Deep Inter Prediction with Error-Corrected Auto-Regressive Network for Video Coding

YUZHANG HU, WENHAN YANG, JIAYING LIU, and ZONGMING GUO,
Wangxuan Institute of Computer Technology, Peking University

Modern codecs remove temporal redundancy of a video via inter prediction, i.e., searching previously coded frames for similar blocks and storing motion vectors to save bit-rates. However, existing codecs adopt block-level motion estimation, where a block is regressed by reference blocks linearly and is doomed to fail to deal with non-linear motions. In this article, we generate virtual reference frames (VRFs) with previously reconstructed frames via deep networks to offer an additional candidate, which is not constrained to linear motion structure and further significantly improves coding efficiency. More specifically, we propose a novel deep Auto-Regressive Moving-Average (ARMA) model, Error-Corrected Auto-Regressive Network (ECAR-Net), equipped with the powers of the conventional statistic ARMA models and deep networks jointly for reference frame prediction. Similar to conventional ARMA models, the ECAR-Net consists of two stages: Auto-Regression (AR) stage and Error-Correction (EC) stage, where the first part predicts the signal at the current time-step based on previously reconstructed frames, while the second one compensates for the output of the AR stage to obtain finer details. Different from the statistic AR models only focusing on short-term temporal dependency, the AR model of our ECAR-Net is further injected with the long-term dynamics mechanism, where long temporal information is utilized to help predict motions more accurately. Furthermore, ECAR-Net works in a configuration-adaptive way, i.e., using different dynamics and error definitions for the Low Delay B and Random Access configurations, which helps improve the adaptivity and generality in diverse coding scenarios. With the well-designed network, our method surpasses HEVC on average 5.0% and 6.6% BD-rate saving for the luma component under the Low Delay B and Random Access configurations and also obtains on average 1.54% BD-rate saving over VVC. Furthermore, ECAR-Net works in a configuration-adaptive way, i.e., using different dynamics and error definitions for the Low Delay B and Random Access configurations, which helps improve the adaptivity and generality in diverse coding scenarios.

CCS Concepts: • **Computing methodologies → Image processing;**

Additional Key Words and Phrases: High Efficient Video Coding (HEVC), inter prediction, deep learning, virtual reference frame, Error-Corrected Auto-Regressive Network, Versatile Video Coding (VVC)

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018AAA0102702, the National Natural Science Foundation of China under Contract No. 62172020, and a research achievement of Key Laboratory of Science, Techonology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology) and State Key Laboratory of Media Convergence Production Technology and Systems.
Authors' address: Y. Hu, W. Yang, J. Liu (corresponding author), and Z. Guo, Wangxuan Institute of Computer Technology, Peking University, Zhongguancun North Street 128#, Haidian, Beijing, China; emails: {yuzhanghu, yangwenhan, liujiaying, guozongming}@pku.edu.cn.

1551-6857/2023/01-ART33 $15.00
https://doi.org/10.1145/3528173

ACM Reference format:
Yuzhang Hu, Wenhan Yang, Jiaying Liu, and Zongming Guo. 2023. Deep Inter Prediction with Error-Corrected Auto-Regressive Network for Video Coding. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1s, Article 33 (January 2023), 22 pages.
https://doi.org/10.1145/3528173

1 INTRODUCTION

Inter prediction is leveraged to squeeze out temporal redundancy among video frames in recent video compression coding standards like H.264/AVC [1] and **High Efficient Video Coding** (**HEVC**) [2]. In the stage of inter prediction, for a block that is to be coded (to-be-coded block), the block-wise motion is searched and used to predict the current block based on reference blocks. After that, only the block-level motion and the predicted residue need to be stored, which largely reduces temporal redundancy and saves bit-rates. However, similar blocks can hardly be found from the previously encoded reference frames after the motion compensation on the to-be-coded block when complex motions exist. As a result, there remains a larger predicted residue after inter prediction, which would significantly affect the coding efficiency. Some new techniques are introduced in the **Versatile Video Coding** (**VVC**) [3] to further enhance the inter prediction. For example, the affine motion compensated prediction is designed to model complex inter-frame motions such as rotation by introducing affine transformation. Although coding efficiency is improved to some extent, these techniques are hand-crafted and thus with limited capacities, lacking the flexibility to model different kinds of complex motions.

Recently, deep learning has shown excellent modeling capacities in both high-level computer vision tasks and low-level vision fields, including image restoration [4, 5], image interpolation [6–8], and so on. Then naturally, deep learning techniques have also been introduced to improve various modules of modern codecs, e.g., intra prediction [9–11], fractional interpolation [12, 13], and in-loop filter [14]. Specifically, some methods have been proposed to predict a virtual frame pixel-wise closer to the to-be-coded frame with previously reconstructed frames as an additional reference frame for the inter prediction. For simplicity, following [15], we call the predicted frame the **Virtual Reference Frame** (**VRF**). Some existing works have adopted different ways to generate VRFs. Zhao et al. [15] first applied the adaptive convolution-based **frame rate up conversion** (**FRUC**) algorithm [16] to predict a VRF with two reconstructed frames. The basic FRUC algorithm is further improved with more side information to guide the VRF prediction in successive works. In [17], the temporal index of the input frame is also taken as the input to make the model deal with different coding configurations. Then in [18], the **Quality Parameter** (**QP**) of the input frame is introduced into the quality attention scheme. These two works make use of more extra guidance information to achieve better prediction results, while the temporal perception range of the model is limited, as the model only takes two adjacent frames as input. Laude et al. [19] explored to predict the VRF with more input frames, i.e., four frames in a progressive manner with the PredNet proposed in [20]. However, the experiment results prove that the performance improvement is limited by simply taking more frames as the input. Besides, some advanced network structures like Laplacian Pyramid of Generative Adversarial Network [21] is used to predict the VRF in [22].

Although significant performance improvements have been achieved, these methods still have the limitation from the perspective of model structure, temporal information perception, and consideration of coding scenarios.

— The model capacity of existing works is limited. Some widely used network architectures [16, 20, 21] are directly applied for the generation of the VRF. However, during the coding process of a video sequence, the characteristics of the encoded frames may vary a lot, e.g.,

the changing quality of the frames and the randomly accessed temporal order of frames. These challenging characteristics call for a more powerful model for the generation process.
— Strong temporal consistency in the video sequence has not been fully investigated in existing works. Most of the existing works only take two reconstructed frames as the input of the prediction model. Without the perception for motions in a long time-span, the predicted frame might not be desirable with blurred details and artifacts caused by the inaccurate frame regression and improper fusion.
— Frame prediction methods are good at perceiving and utilizing motion information. However, they might neglect the spatial redundancy and nature image/video statistics. Especially when coming across complex motions, the frame prediction results inevitably include visual artifacts and distorted details. It is necessary to introduce another module/step to further refine the prediction results via utilizing spatial redundancy and nature image/video statistics to further improve the frame visual quality.

In this article, we aim at addressing the above-mentioned three issues. Specifically, we develop an **Error-Corrected Auto-Regressive Network** (**ECAR-Net**), a combination of deep networks and conventional statistical **Auto-Regressive Moving-Average** (**ARMA**) model for temporal modeling. First, the generation of the VRF is modeled as a regression process with previously reconstructed frames. There is a tradeoff between the regression accuracy and complexity when choosing the number of the past steps for the regression window of the current frame. Taking more input frames as the regression input can bring more temporal information, but pays for more memory and computational cost. *We propose to capture long-term temporal dynamics of videos instead of adopting a large regression window to facilitate motion modeling.* Beyond using nearby frames, the long-term temporal dynamics are introduced to offer the information that the input frames do not include in the regression process. Second, as the motion is usually so complex and hard to be predicted, the directly predicted frame inevitably includes distortions and visual artifacts. Therefore, *we propose to adopt an **error correction** (**EC**) module to further improve the quality of the raw regression result with the guidance of an error map.* Finally, different coding configurations increase the complexity of VRF generation using a unified framework. Specifically, under the LDB configuration, the coding order is totally in the temporal order, while under the RA configuration, the coding order is disordered. *Our model works in a configuration-adaptive way. We use the same network structure but adopt different inputs as the long-term temporal dynamics and error map under different coding configurations.* In this way, our method is more general to deal with various coding scenarios. Our contributions are summarized as follows:

— We combine the conventional statistic ARMA model and deep networks to develop the ECAR-Net for VRF prediction. ECAR-Net works in a configuration-adaptive way for better generality and is equipped with stronger capacity in describing motions and dynamics among videos.
— We propose the novel way to model long-term temporal dynamics during the **Auto-Regression** (**AR**) process. Complex motion patterns can be captured with just a fixed small number of input frames while still perceiving the motion during a long time-span.
— We design an EC module under the guidance of an error map to restore the generated distorted details of the AR result and further improve the quality of the final predicted VRF.

The rest of the article is organized as follows. In Section 2, we provide a brief review of related works. In Section 3, we introduce the methodology of our ECAR-Net and describe the implementation details. Experimental results are shown in Section 4. Finally, we make a conclusion in Section 5.

2 RELATED WORKS

2.1 Deep Learning-Based Video Temporal Modeling

Recently, deep learning is widely used for video temporal modeling to synthesize new frames with temporally adjacent ones, known as frame interpolation and extrapolation. Motion patterns among adjacent frames are estimated first in order to predict the motion between the new frames and adjacent frames. Many optical flow estimation methods [23–25] can describe pixel-level displacement and are suitable for the motion estimation.

In [26], motion patterns are estimated followed with target frame synthesis. However, the generation quality of this method depends heavily on the accuracy of the motion estimation. Then in [27], the target frames are synthesized directly by convolutions with adaptive kernels estimated for each pixel. This method is further improved in [16] by making the adaptive kernels separable in order to reduce the time and memory costs. In this way, explicit motion estimation is bypassed so motion estimation is not the bottleneck of generation results with high quality.

Liu et al. [28] proposed to combine the motion-based and synthesis-based methods with deep voxel flow. Pixels of existing frames are copied and feed-forwarded to the specific location according to the estimated deep voxel flow and then fused to synthesize the target frame through pixel-level weighted summation with the estimated mask. Reda et al. [29] proposed to use vector-based method to deal with large-scale motion while use kernel-based method to deal with small-scale motion. In this way, all kinds of motions can be estimated and handled effectively to generate target frames with clear and sharp details.

2.2 Deep Learning-Based Video Coding

With the strong non-linear mapping capacity, many researchers explore to integrate deep learning techniques into the traditional video codecs to improve the efficiency of both specific modules and whole video coding architecture. Some works try to improve the quality of the reconstructed videos with less storage costs while others make efforts to accelerate the coding speed.

Intra prediction is designed to remove spatial redundancy of frames. Li et al. [30] proposed to use the fully connected network to predict the to-be-coded coding blocks to replace the existing linear predicting algorithm in the traditional video codecs. Hu et al. [10] proposed to use the recurrent neural network which has a stronger mapping capacity for intra prediction and further improved the efficiency of intra prediction.

Image restoration with deep learning has also been studied such as deblocking [31], super resolution [32], and significant gain has been achieved. In video codecs, loop filter is the module to suppress the blocking artifacts caused by coding process. Therefore, many methods are proposed to improve the existing loop filter modules with neural networks. Kang et al. [33] proposed a multi-scale convolution neural network to perform loop filter and achieved significant gains over HEVC. Jia et al. [34] took the context into consideration and designed a context-aware neural network to make the model adaptive to the to-be-filtered region. The **Progressive Rethinking Block** (**PRB**) proposed in [14] makes use of the long-term dependency between subsequent blocks of the neural network as well as the side information of coding partition tree. Therefore, more gains are obtained.

The inter prediction module is also enhanced from many aspects. In [13], CNNs are made use of to generate subpixels with existing pixels during the process of motion compensation to facilitate the sub pixel-level motion description. Wang et al. [35] refined the raw inter prediction result with CNNs to obtain a better one. Zhao et al. [36] firstly applied the interpolation network to directly generate the reconstructed blocks rather than perform the existing inter prediction under the RA configuration. Then in [18], a **Multi-scale Quality Attentive Factorized Kernel**

Convolutional Neural Network (**MQ-FKCNN**) is proposed for the generation of VRFs. Multi-domain hierarchical constraints in spatial, frequency, and quality domains are introduced to regularize the training process. For the image extrapolation, Laude et al. [19] used the PredNet proposed in [20] with **long-short term memory** (**LSTM**) [37] to predict future frames under the LDB configuration. The Laplacian Pyramid of Generative Adversarial Network proposed in [21] is also introduced to the video codecs in [22] for the generation of VRFs. Huo et al. [38] proposed to align the reference frames before the generation process. Choi et al. [17] made use of the temporal index as side information to enable the model to predict the to-be-coded frame under different coding configurations simultaneously. These works either apply existing frame generation networks directly, which are not fully optimized for coding scenarios or do not make full use of temporal redundancy for the frame generation process, which limits their capacity to improve efficiency of video codecs. In our work, we take coding scenarios into consideration and design an ECAR-Net to generate high quality VRFs under different coding configurations. Besides, we also focus on making full use of temporal redundancy to obtain more accurate generation results fully capturing long-term dynamics. The superiority of our proposed method in temporal modeling and inter prediction over existing methods is shown in the experiment part.

3 ERROR-CORRECTED AUTO-REGRESSIVE NETWORK FOR INTER PREDICTION

In this section, we first describe the statistical ARMA Model and our improved version for the video frame sequence regression. Then, we describe each module of our model under different coding configurations in detail. At last, we integrate the proposed model into video codecs and describe the implementation details of our method.

3.1 Auto-Regressive Moving-Average Model

The ARMA model is one of the most widely used statistical models to describe time-series data, which is also used in [5–7, 39, 40] to tackle some computer vision tasks. It is the combination of the AR model and the **Moving-Average** (**MA**) model.

The AR model predicts the value at the current time-step with the observations of previous time-steps as follows:

$$\tilde{I}_t = \sum_{i=1}^{p} \alpha_i \cdot I_{t-i}, \tag{1}$$

where $\{I_t | t = 1, \ldots, T\}$ is a time-series and $\tilde{I}_t$ is the predicted AR result of the time step t. α_i is the AR coefficients denoting what percentage of the current regression target can be explained by previous sampled values. p stands for the order of the model, i.e., the length of the time steps for predicting I_t. The MA model, on the other hand, plays the role to further improve the accuracy of the regression result of the AR model with the past regression errors as follows:

$$Z_t = \sum_{i=1}^{p} \beta_i \cdot R_{t-i}, \tag{2}$$

where R_{t-i} is the regression error between the regression result $\widetilde{I}_{t-i}$ and the real data I_{t-i}. These regression errors are propagated and accumulated by the MA model to obtain the MA item denoted as Z_t. Finally, the regression result $\tilde{I}_t$ is corrected by the MA item Z_t to obtain the output of the whole ARMA model $\hat{I}_t$ in a residual learning way as follows:

$$\hat{I}_t = \tilde{I}_t + Z_t. \tag{3}$$

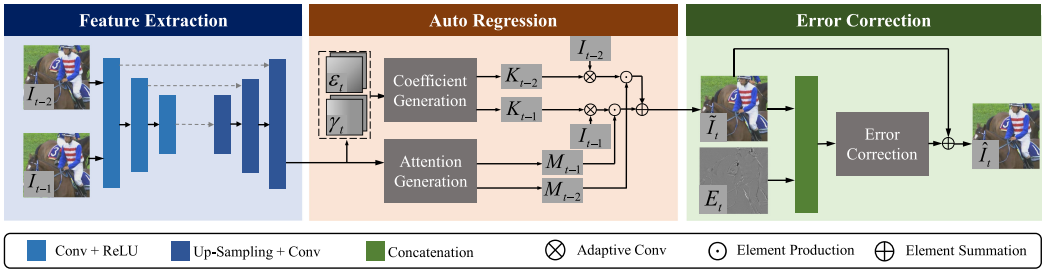


Fig. 1. Architecture of the ECAR-Net. The network uses an auto-encoder to extract features from input frames. The AR coefficients are generated based on the combination of short-term temporal redundancy γ_t and long-term temporal dynamics ε_t, where the quality attention maps are injected for better prediction quality. The raw regression result $\widetilde{I}_t$ is further enhanced with the error map E_t to obtained the corrected $\hat{I}_t$ with better quality, which is fed into the video codecs as the virtual reference frame to facilitate the inter prediction.

3.2 Error-Corrected Auto-Regressive Netwrok

Inspired by the ARMA model, we connect the video frames in an AR way for VRF generation. We first analyze the issue of directly applying ARMA model to VRF generation and then introduce our improved version, called ECAR-Net as shown in Figure 1.

First, describing the relationship of the continuous frames with a constant α_i will lead to inaccurate regression results. In order to solve this problem, it will be better if α_i is dynamically dependent on the current temporal context. Therefore, we regard α_i as a function $\alpha_i(\cdot)$. It is natural to make use of the adjacent p frames for the calculation of $\alpha_i(\cdot)$. However, this will lead to the loss of the temporal information out of the adjacent p frames. In order to solve this problem, we propose the novel long-term temporal dynamics injection scheme, which contains temporal information out of the input p frames and thus can trace the motion patterns of a larger time-span and will be described in detail later. Moreover, the motion and content information of different reconstructed frames might differ a lot so the quality attention modeling should be embedded into $\alpha_t(\cdot)$, which allocates larger linear blend weights to the regions owning more information for better motion and content quality of the target frame. Based on the above analysis, we split $a_t(\cdot)$ into $K_t(\cdot)$ and $M_t(\cdot)$, where the former denotes the regression coefficients generated based on both the short-term redundancy and long-term dynamics, and the latter denotes the quality attention map. Then, the improved AR coefficients are developed as follows:

$$a_i(\varepsilon_t, \gamma_t) = \{K_i(\varepsilon_t, \gamma_t), M_i(\gamma_t)\}, \tag{4}$$

where γ_t stands for short-term temporal redundancy extracted from the previous input frames $\{I_{t-i}\}$ and ε_t stands for the long-term temporal dynamics. Then, the improved AR model is developed as follows:

$$\tilde{I}_t = \sum_{i=1}^{p} I_{t-i} \otimes K_i(\varepsilon_t, \gamma_t) \odot M_i(\gamma_t), \tag{5}$$

where $\otimes$ denotes the adaptive convolution and $\odot$ is the pixel-wise multiplication.

Second, the regression result $\tilde{I}_t$ can be enhanced by better utilizing spatial redundancy and nature image/video statistics, namely, further correction or refinement. In a conventional MA model, it is reflected in Equation (2) that, the AR result is further corrected with p previous regression errors, which results in a heavy burden of storage and calculation. In fact, the regression errors of the frames far away from the current time step contribute little to the correction process of the

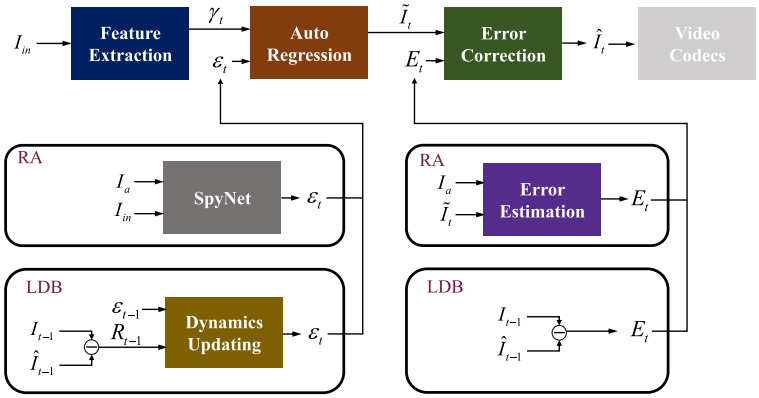


Fig. 2. Pipeline to connect all the sub-modules of the ECAR-Net. The AR model in Equation (5) generates the regression result $\widetilde{I}_t$, which is further refined with the EC Module under the guidance of the Error Map E_t to generate the final VRF $\hat{I}_t$. $\hat{I}_t$ is then utilized to enhance the inter prediction of video codecs. The configuration-adaptive long-term dynamics ε_t and error map E_t are calculated by different sub-modules, which will be described in detail in the following sections.

current frame due to a large time interval. In order to address this issue, we propose an EC module to further enhance the AR regression result with only the nearest prediction error information. This module takes an error map E_t that reflects the regressing error under the given temporal context of the tth frame, together with $\tilde{I}_t$ to calculate a correction term, which is added to $\tilde{I}_t$ to generate the final output of our model $\hat{I}_t$ as follows:

$$\hat{I}_t = \tilde{I}_t + f_{EC}([E_t, \tilde{I}_t]), \tag{6}$$

where $f_{EC}(\cdot)$ denotes the forwarding process of the EC module and $[\cdot]$ denotes the concatenation operation. $\hat{I}_t$ is then fed into the video codecs as the VRF to facilitate the coding process of tth frame.

This is the basic framework of our ECAR-Net, and the overall pipeline to generate the VRF is shown in Figure 2. While the physical meanings of the long-term dynamics ε_t and error map E_t are configuration-adaptive to adapt to different coding scenarios, which will be described in the following sections.

3.3 ECAR-Net for LDB Configuration

When coding under the LDB configuration, the frame with a smaller POC is coded first, which means the coding order is totally in the temporal order. As a result, it becomes more tractable to capture the successive long-term motion. We design ECAR-Net for LDB configuration from two aspects:

First, we capture and update the long-term temporal dynamics iteratively. The long-term dynamics are regarded as a kind of memory and are initialized to zero and updated after each prediction, while the updated one is used for the next prediction as shown in Figure 4. The key role of this consecutive propagation process is the Dynamics Updating module as shown in Figure 3, which consists of a Convolution LSTM [41] subnet and ε_t is the hidden state of the ConvLSTM. The tth predicted frame of our ECAR-Net $\hat{I}_t$ is fed into the video codecs as a reference frame. After the video codecs finish coding the tth frame with the reconstructed frame denoted as I_t, the residue between $\hat{I}_t$ and I_t is calculated. This residue denoted as R_t reflects the prediction error at the time

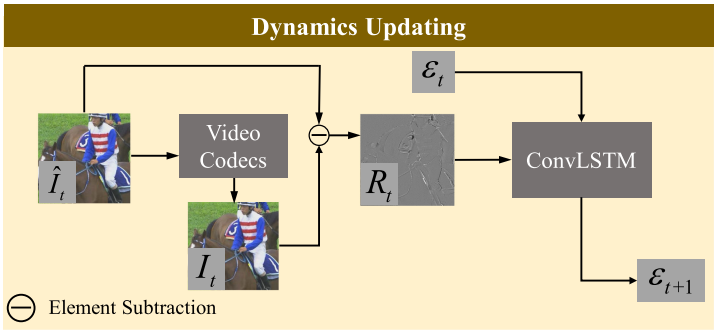


Fig. 3. The Dynamic Updating module under the LDB configuration. The prediction result of our model $\hat{I}_t$ is fed into the video codecs as a reference frame for the tth frame. Then, the residues between $\hat{I}_t$ and the reconstructed tth frame I_t are fed into the ConvLSTM with hidden state ε_t. After that, the hidden state is updated to ε_{t+1}, which is the long-term dynamics of the next iteration.

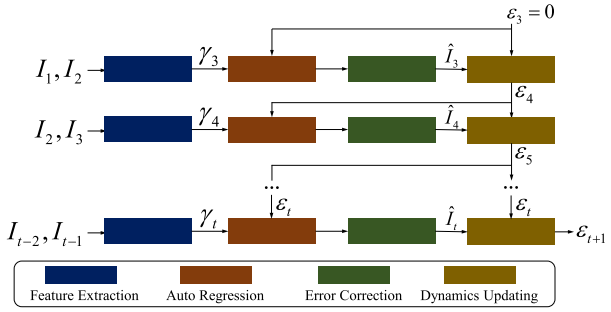


Fig. 4. The coding process under the LDB configuration. Before coding the tth frame, the previous frames I_{t-1} and I_{t-2}, together with the long-term dynamics at the current timestep ε_t, are taken as the input to regress the current frame. The output $\hat{I}_t$ is used as the VRF to facilitate the coding of the current frame. The long-term dynamics ε_t are then be updated and propagated to the next timestep ε_{t+1}. Due to the lack of input frames, this process is not performed for the first two frames so the current frame index of the first row is 3.

step t. Then current long-term temporal dynamics ε_t are updated to ε_{t+1} with R_t as follows:

$$\varepsilon_{t+1} = f_{LSTM}(R_t, \varepsilon_t), \tag{7}$$

where $f_{LSTM}(\cdot)$ is the forwarding process of the ConvLSTM. Then, the updated ε_{t+1} is the long-term temporal dynamics to predict the $(i+1)$-th frame. The long-term dynamics will be updated and propagated as described above until all frames are coded. In this way, we do not have to introduce more input frames at each timestep but have already succeeded to capture motion information of a long time-span during the AR process.

Second, we make use of the past prediction error to further enhance the AR result $\widetilde{I}_t$ in the EC module. As discussed in Section 3.2, the traditional MA model has to take a large number of previous prediction errors as input, which increases the calculation and storage burden. Considering that the most nearby adjacent frames are usually most correlated to the current frame, we use the prediction error of the last predicted frame R_{t-1} as the error map E_t to achieve the balance between correction effectiveness and complexity.

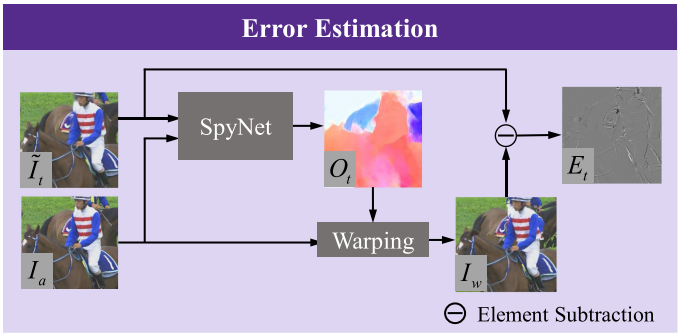


Fig. 5. The Error Estimation module under the RA configuration. The anchor frame I_a is first aligned to the regression result $\widetilde{I}_t$ with the optical flow estimated by the SpyNet. The residue is viewed as the error map E_t to guide the EC process.

3.4 ECAR-Net for RA Configuration

When coding under the RA configuration, all frames of a video are divided into different **group of pictures** (**GOP**). The first frame of a GOP is the key frame and others are non-key frames. During the coding process, the frames of an earlier GOP are coded before the ones of a later GOP. The frames of the same GOP are further divided into different coding layers. It is the depth of the coding layer of each frame that decides its coding order and the frames of a deeper layer are coded later than the ones of a shallower layer. This coding scenarios result in issues when applying the ECAR-Net for LDB configuration directly to the RA configuration, which incurs the further adjustment from two aspects:

First, for the AR process, we use different input frames and change the definition of the long-term temporal dynamics. It is possible that the tth frame has been coded before some frames with POCs smaller than t as they might belong to a deeper coding layer, so it is not feasible to always use the reconstructed frames with smaller POCs to predict the current frame. However, the frames of a shallower coding layer have been coded before the current frame due to the hierarchy coding structure, which can be taken for the VRF generation in the frame interpolation manner like MQ-FKCNN [18]. Specifically, the input frames I_{t-1} and I_{t-2} in Figure 1 are replaced by I_{t-k} and I_{t+k}, which are in the shallower coding layers. k is the temporal interval between the input frames and the current to-be-coded frame, which is formulated as follows:

$$k = 2^{\tau(I_t)}, \tag{8}$$

where $\tau(I_t)$ stands for the difference between the coding depth of I_t and the maximum coding depth.

Besides, the iteratively updated long-term temporal dynamics under the LDB configuration cannot be applied under the RA configuration due to the coding order issue, so we design a flow-based scheme to achieve the goal of long-term motion modeling. First, we introduce an anchor frame I_a, which is also previously coded and temporally farther from the to-be-coded frame than the input frames I_{t-k} and I_{t+k}. Note that many candidate frames can be chosen as the anchor frame, which will be discussed later. Then, we use the SpyNet [42] to estimate the optical flow from I_a to I_{t-k} and I_{t+k}, denoted as O_{t-k} and O_{t+k}. The concatenation of O_{t-k} and O_{t+k} is viewed as the long-term temporal dynamics ε_t under the RA configuration to obtain the AR result $\widetilde{I}_t$ together with I_{t-k} and I_{t+k} as shown in Figure 1.

Similarly, we further enhance $\widetilde{I}_t$ with the EC module. The error map E_t is re-defined and calculated with an Error Estimation module as shown in Figure 5 for the RA configuration. This

Table 1. Intermediate Feature Dimensions of Feature Extraction Module

Layer	C1	C2	C3	C4	C5	DC5	DC4	DC3	DC2	DC1
Size	$\frac{H}{2},\frac{W}{2}$	$\frac{H}{4},\frac{W}{4}$	$\frac{H}{8},\frac{W}{8}$	$\frac{H}{16},\frac{W}{16}$	$\frac{H}{32},\frac{W}{32}$	$\frac{H}{16},\frac{W}{16}$	$\frac{H}{8},\frac{W}{8}$	$\frac{H}{4},\frac{W}{4}$	$\frac{H}{2},\frac{W}{2}$	H,W
Channel	32	64	128	256	512	512	256	128	64	32

estimation process can be formulated as follows:

$$E_t = \widetilde{I_t} - f_{warp}(I_a, O_t), \tag{9}$$

where O_t denotes the optical flow between $\widetilde{I_t}$ and I_a estimated by the SpyNet. $f_{warp}(\cdot)$ denotes the warping operation. By warping I_a to $\widetilde{I_t}$, the pixels of these two images are well aligned. Some regions in $\widetilde{I_t}$ might come across detail loss and visual artifacts due to inaccurate regression. In that case, those regions in E_t have large values to reveal the regression errors, which further guides the EC process. Some visualization will be provided in the experiment part to prove the effectiveness of the EC process under the guidance of the Error Estimation module. After calculating E_t, the final output $\hat{I}_t$ is obtained as shown in Equation (6) with the EC Module, which is fed into video codecs as an VRF to facilitate the coding process.

3.5 Architecture of the Network

Figure 1 shows the overall architecture of our network, which consists of three main modules.

Feature Extraction Module. We apply an encoder-decoder structure to extract short-term redundancy γ_t from the two input frames as shown in Figure 1. We introduce continuous down-sampling and up-sampling operations for larger receptive fields and skip connections from the encoder to the decoder to bypass information at different levels to make this module more aware of subtle motions. The extracted feature γ_t encodes main spatial and temporal structures of temporally nearby adjacent frames. Table 1 shows the detailed output shape of each layer, where H and W denote the height and width of the input image. Two input images are concatenated and then fed into the network. C_i and stand for the convolution block of the encoder and DC_i is the deconvolution block of the decoder, which consist of three convolution layers with 3×3 kernel size followed by **Rectified Linear Unit** (**ReLU**) activation function. The resolution is downsampled with an average pooling by C_i and upsampled with a bilinear interpolation by DC_i.

Auto Regression Module. The AR Module takes both the long-term dynamics ε_t and the extracted feature γ_t as the input to regress the current frame. It contains two submodules: **Coefficient Generation** (**CG**) module and **Attention Generation** (**AG**) module. The structures of the two modules are described in detail in Figure 6. In the CG module, γ_t and ε_t are concatenated to merge the short-term and long-term temporal information to generate the regression coefficients. In the AG module, the attention map is calculated with the short-term temporal redundancy γ_t, followed by the tanh activation function to constrain the data range to $[-1, 1]$. The subsequent summation and multiplication with constant coefficients make the output attention maps M_{t-1} and M_{t-2} sum up to an all-one map.

Error Correction Module. This module plays the role to improve the quality of the AR regression result under the guidance of the error map E_t. We use the MIRNet [43], which achieves satisfactory performance in image enhancement as its basic structure. The original network structure only takes to-be-enhanced images as input of three channels, while we modify the first network layer to accept input of six channels, including the AR regression result and the error map. The channel number of the intermediate features is set to 32, which is half of the original configuration, to further decrease the calculation burden while still achieving significant performance.

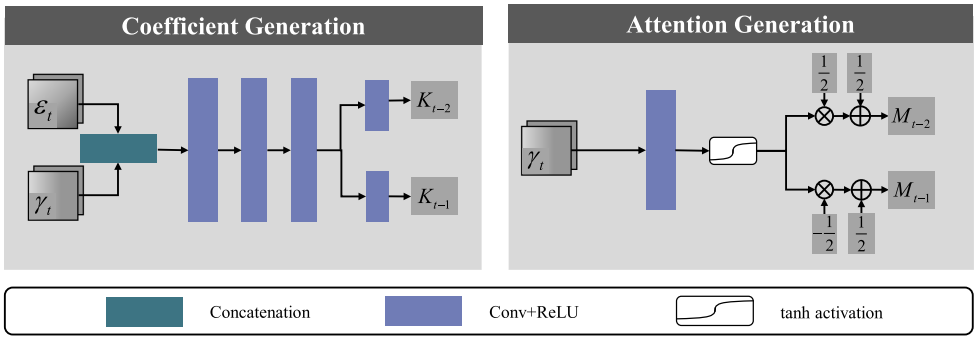


Fig. 6. Architecture of the CG module and AG module. For the Coefficient Generation module, both the short-term temporal redundancy γ_t and long-term temporal dynamics ϵ_t are taken as inputs to generate the regression coefficients. In this way, the temporal information of a long time-span can be traced with only a samll number of input frames. For the Attention Generation module, the attention map is generated with the short-term temporal redundancy γ_t. The tanh function is used to constrain the data range of the attention map. The following summation and multiplication with constant coefficients make the summation of M_{t-2} and M_{t-1} equal to an all-one map.

These three modules make up our final model. Under the architecture details described above, the number of parameters is 249.6 M, which takes 95.3 MB of disk storage. For an input with a size of 128×128, the number of the floating point operations during the network inference stage is 9.79 G. More analysis on computational complexity is provided in the experimental section.

3.6 Integration Into HEVC

Encoding Stage: There are two reference frame lists denoted as *List*0 and *List*1 where previous encoded frames are placed as reference frames under the LDB and RA configurations. Before the encoding of a to-be-coded frame, the corresponding input frames of our model are read from the reference frame lists, and fed into ECAR-Net to generate the VRF $\hat{I}_t$. Then, the farthest reference frame from the to-be-coded frame in each reference frame list is replaced with $\hat{I}_t$ to begin the encoding process. When the encoding process ends, $\hat{I}_t$ is removed from the reference frame list and the previously replaced reference frame is put back to the reference frame list, which means the reference frame list is restored to the original state.

Decoding Stage: The overall process for VRF generation is consistent with the one on the encoding side. Before the decoding of a frame, the reference frame list is filled with decoded frames. The VRF will be generated with the input frames, which are the same as the ones used on the encoding side and then replace the farthest reference frame in the reference frame list to begin the decoding process. When the decoding process ends, the reference frame list is restored in the way same as the encoding side.

The key target on the decoding side is to make sure that the same VRF as the one on the encoding side should be provided. As described above, the VRF only exists in the reference frame list when the encoding process is ongoing and will be removed by the previously replaced reference frame to restore *List*0 and *List*1 to the original state when the encoding process ends. In this way, the state of *List*0 and *List*1 on the decoding side is the same as the one on the encoding stage. Thus, the input frames fed into our model during the decoding stage is the same as the ones in the encoding stage, which means the generated VRFs in the encoding stage and the decoding stage are also the same to achieve the consistency of VRFs.

Integration into VVC: The overall implementation details are the same with HEVC. However, for the RA configuration, we only generate a VRF for the frame in the deepest coding depth. This

is because the interval of the input frames is larger for the frames in a shallower coding depth. As a result, the generated VRF is of lower quality. We find that the generated VRF with the input frames of a larger interval brings in little improvement to the coding performance of VVC, so we choose only to generate VRFs for the frame in the deepest coding depth.

Other Details: Under the LDB configuration, the generated frame $\hat{I}_t$ is kept until this frame has been encoded and then the generation residue R_t is calculated to update the hidden state of the ConvLSTM module. Under the RA configuration, for most frames, there are two forward reference frames in $List0$ and two backward reference frames in $List1$. We choose the frame of each list that is closer to the to-be-coded frame as the input frame of ECAR-Net. Due to the hierarchical coding structure, the two chosen frames take the to-be-coded frame as the center of symmetry, so we denote these two frames as I_{t-k} and I_{t+k}. As for the selection strategy of the anchor frame, there are many possible ways. In our implementation, we design a two-candidates selection strategy as follows:

$$I_a = \begin{cases} I_{t-2k}, & if\ I_{t-2k}\ is\ coded, \\ I_{key}, & else. \end{cases} \tag{10}$$

In this strategy, I_{t-2k} is chosen as the anchor if it has been coded. Else we directly use the key frame of the current GOP as the anchor frame. We also explore other strategies in the experiment part and find that this strategy can bring in a good performance so we finally choose it.

3.7 Training Details

3.7.1 Training Data and Configuration. We choose the Vimeo-90K dataset [44] as our training data. The dataset has 89,800 clips with a resolution of 448×256. Each clip has seven consecutive frames. We use 87,902 clips as the training data and the rest as the validation data. In order to simulate the quality degradation due to quantization, we compress all the data with HEVC to simulate the quality degradation caused by lossy compression, and QP values range from 1 to 51. For the frames of each clip, the degraded frames are denoted as I_1 to I_7, while the original frames denoted as $\overline{I}_1$ to $\overline{I}_7$ are used as the ground truth. Note that in this section, I_t stands for the t th frame in each 7-frame clip. During the training stage, every image is randomly cropped into a 128×128 patch while randomly flipped both horizontally and vertically for data augmentation. The network is implemented in Pytorch and AdaMax [45] is used as the optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is set to 16 and the learning rate is firstly set to 10^{-3} while turned down gradually until convergence. We choose the **sum of absolute transformed difference** (**SATD**) loss function proposed in [10, 18] between the generation result and the ground truth as the loss function. The training process consists of two stages. In the first stage, we only train the Feature Extraction Module and the AR Module with the SATD loss as follows:

$$L_1 = SATD(\tilde{I}_t, \overline{I}_t), \tag{11}$$

where $\tilde{I}_t$ is the output of the AR Module and $\overline{I}_t$ is the corresponding Ground Truth. After the first stage, the Feature Extraction Module and the AR Module have the ability to generate reasonable VRF. In the second stage, we add the EC Module to correct $\tilde{I}_t$ to obtain the final VRF $\hat{I}_t$ of our full model. We introduce another SATD loss function for the optimization of the EC Module as follows:

$$L_2 = SATD(\hat{I}_t, \overline{I}_t). \tag{12}$$

The whole network is jointly trained with the total loss function as follows:

$$L_{total} = L_1 + \lambda L_2, \tag{13}$$

where λ is the weight to balance two items, which is set to 0.5 in our implementation. For the optical estimation network SpyNet, we directly use the pretrained weights provided by the author

and the weights are fixed during the training stage because satisfactory estimation results have already been obtained by SpyNet with the pretrained weights. While for MIRNet, which is originally designed for image enhancement tasks like image denoising, is re-trained in the second training stage described above to make it capable for the goal of VRF correction.

3.7.2 Training Model for LDB Configuration. In the coding process, only the reconstructed frames instead of the lossless frames are available, which will be used for calculating the generated residues. In order to simulate the quality degradation of the reconstructed frames, in the training stage, we calculate the generated residues with the degraded frames rather than the ground truth to make the training process closer to the real coding process. For each clip, we generate five consecutive results $\hat{I}_3$–$\hat{I}_7$. The generation residues, denoted by R_3–R_7, are used to update the memory according to the way described above.

3.7.3 Training Model for RA Configuration. Different from the training method for LDB configuration, we design a random-interval training scheme for RA configuration. For each clip, we randomly select I_{t-1} and I_{t+1} as the input to predict $\overline{I}_t$, where $2 <= (t \pm 1) <= 7$. I_1 is chosen as the anchor frame. Training our model with different t can make it adapt to different anchor frames.

4 EXPERIMENTAL RESULTS

4.1 Overall Performance

We test the proposed method on HEVC reference software HM-16.20 with the common test conditions [46]. The QP values are set to 22, 27, 32, and 37 but only one model is trained for all QPs. BD-rate [47] is chosen to measure the rate-distortion. Table 2 shows the overall performance of our proposed method for classes A–E. Our method has obtained on average 5.0% and 6.6% BD-rate saving under LDB and RA configuration, respectively. For the testing sequence *BQSquare*, up to 15.9% BD-rate saving can be obtained for the luma component under the RA configuration. For further verification, we show some rate-distortion curves of four test sequences in Figure 7.

When coding under the RA configuration, the GOP size can be set to different values. In order to prove the generality of our method, we further test our method under the RA configuration with a GOP size of 8. As shown in Table 3, significant gains can still be achieved with our method.

4.2 Comparison with Existing Methods

We compare our ECAR-Net with state-of-the-art methods under the LDB and RA configurations to show the superiority of our model.

For the LDB configuration, the **Video Coding oriented LAplacian Pyramid of Generative Adversarial Networks** (**VC-LAPGAN**) [22] and the **Deep Frame Prediction** (**DFP**) framework [17] are compared. VC-LAPGAN introduces an existing video prediction model for the generation of VRFs. DFP uses adaptive convolutions to generate the virtual frame. But only the two closest reference frames are taken as input, while the temporal information beyond the input is not taken into consideration, which makes it fail to model the long-term dynamics. As Table 4 shows, our method outperforms VC-LAPGAN and DFP on all classes with 2.8% and 2.2% more BD-rate reduction obtained, respectively.

For the RA configuration, the DFP and MQ-FKCNN [18] are compared with our method. MQ-FKCNN improves the interpolation network proposed in [16] by introducing multi-domain hierarchical constraints. However, it also ignores the long-term motion modeling. As Table 5 shows, our method outperforms DFP and MQ-FKCNN on all classes with 4.0% and 2.2% more BD-rate reduction, respectively.

Table 2. BD-rate Reduction of the Proposed Method Compared to HEVC

Class	Sequence	LDB			RA		
		Y	U	V	Y	U	V
Class A	Traffic	-	-	-	−7.0%	−10.1%	−13.0%
	PeopleOnStreet	-	-	-	−11.4%	−24.2%	−18.5%
	Nebuta	-	-	-	−3.2%	−10.1%	−1.0%
	SteamLocomotive	-	-	-	−5.6%	−1.7%	5.8%
	Average	-	-	-	−6.8%	−11.5%	−6.7%
Class B	Kimono	−5.4%	−14.8%	−5.4%	−5.1%	−15.4%	−7.3%
	BQTerrace	−2.1%	−13.3%	−13.7%	−3.4%	−14.8%	−10.6%
	BasketballDrive	−2.0%	−8.7%	−10.1%	−3.4%	−11.2%	−12.5%
	ParkScene	−3.2%	−13.3%	−6.8%	−5.4%	−14.9%	−7.1%
	Cactus	−7.1%	−19.4%	−12.1%	−6.8%	−14.5%	−10.6%
	Average	−4.0%	−13.9%	−9.6%	−4.8%	−14.2%	−9.6%
Class C	BasketballDrill	−3.2%	−20.8%	−20.3%	−4.8%	−17.6%	−20.5%
	BQMall	−6.4%	−15.3%	−16.1%	−8.8%	−15.1%	−16.9%
	PartyScene	−4.4%	−12.7%	−15.9%	−7.0%	−12.7%	−16.6%
	RaceHorsesC	−0.7%	−2.7%	−3.9%	−2.2%	−6.5%	−8.6%
	Average	−3.7%	−12.9%	−14.1%	−5.7%	−13.0%	−15.6%
Class D	BasketballPass	−5.8%	−17.5%	−15.7%	−8.8%	−21.5%	−21.8%
	BlowingBubbles	−4.6%	−11.8%	−15.5%	−7.6%	−10.3%	−15.6%
	BQSquare	−6.3%	−6.1%	−17.7%	−15.9%	−6.4%	−14.4%
	RaceHorses	−1.6%	−8.6%	−10.0%	−5.6%	−15.2%	−17.8%
	Average	−4.6%	−11.0%	−14.7%	−9.5%	−13.3%	−17.4%
Class E	FourPeople	−12.1%	−11.8%	−5.7%	-	-	-
	Johnny	−6.4%	1.8%	−0.9%	-	-	-
	KristenAndSara	−8.7%	−6.4%	−4.1%	-	-	-
	Average	−9.1%	−5.4%	−3.5%	-	-	-
All Sequences	Overall	−5.0%	−11.3%	−10.9%	−6.6%	−13.1%	−12.2%

4.3 Computational Complexity

The computational complexity under the RA configuration of the proposed method is shown in Table 6. The network forwarding is performed with a GTX 1080Ti GPU. The overall encoding time complexity is 184% and decoding time complexity is 7,801% compared with HEVC. We further compared our method with MQ-FKCNN [18] in the computational complexity. As shown in Table 6, our method takes less time, especially at the decoding end which is more critical for practical applications. In the future, our method can be further optimized with network acceleration techniques like model prunning and hardware acceleration.

First, we compare our method with the end-to-end DNN-based video compression method DVC [49] in terms of the BD-Rate Reduction and Relative Complexity of the decoder side simultaneously. As Figure 8(b) shows, the relative complexity of our method is comparable with DVC, while our method achieves more BD-Rate saving on Class B. Note that only the computational complexity of DVC on Class B is available so the comparison results on other classes are not shown.

Furthermore, we compare with two methods, which also explore to generate VRFs to enhance the inter prediction. Figure 8(a) shows the average BD-Rate Reduction and Relative Complexity results on all classes. It can be observed that our method achieves the highest performance with

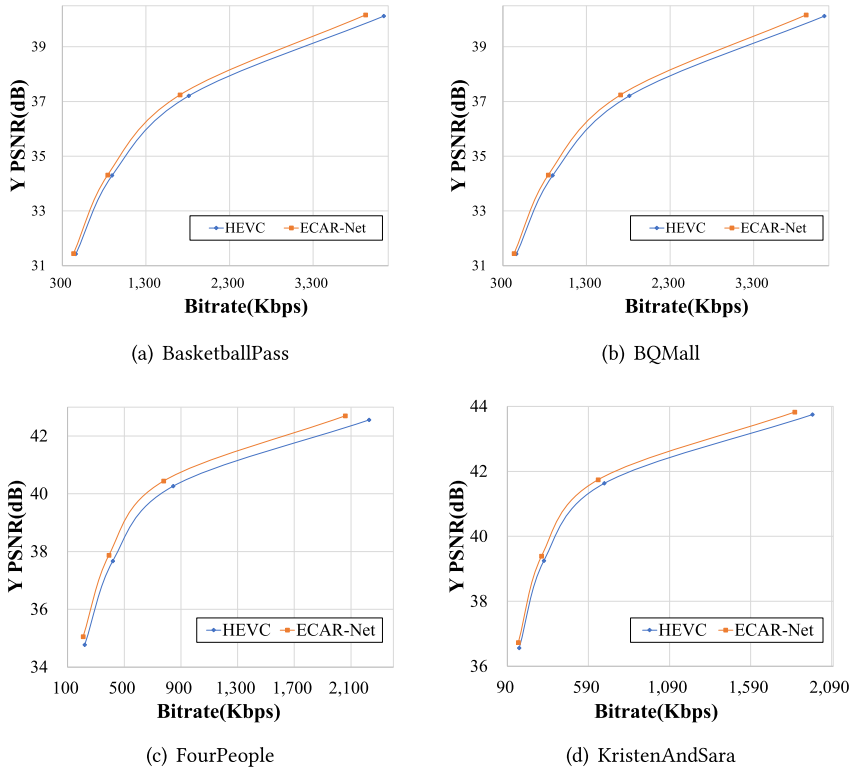


Fig. 7. R–D curves of the test sequences *BasketballPass, BQMall, FourPeople, and KristenAndSara* for the luma component under the LDB configuration.

Table 3. BD-rate Reduciton under the RA Configuration with a GOP Size of 8

Class	Y	U	V
Class A	−5.9%	−17.3%	−5.6%
Class B	−4.7%	−12.7%	−7.7%
Class C	−5.6%	−11.5%	−14.1%
Class D	−9.4%	−11.6%	−14.4%
All Sequences	−6.3%	−13.2%	−10.3%

the lowest relative complexity. Besides, the author of MQ-FKCNN provides the detailed Relative Complexity of each class. As Figure 8(b) shows, our method still achieves higher performance with lower relative complexity in every class.

4.4 Comparison with VVC

We integrate ECAR-Net into the recent VVC reference software VTM (version 15.0) to evaluate the coding efficiency of our method following the VVC common test condition under the RA configuration. The QP values are set to 22, 27, 32, and 37. We use the standard coding configurations provided by the VTM reference software. Under these configurations, the advanced tools in VVC for inter-prediction like the Affine Motion Compensated Prediction and the Bi-Directional Optical

Table 4. BD-rate Reduction Comparison between Existing Methods and ECAR-Net under the LDB Configuration

Class	VC-LAPGAN [22]	DFP [17]	MAAR-Net [48]	Ours
Class B	−2.7%	−2.1%	−3.1%	**−4.0%**
Class C	−1.3%	−2.2%	−2.6%	**−3.7%**
Class D	−1.4%	−2.2%	−3.2%	**−4.6%**
Class E	−3.4%	−5.8%	−8.2%	**−9.1%**
All Sequences	−2.2%	−2.8%	−4.0%	**−5.0%**

The best results are highlighted in bold.

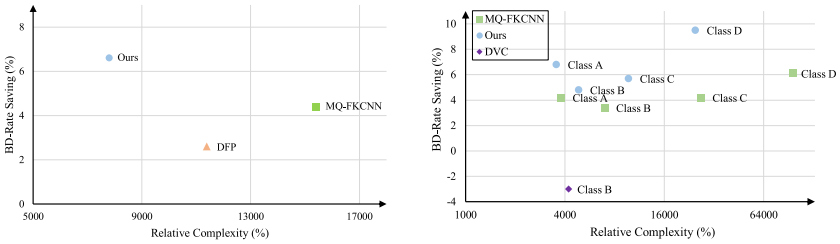
Table 5. BD-rate Reduction Comparison between Existing Methods and ECAR-Net under the RA Configuration

Class	DFP [17]	MQ-FKCNN [18]	Ours
Class A	−3.3%	−4.2%	**−6.8%**
Class B	−1.8%	−3.4%	**−4.8%**
Class C	−2.7%	−4.2%	**−5.7%**
Class D	−1.7%	−6.1%	**−9.5%**
All Sequences	−2.6%	−4.4%	**−6.6%**

The best results are highlighted in bold.

Table 6. Computational Complexity of the Proposed Method and MQ-FKCNN [18]

Class	MQ-FKCNN [18]		Ours	
	Enc (%)	Dec (%)	Enc (%)	Dec (%)
Class A	184%	3,797%	172%	3,550%
Class B	180%	7,002%	177%	4,865%
Class C	232%	26,820%	183%	9,745%
Class D	402%	96,649%	206%	24,768%
All	232%	15,422%	184%	7,801%



(a) All Class Comparison (b) Per Class Comparison

Fig. 8. BD-Rate Reduction and Relative Complexity of different methods.

Flow are activated. As Table 7 shows, our method achieves on average 1.54%, 3.67%, and 4.12% BD-rate reduction on Y, U, and V, respectively.

We further compare the BD-rate reduction with existing methods to prove the superiority of our method. We choose MQ-FKCNN as the compared method as most existing methods do not provide the BD-rate reduction result anchored on VVC. MQ-FKCNN is implemented on VTM 3.0.

Table 7. BD-rate Reduction of the Proposed Method Anchored on VTM-15.0

Class	BD-Rate Reduction		
	Y	U	V
Class A1	−0.74%	−1.85%	−2.35%
Class A2	−0.70%	−3.23%	−2.52%
Class B	−0.81%	−2.88%	−3.33%
Class C	−2.13%	−4.76%	−4.85%
Class D	−3.08%	−5.24%	−6.90%
All	−1.54%	−3.67%	−4.12%

Table 8. BD-rate Reduction Comparison with MQ-FKCNN Anchored on VTM-3.0

Class	MQ-FKCNN [18]			Ours		
	Y	U	V	Y	U	V
Class A1	−0.17%	−0.89%	−1.10%	−1.01%	−2.30%	−4.04%
Class A2	−0.59%	−2.71%	−2.31%	−0.97%	−7.02%	−5.20%
Class B	−0.31%	−1.79%	−1.69%	−1.02%	−6.24%	−6.17%
Class C	−2.16%	−3.70%	−3.64 %	−2.23%	−5.39%	−5.31%
Class D	−3.08%	−4.62%	−5.03%	−3.30%	−5.48%	−7.23%
All	−1.30%	−2.79%	−2.81%	−1.74%	−5.40%	−5.72%

Table 9. Ablation Study under the LDB Configuration

Class	w/o EC	w/o ε_t	ECAR-Net
Class B	−2.8%	−3.8%	−4.0%
Class C	−2.8%	−3.4%	−3.7%
Class D	−2.9%	−3.8%	−4.6%
Class E	−5.6%	−8.9%	−9.1%
All	−3.4%	−4.7%	−5.0%

Table 10. Ablation Study under the RA Configuration

Class	w/o EC	w/o ε_t	ECAR-Net
Class A	−5.2%	−6.2%	−6.8%
Class B	−3.9%	−3.9%	−4.8%
Class C	−4.3%	−4.6%	−5.7%
Class D	−6.0%	−7.3%	−9.5%
All	−4.8%	−5.4%	−6.6%

For a fair comparison, we also implement our method on VTM 3.0. As Table 8 shows, our method outperforms MQ-FKCNN on all classes with 0.44%, 2.61%, and 2.91% more BD-rate reduction on Y, U, and V, respectively.

4.5 Ablation Study

4.5.1 Verification of the EC Module. We compare the performance gain between the plain AR model and full ECAR-Net equipped with the EC module to verify its efficiency. As shown in Tables 9 and 10, the model with the EC module can obtain 1.6% and 1.8% BD-rate reduction,

Table 11. Ratios of CUs that Choose VRF for Inter Prediction under Different QPs

Class	22	27	32	37
Class A	52.5%	53.5%	54.4%	54.7%
Class B	36.7%	34.8%	39.5%	41.2%
Class C	31.5%	33.7%	38.3%	43.1%
Class D	47.6%	53.5%	58.3%	59.4%
All Sequences	41.7%	43.2%	47.1%	49.1%

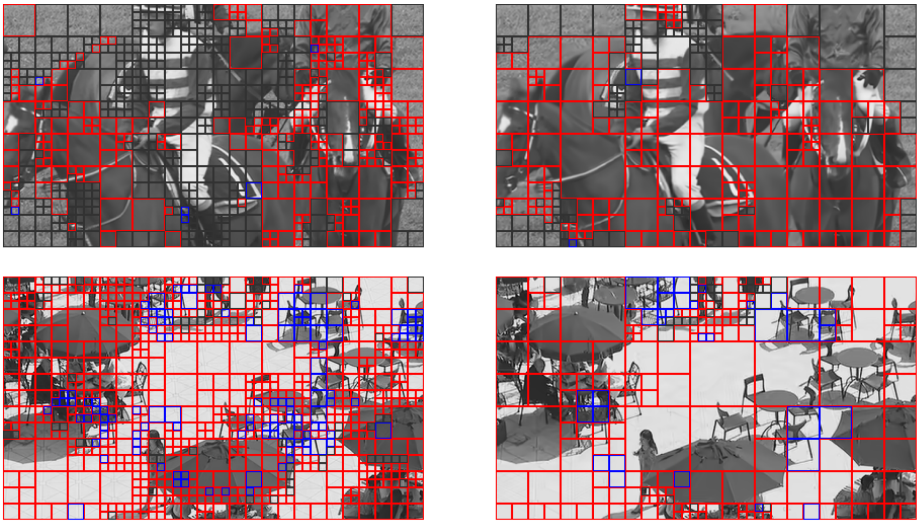


Fig. 9. Visualization of the CU Prediction Mode. Red indicates CUs that choose our VRF as reference. Top: 3rd frame of RaceHorses. Bottom: 12th frame of BQSquare. Left: QP is 22. Right: QP is 32.

respectively. Especially on Class E under the LDB configuration, the proposed module obtains up to 3.5% BD-rate reduction.

4.5.2 Verification of the Long-term Dynamics Injection. For the purpose of the verification of our model to capture long-term dynamics, we reset the long-term dynamics to zero before each prediction process. In this way, the perception of the long-term motion is screened and only short-term temporal information is used to the prediction process. As shown in Tables 9 and 10, the model with the injection of the long-term dynamics can obtain 0.3% and 1.2% BD-rate reduction, respectively.

4.6 Visualization of the VRF Selection Ratio

We further calculate the ratio of the CUs that choose the VRF generated by ECAR-Net for inter prediction. As Table 11 shows, a considerable number of CUs adopt the generated VRF for inter prediction. Figure 9 presents visual results of the prediction mode result of the **Coding Unit** (**CU**) under different QPs. Red indicates that the CU chooses our generated VRF as the reference frame while Black indicates that the CU chooses existing reference frames to perform inter prediction. Blue indicates that the CU is coded under the intra prediction mode. It can be observed that our VRFs are chosen in large numbers, especially in the regions with motions and object edges, while

(a) Auto Regression Result $\tilde{I}_t$ (b) Aligned Anchor Frame I_w (c) Error Corrected Result $\hat{I}_t$ (d) Ground Truth

Fig. 10. Visualization results of the EC process under the RA configuration. The raw AR result $\tilde{I}_t$ suffers from visual artifacts. The aligned anchor frame I_w plays the role as a reference to obtain the EC result $\hat{I}_t$, which is of better quality compared with $\tilde{I}_t$.

Table 12. Comparisons of Different Anchor Frame Selection Strategies

Sequence	$Anchor_C$	$Anchor_K$	$Anchor_T$
BasketballPass	−5.3%	−7.5%	−8.8%
BQSquare	−1.1%	−12.3%	−15.9%
BlowingBubbles	−2.9%	−6.7%	−7.6%
RaceHorses	−2.6%	−4.4%	−5.6%
All	−3.0%	−7.7%	−9.5%

existing reference frames are mainly selected in the background, which proves that our VRF is preferred to deal with the regions with complex motion patterns.

4.7 Visualization of the Error Correction Module

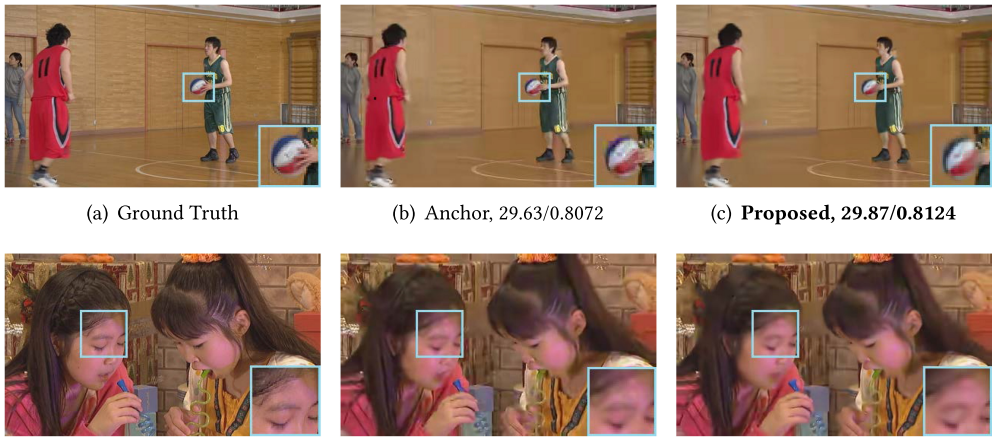
Figure 10 visualizes the efficiency of the EC Module under the RA configuration. The output of the AR stage $\tilde{I}_t$ suffers from visual artifacts as shown in Figure 10(a). Then in the Error Estimation module, the aligned anchor frame I_w works as a reference to calculate the error map, which reveals the regions with detail loss and visual artifacts. Finally, $\tilde{I}_t$ is refined under the guidance of the error map to obtain the final result $\hat{I}_t$ with the EC module. It can be observed that $\hat{I}_t$ is of better quality with fewer visual artifacts compared with $\tilde{I}_t$.

4.8 Discussion of the Anchor Frame

For the RA configuration, the way to choose the anchor frame is not fixed. We design three different selection strategies to choose the anchor frame:

— **Closest Strategy.** The input frame I_{t-k} is chosen as the anchor frame. Thus, the anchor frame is close enough to the current frame. We denote this strategy as $Anchor_C$.
— **Key Frame Strategy.** The key frame of the current GOP is always chosen as the anchor frame, this strategy is denoted as $Anchor_K$.
— **Two-Candidates Strategy.** This strategy is the one we described in Section 3.6, denoted as $Anchor_T$.

These strategies are tested on the Class D sequence, and the BD-Rate reduction results are shown in Table 12. It can be observed that the Two-Candidates Strategy achieves the best performance. For the Closest Strategy, the anchor frame is too close to the current frame and thus brings in limited long-term temporal information. While under the Key Frame Strategy, the temporal interval between the anchor frame and the current frame can be too large, especially when the current frame is too far from the beginning of the current GOP, which results in the anchor frame of a



(a) Ground Truth (b) Anchor, 29.63/0.8072 (c) **Proposed, 29.87/0.8124**

(d) Ground Truth (e) Anchor, 27.07/0.7328 (f) **Proposed, 27.18/0.7341**

Fig. 11. Visual results and PSNR (dB)/SSIM of HEVC anchor and our proposed method.

limited contribution to the prediction process. Based on the discussion above, we finally choose the Two-Candidates Strategy in our implementation.

4.9 Subjective Comparison with HEVC

Figure 11 shows the comparison of the visual quality of the reconstructed frames generated by HEVC anchor and our proposed method. These videos are compressed with the QP 37. It can be observed that the reconstructed frames of our method are quantitatively and qualitatively better. The contents of moving object are difficult for HEVC anchor to handle, which always results in blurry results. On the contrary, our method can deal with this situation better with the aid of the VRFs generated by the ECAR-Net.

5 CONCLUSION

In this article, we propose the ECAR-Net for the inter prediction of video coding. We take the coding scenarios into consideration and modify the traditional ARMA model to make it more suitable for the modeling of video sequences in modern codecs. The long-term temporal dynamics are introduced to the AR process to make use of both correlated reconstructed frames and a long time-span of temporal redundancy. The regression result is further enhanced with the EC module under the guidance of the error map to achieve better reference quality. Experimental results show that our method has obtained significant BD-rate savings on the testing sequences compared with HEVC and VVC.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. 2003. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (2003), 560–576.

[2] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (2012), 1649–1668.

[3] Benjamin Bross, Jianle Chen, Shan Liu, and Ye-Kui Wang. 2020. Versatile video coding (draft 9). In *Proceedings of the Document JVET-R2001*.

[4] X. Zhang, W. Yang, Y. Hu, and J. Liu. 2018. Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 390–394.

[5] Mading Li, Jiaying Liu, Xiaoyan Sun, and Zhiwei Xiong. 2019. Image/video restoration via multiplanar autoregressive model and low-rank optimization. *ACM Transactions on Multimedia Computing, Communications, and Applications* 15, 4 (2019), 1–23.

[6] Wenhan Yang, Jiaying Liu, Mading Li, and Zongming Guo. 2016. Isophote-constrained autoregressive model with adaptive window extension for image interpolation. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 5 (2016), 1071–1086.
[7] Mading Li, Jiaying Liu, Jie Ren, and Zongming Guo. 2014. Adaptive general scale interpolation based on weighted autoregressive models. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 2 (2014), 200–211.
[8] Xin Jin, Zhibo Chen, Sen Liu, and Wei Zhou. 2018. Augmented coarse-to-fine video frame synthesis with semantic loss. In *Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision*. Springer, 439–452.
[9] Yueyu Hu, Wenhan Yang, Sifeng Xia, Wen-Huang Cheng, and Jiaying Liu. 2018. Enhanced intra prediction with recurrent neural network in video coding. In *Proceedings of the Data Compression Conference*. 413–413.
[10] Y. Hu, W. Yang, M. Li, and J. Liu. 2019. Progressive spatial recurrent neural network for intra prediction. *IEEE Transactions on Multimedia* 21, 12 (2019), 3024–3037.
[11] Y. Hu, W. Yang, S. Xia, and J. Liu. 2018. Optimized spatial recurrent network for intra prediction in video coding. In *Proceedings of the IEEE Visual Communications and Image Processing*. IEEE, 1–4.
[12] Sifeng Xia, Wenhan Yang, Yueyu Hu, Siwei Ma, and Jiaying Liu. 2018. A group variational transformation neural network for fractional interpolation of video coding. In *Proceedings of the Data Compression Conference*. 127–136.
[13] Jiaying Liu, Sifeng Xia, Wenhan Yang, Mading Li, and Dong Liu. 2018. One-for-all: Grouped variation network-based fractional interpolation in video coding. *IEEE Transactions on Image Processing* 28, 5 (2018), 2140–2151.
[14] Dezhao Wang, Sifeng Xia, Wenhan Yang, Yueyu Hu, and Jiaying Liu. 2019. Partition tree guided progressive rethinking network for in-loop filtering of HEVC. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2671–2675.
[15] L. Zhao, S. Wang, X. Zhang, S. Wang, and S. Ma. 2019. Enhanced motion-compensated video coding with deep virtual reference frame generation. *IEEE Transactions on Image Processing* 28, 10 (2019), 4832–4844.
[16] S. Niklaus, L. Mai, and F. Liu. 2017. Video frame interpolation via adaptive separable convolution. In *Proceedings of the International Conference on Computer Vision*. 261–270.
[17] Hyomin Choi and Ivan V. Bajić. 2019. Deep frame prediction for video coding. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 7 (2019), 1843–1855.
[18] J. Liu, S. Xia, and W. Yang. 2020. Deep reference generation with multi-domain hierarchical constraints for inter prediction. *IEEE Transactions on Multimedia* 22, 10 (2020), 2497–2510.
[19] T. Laude, F. Haub, and J. Ostermann. 2019. HEVC inter coding using deep recurrent neural networks and artificial reference pictures. In *Proceedings of the Picture Coding Symposium*. 1–5.
[20] William Lotter, Gabriel Kreiman, and David Cox. 2017. Deep predictive coding networks for video prediction and unsupervised learning. In *Proceedings of the International Conference on Learning Representations*.
[21] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *Proceedings of the Advances in Neural Information Processing Systems*.
[22] J. Lin, D. Liu, H. Li, and F. Wu. 2018. Generative adversarial network-based frame extrapolation for video coding. In *Proceedings of the IEEE Visual Communication and Image Processing*. IEEE, 1–4.
[23] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. 2017. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2462–2470.
[24] D. Sun, X. Yang, M. Liu, and J. Kautz. 2018. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 8934–8943.
[25] Ping Hu, Gang Wang, and Yap-Peng Tan. 2018. Recurrent spatial pyramid CNN for optical flow estimation. *IEEE Transactions on Multimedia* 20, 10 (2018), 2814–2823.
[26] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. 2015. Phase-based frame interpolation for video. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 1410–1418.
[27] S. Niklaus, L. Mai, and F. Liu. 2017. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 670–679.
[28] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. 2017. Video frame synthesis using deep voxel flow. In *Proceedings of the International Conference on Computer Vision*. 4463–4471.
[29] Fitsum A. Reda, Guilin Liu, Kevin J. Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. 2018. Sdc-net: Video prediction using spatially-displaced convolution. In *Proceedings of the European Conference on Computer Vision*. 718–733.
[30] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao. 2018. Fully connected network-based intra prediction for image coding. *IEEE Transactions on Image Processing* 27, 7 (2018), 3236–3247.
[31] X. Zhang, W. Yang, Y. Hu, and J. Liu. 2018. DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 390–394.

[32] W. Yang, S. Xia, J. Liu, and Z. Guo. 2018. Reference-guided deep super-resolution via manifold localized external compensation. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 5 (2018), 1270–1283.
[33] J. Kang, S. Kim, and K. M. Lee. 2017. Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 26–30.
[34] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma. 2019. Content-aware convolutional neural network for in-loop filtering in high efficiency video coding. *IEEE Transactions on Image Processing* 28, 7 (2019), 3343–3356.
[35] Y. Wang, X. Fan, C. Jia, D. Zhao, and W. Gao. 2018. Neural network based inter prediction for HEVC. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 1–6.
[36] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao. 2018. Enhanced ctu-level inter prediction with deep frame rate up-conversion for high efficiency video coding. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 206–210.
[37] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[38] Shuai Huo, Dong Liu, Bin Li, Siwei Ma, Feng Wu, and Wen Gao. 2020. Deep network-based frame extrapolation with reference frame alignment. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 3 (2020), 1178–1192.
[39] Jie Ren, Jiaying Liu, Wei Bai, and Zongming Guo. 2011. Similarity modulated block estimation for image interpolation. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 1177–1180.
[40] Mading Li, Jiaying Liu, Zhiwei Xiong, Xiaoyan Sun, and Zongming Guo. 2016. Marlow: A joint multiplanar autoregressive and low-rank approach for image completion. In *Proceedings of the European Conference on Computer Vision*. 819–834.
[41] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proceedings of the Advances in Neural Information Processing Systems*. 802–810.
[42] Ranjan Anurag and Black Michael J. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 4161–4170.
[43] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2020. Learning enriched features for real image restoration and enhancement. In *Proceedings of the European Conference on Computer Vision*. 492–511.
[44] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision* 127, 8 (2019), 1106–1125.
[45] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.
[46] Frank Bossen. 2013. Common test conditions and software reference configurations. Technical Report JCTVC-L1100 (2013).
[47] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. Technical Report VCEG-M33 (2001).
[48] Y. Hu, S. Xia, W. Yang, and J. Liu. 2020. Memory-augmented auto-regressive network for frame recurrent inter prediction. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. IEEE, 1–5.
[49] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 11006–11015.

Received 25 October 2021; revised 19 February 2022; accepted 21 March 2022