1.43	−0.31	−0.52
	Class F	BasketballDrillText	−0.46	0.21	0.06
		ChinaSpeed	0.21	0.56	0.33
		SlideEditing	0.13	−0.07	0.67
		SlideShow	0.27	−0.03	0.46
	Summary	Class A1	−5.20	−1.70	−1.91
		Class A2	−3.98	−1.16	−1.31
		Class B	−2.76	−0.66	−0.99
		Class C	−0.58	0.12	−0.00
		Class E	−1.65	−0.32	−0.51
		Overall	**−2.70**	**−0.69**	**−0.90**
		Class D	0.03	0.24	0.13
		Class F	0.04	−0.17	0.38
	Complexity	Enc. Time (CPU)	603%	503%	432%
		Dec. Time (CPU)	513%	452%	313%
		Dec. Time (GPU)	401%	361%	254%

TABLE 3. Block size distribution of valid residual blocks.

Sequence	Resolution ≤ 720p (ClassC, ClassD, ClassF)	Resolution ≥ 720p (ClassA1, ClassA2, ClassB, ClassE)
$s \le 4$	32%	20%
$4 \le s \le 8$	44%	53%
$8 \le s \le 16$	18%	18%
$16 \le s \le 32$	5%	6%
$32 \le s \le 64$	1%	3%

is 64×64. Taking the nonlinear model as an example, we analyze the hitting ratio of the CAIQ mode on each TU size. The mode hitting ratio results are shown in Fig. 8, where (a) and (b) correspond to the results of using linear and nonlinear models, respectively. Linear models are only trained and utilized for blocks that do not exceed 32×32

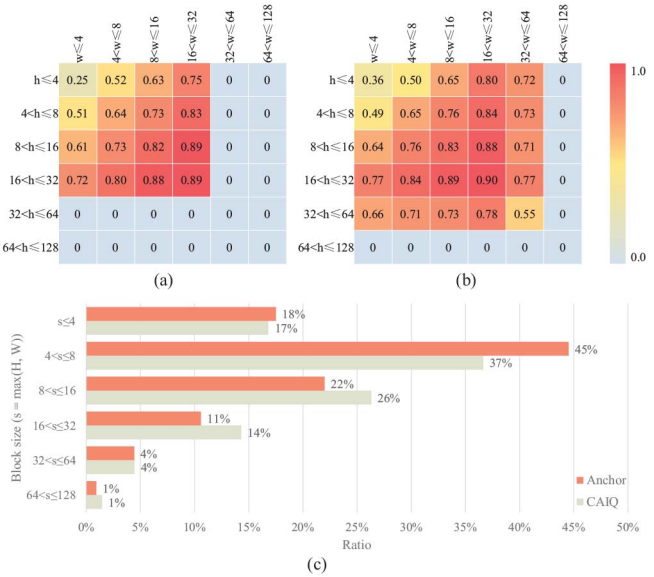


FIGURE 8. (a) Block-level mode hitting ratio, where the utilized model is "$\mathbb{L}(f_P) + \hat{f}_R$". Note that "0" indicates that the ratio is less than 1%. (b) Block-level mode hitting ratio, where the utilized model is "$\mathbb{F}_{3\times3}(f_P) + \hat{f}_R$". (c) Influence of CAIQ mode using the same configuration as (b) on block partition.

due to the limitation of the training set (the absolute number of larger blocks is relatively small). Nonlinear models can be used for any block size due to the characteristics of a pure convolutional neural network structure (the maximum allowed transformation block size is 64×64). Note that the hitting ratio of the CAIQ mode is relatively low on extremely small blocks (e.g., 4 × 4), and the hitting ratio increases with an increasing block size. This indicates that the frequency-based mapping scheme is more friendly for larger blocks. We surmise two causes. One is that the decorrelation effect of the transform is weak in small blocks [2], and the independence between the coefficients is weakened, which limits the performance of CAIQ. Another cause is that small blocks are more likely to achieve high precision prediction, thereby weakening the association between the prediction signals and residual signals, further affecting the hitting ratio.

As shown in Fig. 8-(c), we analyze the effect of the CAIQ mode on the distribution of block size, where $s = max(h, w)$. Note that when the CAIQ mode is disabled, the distribution of various block sizes is {18%, 45%, 22%, 11%, 4%, 1%}. When the CAIQ mode is enabled, the block size distribution changes to {−1%, −8%, 4%, 3%, 0%, 0%}. Therefore, our proposed CAIQ mode tends to divide the input signal into larger TUs.

2) EFFECT OF BLOCK SIZE AND BLOCK CONTENT

We analyze the influence of block sizes on CAIQ by adjusting the maximum allowable block size S_{bound}. When S_{bound} is equal to 16, we will directly skip checking the CAIQ mode for TUs whose width or height exceeds 16. Simultaneously, there is no need to encode the mode flag for these skipped TUs. Fig. 10 shows the trend of the BD-rate performance

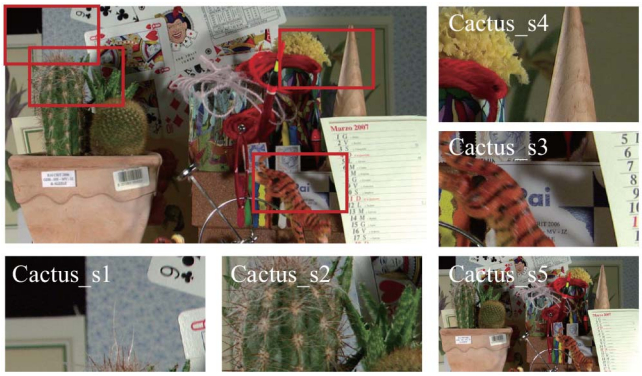


FIGURE 9. Cactus_s1∼s4 are low-definition videos (416×240) cropped from the original sequence, where Cactus_s1 and Cactus_s3 have complex motion (rotation), Cactus_s2 and Cactus_s4 have complex texture. Note that Cactus_s4 can be considered approximately stationary. Cactus_s5 is a low-resolution video (416×240) down-sampled from the original sequence.

TABLE 4. BD-rate of the proposed nonlinear model ($\hat{f}_R^c = \mathbb{F}_{3\times3}(f_P) + \hat{f}_R$) on cropped/downsampled sequences in Fig. 9.

Sequence	LDP	LDB	RA
Cactus	−5.52%	−2.00%	−2.64%
Cactus_s1	−7.24%	−2.58%	−3.49%
Cactus_s2	−5.23%	−1.19%	−1.75%
Cactus_s3	−5.90%	−2.66%	−3.46%
Cactus_s4	−4.72%	−1.61%	−1.73%
Cactus_s5	−0.12%	0.46%	0.03%

and decoding complexity as S_{bound} increases. Specifically, when S_{bound} changes from 8 to 32, the performance shows an approximate linear growth trend. However, when it is further increased to 64, the growth trend decreased significantly. The difference is that the decoding complexity has always maintained a relatively stable growth.

An observed phenomenon is that compared to low-resolution sequences, our approach works better in high-resolution sequences. First, we divide the standard test sequence into two groups according to the resolution. Sequences below 720p are included in ClassC, ClassD, and ClassF, and sequences higher than 720p are included in ClassA1, ClassA2, ClassB and ClassE. The statistical results in Table 3 show that the proportion of small blocks in low-resolution sequences is higher, which limits the performance of CAIQ.

Considering that the resolution reflects the complexity of the content in the unit area, we further attempt to crop/downsample high-resolution sequences into low-resolution sequences. We use Cactus as an example to analyze the impact of content on CAIQ performance. Specifically, we crop/down-sample the original "Cactus" (1920×1080) into multiple sub-sequences (416×240) with various content characteristics. Table 4 shows the BD-rate performance of different sequences. Taking the LDP configuration as an example, we can observe a significant performance difference among the cropped subsequences (Cactus_s1∼s4). The average performances of Cactus_s1 and Cactus_s3 with large motion is the highest. However, Cactus_s4 basically does not have any motion, so its overall

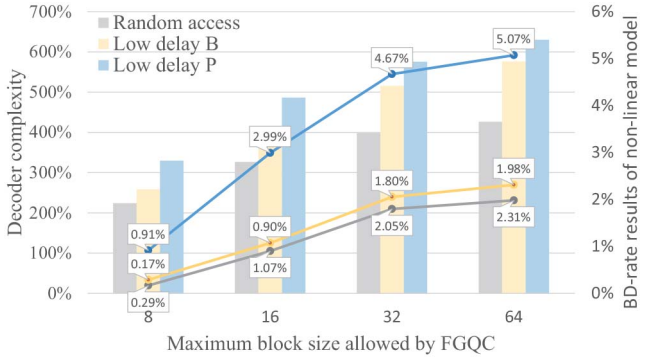


FIGURE 10. Relationship between the maximum allowable block size and performance and decoding complexity.

performance is the lowest. This indicates that our proposed CAIQ is more suitable for processing content with complex motion and texture. In particular, Cactus_s5 is achieved based on the original sequence, so its content is similar to the original sequence, but the content complexity in the unit area is higher. Since CAIQ has almost no performance gain on this sequence, we can conclude that content, instead of resolution, plays a more important role in the performance of CAIQ.

3) EFFECT OF MODEL COMPLEXITY

We compared a variety of inverse quantization models and analyzed the impact of model complexity on the CAIQ mode. First, we designed a first-order linear model Φ_1 based on the assumption of coefficient independence and optimized the weights and biases, position by position. Second, the CNN-based nonlinear model Φ_2 is utilized to achieve multi-position parameter sharing and joint optimization, where the convolution kernel size is set to 1×1, and only a single input f_P is modeled. Third, we expanded Φ_2 to Φ_3 by increasing the convolution kernel size to 3×3 while, at the same time, keeping the input unchanged. Fourth, we used a dual-channel input, namely, $\{f_P, \hat{f}_R\}$, and constructed the model Φ_4 based on a 3×3 convolution. Fifth, we used a 5×5 convolution kernel to replace the corresponding kernels in Φ_4 to construct model Φ_5. For each model, we used the same training data and configuration parameters and started training from the initial state.

The overall BD-rate performance and decoding complexity are shown in Table 5. We can conclude that based on a simple linear model, a certain performance gain can be achieved without increasing the complexity of the decoder, which confirms the effectiveness and potential of our proposed CAIQ mode. In addition, the introduction of the nonlinear model greatly improves the performance. Simultaneously, the decoding complexity also increases rapidly. Note that we did not execute any speed optimizations for the neural network inference. The performance of CAIQ (e.g., in RA configuration) based on the linear model Φ_1 is only −0.11%. Replacing it with a nonlinear model Φ_2 can increase the performance by more than 4 times under the condition

TABLE 5. BD-rate (and decoding time in parentheses) of different CAIQ models.

Model	QP = {17, 22, 27, 32}			QP = {22, 27, 32, 37}		
	LDP	LDB	RA	LDP	LDB	RA
$\Phi_1 = \mathbb{L}(f_P) + \hat{f}_R$	−1.66% (103%)	−0.50% (103%)	−0.65% (102%)	−0.69% (111%)	0.04% (105%)	−0.11% (104%)
$\Phi_2 = \mathbb{F}_{1\times1}(f_P) + \hat{f}_R$	−3.92% (505%)	−1.48% (438%)	−1.70% (338%)	−2.06% (377%)	−0.32% (320%)	−0.54% (238%)
$\Phi_3 = \mathbb{F}_{3\times3}(f_P) + \hat{f}_R$	−5.07% (630%)	−1.98% (576%)	−2.31% (426%)	−2.70% (513%)	−0.69% (452%)	−0.90% (313%)
$\Phi_4 = \mathbb{F}_{3\times3}(f_P, \hat{f}_R) + \hat{f}_R$	−5.43% (1109%)	−2.14% (997%)	−2.59% (736%)	−3.08% (959%)	−0.92% (843%)	−1.06% (542%)
$\Phi_5 = \mathbb{F}_{5\times5}(f_P, \hat{f}_R) + \hat{f}_R$	−5.24% (2086%)	−2.04% (1673%)	−2.26% (1104%)	−2.97% (1847%)	−0.86% (1395%)	−0.93% (785%)

of a limited increase in complexity. In Φ_2, only using a 1×1 convolution makes the receptive field of the neural network always focus on the current position, it is difficult to effectively utilize the intracorrelations, and the network's nonlinear modeling ability is relatively weak. As a result, when the convolution kernel size is expanded to 3×3, we can again observe a significant performance improvement.

Although Φ_2 and Φ_3 consider the influence of $\hat{f}_R$ in the inverse quantization process by using a residual connection structure, we believe that adding $\hat{f}_R$ to the network may bring additional performance improvements. In Φ_4 and Φ_5, we constructed a dual-branch network structure based on 3×3 and 5×5 convolution kernels, respectively. The results show that compared to Φ_3, dual-branch networks can bring certain performance improvements. However, on the one hand, the average gain is approximately -0.3%, while the decoding complexity is doubled. On the other hand, a larger convolution kernel size does not mean higher performance, which may increase the difficulty of model training. Therefore, the performance of the model Φ_5 is slightly lower than that of Φ_4.

4) EFFECT OF COMBINING WITH RECURSIVE PARTITIONING PROCESS

Our proposed CAIQ is integrated with the inverse quantizer, thereby serving as a TU-level optional mode. As a result, the CAIQ mode decision is bound to the recursive block partition decision process, namely as TU-Loop. We want to claim that CAIQ has greater performance potential when combined with block partition decisions. Therefore, we apply the CAIQ mode at the frame level, that is, check the CAIQ mode TU-by-TU before the in-loop de-blocking filters, namely, Frame-Loop. Table 6 shows the BD-rate results. The TU-Loop-based strategy is significantly better than that of Frame-Loop, indicating that dynamically considering contextual information in the recursive partition process is more efficient.

5) EXTEND TO BETTER BASELINE CODECS

We test the BD-rate performance of the nonlinear models on VTM-6.0 and VTM-13.0 under the same configuration. Note that we reacquire the training set based on the corresponding reference software version and execute model training. The results are shown in Table 7. Overall, we can observe obvious performance drops or even performance loss. However, for partial high-resolution sequences, the proposed CAIQ method can still obtain a certain performance gain. We

TABLE 6. BD-rate of different integrated strategies ($\Phi_3 = \mathbb{F}_{3\times3}(f_P) + \hat{f}_R$).

QP	Method	Class	LDP(%)	LDB(%)	RA(%)
{17, 22, 27, 32}	TU-Loop	**Overall**	**−5.07**	**−1.98**	**−2.31**
	Frame-Loop	Class A1	−6.53	−2.54	−2.67
		Class A2	−3.99	−0.95	−0.59
		Class B	−3.25	−0.72	−1.29
		Class C	−0.56	0.18	−0.02
		Class E	−2.89	−0.66	0.20
		Overall	**−3.26**	**−0.85**	**−1.06**
		Class D	−0.05	0.44	0.20
		Class F	−0.07	−0.03	0.31
{22, 27, 32, 37}	TU-Loop	**Overall**	**−2.70**	**−0.69**	**−0.90**
	Frame-Loop	Class A1	−4.36	−1.40	−1.33
		Class A2	−2.65	−0.43	−0.25
		Class B	−2.02	−0.18	−0.53
		Class C	−0.22	0.35	0.08
		Class E	−1.22	−0.01	−0.12
		Overall	**−1.98**	**−0.28**	**−0.44**
		Class D	0.15	0.51	0.12
		Class F	0.03	0.29	0.40

attribute the reasons for the performance degradation to the following three points. First, our method relies on the correlation between the prediction signal and the original residual, and it has better hitting results for regions with low prediction accuracy (such as complex motion, texture, edge regions, etc., see Fig. 7). Therefore, the improvement of inter-frame prediction efficiency may reduce the correlation between signals, thereby compressing the performance space of our method. Second, our proposed CAIQ utilizes more encoded information to achieve context-adaptive loss compensation, which may overlap with other context-adaptive technologies, such as adaptive loop filtering (ALF). When the ALF is turned off, we can observe a significant performance improvement of the proposed method. Third, the introduction of a large number of new technologies makes the block-level mode more complicated, and we still follow the strategy of using the same model for various contents. Therefore, we think that adding more context information may be useful to further enhance the performance of CAIQ when using a better baseline codec.

VI. CONCLUSION

In this paper, we propose a TU-level context-adaptive inverse quantization method based on already coded information and frequency-domain correlations. Unlike the video coding standard that separates the predictive coding module and residual coding module at the block level, we propose that the prediction signal helps assist residual reconstruction. Based on correlation analyses, we design linear and nonlinear coefficient-level mapping models and apply them to the

TABLE 7. BD-rate of proposed nonlinear model ($\hat{f}_R^C = \mathbb{F}_{3\times3}(f_P) + \hat{f}_R$) and linear model on better baseline codecs.

VTM Version QP={17,22,27,32}	Class	Non-Linear Model						Linear Model					
		w. ALF			w/o ALF			w. ALF			w/o ALF		
		LDP	LDB	RA	LDP	LDB	RA	LDP	LDB	RA	LDP	LDB	RA
VTM-6.0	Class A1	−0.64	−0.24	−0.43	−1.93	−0.99	−1.13	−0.04	0.04	0.04	−0.08	0.00	−0.01
	Class A2	−0.38	0.04	−0.26	−3.01	−1.20	−1.52	0.09	0.04	0.07	0.09	0.05	0.01
	Class B	−0.31	−0.10	−0.26	−1.48	−0.74	−0.92	−0.00	0.07	0.02	0.04	0.02	0.02
	Class C	0.14	0.16	0.05	−0.27	−0.06	−0.12	0.02	−0.01	0.06	0.01	−0.09	−0.01
	Class E	−0.28	−0.12	−0.16	−1.49	−0.66	−0.65	−0.03	0.06	−0.14	0.21	0.11	−0.03
	Overall	**−0.30**	**−0.05**	**−0.21**	**−1.64**	**−0.73**	**−0.87**	**0.02**	**0.05**	**0.04**	**0.02**	**0.03**	**0.01**
	Enc. Time	365%	339%	276%	366%	337%	276%	99%	99%	99%	102%	102%	103%
	Dec. Time	342%	305%	256%	408%	360%	282%	103%	106%	105%	111%	110%	108%
VTM-13.0	Class A1	−0.01	0.06	−0.00	−0.39	−0.29	−0.26	−0.00	−0.02	−0.06	0.09	0.08	0.07
	Class A2	−0.14	0.04	0.04	−0.50	−0.23	−0.12	0.06	0.02	0.01	0.13	0.09	0.05
	Class B	−0.10	−0.04	0.06	−0.09	−0.26	−0.20	0.04	0.00	0.08	0.09	0.04	0.05
	Class C	0.30	0.23	0.25	−0.03	0.07	0.22	0.09	0.05	0.06	0.16	0.12	0.09
	Class E	0.04	0.04	0.14	0.07	0.02	0.15	0.16	0.04	0.11	0.04	0.04	0.08
	Overall	**0.02**	**0.07**	**0.10**	**−0.19**	**−0.14**	**−0.04**	**0.07**	**0.02**	**0.04**	**0.07**	**0.01**	**0.07**
	Enc. Time	361%	323%	260%	388%	320%	293%	111%	110%	107%	113%	111%	108%
	Dec. Time	274%	240%	196%	391%	280%	277%	100%	102%	99%	104%	102%	102%

de-quantizer to construct a new optional inverse quantization mode, namely, CAIQ. The experimental results show that a simple linear model can bring a certain BD-rate performance gain, and the introduction of a nonlinear operation can further improve the performance.

In the future, more block-level mode information can be introduced as context to further enhance the performance of CAIQ. The processing strategy based on the frequency domain will also facilitate coefficient-level independent mapping. Furthermore, the method can be further extended to intra-frame coding.

ACKNOWLEDGMENT

The authors acknowledge the support of the GPU cluster built by the MCC Lab of the School of Information Science and Technology of USTC.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] 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.

[4] T. Wedi and S. Wittmann, "Quantization offsets for video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 324–327.

[5] Y. Mo, J. Xiong, J. Chen, and F. Xu, "Quantization matrix coding for high efficiency video coding," in *Advances on Digital Television and Wireless Multimedia Communications*. Berlin, Germany: Springer, 2012, pp. 244–249.

[6] V. Sze, M. Budagavi, and G. J. Sullivan, *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. Cham, Switzerland: Springer, 2014.

[7] Y. Wang, Z. Mei, C.-Y. Tsai, I. Katsavounidis, and C.-C. J. Kuo, "A machine learning approach to optimal inverse discrete cosine transform (IDCT) design," 2021, *arXiv:2102.00502*.

[8] J. He, E.-H. Yang, F. Yang, and K. Yang, "Adaptive quantization parameter selection for H.265/HEVC by employing inter-frame dependency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3424–3436, Dec. 2018.

[9] G. Xiang, H. Jia, M. Yang, Y. Li, and X. Xie, "A novel adaptive quantization method for video coding," *Multimedia Tools Appl.*, vol. 77, no. 12, pp. 14817–14840, 2018.

[10] H. Yin, H. Wang, X. Huang, and H. Yin, "Efficient hard-decision quantization using an adaptive deadzone offset model for video coding," *IEEE Access*, vol. 7, pp. 151215–151229, 2019.

[11] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[12] M. Xu, T. N. Canh, and B. Jeon, "Simplified level estimation for rate-distortion optimized quantization of HEVC," *IEEE Trans. Broadcast.*, vol. 66, no. 1, pp. 88–99, Mar. 2020.

[13] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 82–93, Jan. 1990.

[14] H. Schwarz, T. Nguyen, D. Marpe, and T. Wiegand, "Hybrid video coding with trellis-coded quantization," in *Proc. Data Compression Conf. (DCC)*, 2019, pp. 182–191.

[15] C. A. Gonzales and E. Viscito, "Motion video adaptive quantization in the transform domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 4, pp. 374–378, Dec. 1991.

[16] J. Luo, C. W. Chen, K. J. Parker, and T. S. Huang, "A scene adaptive and signal adaptive quantization for subband image and video compression using wavelets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 343–357, Apr. 1997.

[17] D. Grois and A. Giladi, "Perceptual quantization matrices for high dynamic range H.265/MPEG-HEVC video coding," in *Proc. Appl. Digit. Image Process. XLII*, vol. 11137, 2020, Art. no. 111370O. [Online]. Available: https://doi.org/10.1117/12.2525406

[18] X. HoangVan, "Adaptive quantization parameter estimation for HEVC based surveillance scalable video coding," *Electronics*, vol. 9, no. 6, p. 915, 2020.

[19] Y. Yan, G. Xiang, Y. Li, X. Xie, and H. Jia, "An adaptive spatio-temporal perception aware quantization algorithm for AVS2," *J. Vis. Commun. Image Represent.*, vol. 73, Nov. 2020, Art. no. 102917.

[20] M. A. Robertson and R. L. Stevenson, "DCT quantization noise in compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 27–38, Jan. 2005.

[21] X. Wei and Z. Xia, "An improved context adaptive hard-decision quantization algorithm," in *Proc. 13th Int. Conf. Natural Comput. Fuzzy Syst. Knowl. Disc. (ICNC-FSKD)*, 2017, pp. 292–297.

[22] S. Liu, J. Chen, Y. Ai, and S. Rahardja, "An optimized quantization constraints set for image restoration and its GPU implementation," *IEEE Trans. Image Process.*, vol. 29, pp. 6043–6053, 2020.

[23] M. Winken, A. Roth, H. Schwarz, and T. Wiegand, "Multi-frame optimized quantization for high efficiency video coding," in *Proc. Picture Coding Symp. (PCS)*, 2015, pp. 159–163.

[24] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 594–602, May 2001.

[25] A. Ghosh and R. Chellappa, "Deep feature extraction in the DCT domain," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, 2016, pp. 3536–3541.
[26] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2018, pp. 3933–3944.
[27] M. Ehrlich and L. S. Davis, "Deep residual learning in the JPEG transform domain," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3484–3493.
[28] M. Sun, X. He, S. Xiong, C. Ren, and X. Li, "Reduction of JPEG compression artifacts based on DCT coefficients prediction," *Neurocomputing*, vol. 384, pp. 335–345, Apr. 2020.
[29] U. S. Kim, S. D. Kim, and M. H. Sunwoo, "Novel intra prediction algorithm using residual prediction for low power multimedia codecs," in *Proc. 12th Int. Symp. Integr. Circuits*, 2009, pp. 324–327.
[30] J.-W. Kang, C.-C. Lou, S.-H. Kim, and C.-C. J. Kuo, "Efficient HD video coding with joint first-order-residual (FOR) and second-order-residual (SOR) coding technique," *J. Vis. Commun. Image Represent.*, vol. 24, no. 1, pp. 1–11, 2013.
[31] B. Huang, F. Henry, C. Guillemot, and P. Salembier, "Mode dependent vector quantization with a rate-distortion optimized codebook for residue coding in video compression," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2015, pp. 1433–1437.
[32] C.-H. Yeh, C.-W. Lee, S.-J. F. Jiang, Y.-H. Sung, and W.-J. Huang, "Second order residual prediction for HEVC inter coding," in *Proc. Annu. Summit Conf. Asia-Pac. Signal Inf. Process. Assoc. (APSIPA)*, 2014, pp. 1–4.
[33] K. Liu, D. Liu, H. Li, and F. Wu, "Convolutional neural network-based residue super-resolution for video coding," in *Proc. IEEE Visual Commun. Image Process. (VCIP)*, 2018, pp. 1–4.
[34] D. N. Reshef *et al.*, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[36] J. Chen and E. Alshina, "Algorithm description for versatile video coding and test model 1 (VTM 1)," Joint Video Experts Team (JVET), San Diego, CA, USA, Rep. JVET-J1002, 2018.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

KANG LIU (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Xidian University, Xi'an, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Big Data, University of Science and Technology of China, Hefei, China. His research interests include image/video/feature coding and machine learning.

DONG LIU (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively.

He was a Member of Research Staff with Nokia Research Center, Beijing, China, from 2009 to 2012. He joined USTC in 2012 and became a Professor in 2020. He has authored or coauthored more than 100 papers in international journals and conferences. He has 20 granted patents. He has several technical proposals adopted by international and domestic standardization groups. His research interests include image and video processing, coding, analysis, and data mining. He received the 2009 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award and the VCIP 2016 Best 10% Paper Award. He and his students were winners of several technical challenges held in ICCV 2019, ACM MM 2019, ACM MM 2018, ECCV 2018, CVPR 2018, and ICME 2016. He serves or had served as the Chair of IEEE Future Video Coding Study Group, the Publicity Co-Chair for ICME 2021, and the Registration Co-Chair for ICME 2019. He is a Senior Member of CCF and CSIG, an Elected Member of MSA-TC of IEEE CAS Society.

LI LI (Member, IEEE) received the B.S. and Ph.D. degrees in electronic engineering from University of Science and Technology of China (USTC), Hefei, Anhui, China, in 2011 and 2016, respectively.

He is a Research Fellow with the Department of Electronic Engineering and Information Science, USTC. He was a Visiting Assistant Professor with the University of Missouri, Kansas City, from 2016 to 2020. His research interests include image/video coding and processing. He received the Best 10% Paper Awards at the 2016 IEEE Visual Communications and Image Processing and the 2019 IEEE International Conference on Image Processing.

HOUQIANG LI (Fellow, IEEE) received the B.S., M.Eng., and Ph.D. degrees in electronic engineering from the University of Science and Technology of China, Hefei, China, in 1992, 1997, and 2000, respectively.

He is currently a Professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. He has authored and coauthored over 200 papers in journals and conferences. His research interests include image/video coding, image/video analysis, computer vision, and reinforcement learning. He is the recipient of National Technological Invention Award of China (Second Class) in 2019 and the National Natural Science Award of China (Second Class) in 2015. He was the recipient of the Best Paper Award for VCIP 2012, the Best Paper Award for ICIMCS 2012, and the Best Paper Award for ACM MUM 2011. He is the Winner of NSFC for Distinguished Young Scientists, the Distinguished Professor of Changjiang Scholars Program of China, and the Leading Scientist of Ten Thousand Talent Program of China. He served as the General Co-Chair of ICME 2021 and the TPC Co-Chair of VCIP 2010. He serves as an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA, and had served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2013.