End-to-End Transformer for Compressed Video Quality Enhancement

Li Yu©, Wenshuai Chang©, Shiyu Wu, and Moncef Gabbouj©, Fellow, IEEE

Abstract—Convolutional neural networks have achieved excellent results in compressed video quality enhancement task in recent years. State-of-the-art methods explore the spatio-temporal information of adjacent frames mainly by deformable convolution. However, the CNN-based methods can only exploit local information, thus lacking the exploration of global information. Moreover, current methods enhance the video quality at a single scale, ignoring the multi-scale information, which corresponds to information at different receptive fields and is crucial for correlation modeling. Therefore, in this work, we propose a Transformer-based compressed video quality enhancement (TVQE) method, consisting of Transformer based Spatio-Temporal feature Fusion (TSTF) module and Multi-scale Channel-wise Attention based Quality Enhancement (MCQE) module. The proposed TSTF module learns both local and global features for correlation modeling, in which window-based Transformer and the encoder-decoder structure greatly improve the execution efficiency. The proposed MCQE module calculates the multi-scale channel attention, which aggregates the temporal information between channels in the feature map at multiple scales, achieving efficient fusion of inter-frame information. Extensive experiments on the JCT-V7 test sequences show that the proposed method increases PSNR by up to 0.98 dB when QP=37. Meanwhile, the inference speed is improved by up to 9.4%, and the number of Flops is reduced by up to 84.4% compared to competing methods at 720p resolution. Moreover, the proposed method achieves the BD-rate reduction up to 23.04%.

Index Terms—Compressed video quality enhancement, video compression, transformer, deep learning.

I. INTRODUCTION

THE MULTIMEDIA industry is growing rapidly and consumers are expecting videos of higher quality. On the one hand, video is becoming the main form of information carrier in increasing applications, including remote education, telemedicine, live broadcasting, digital TV, video conference and so on. On the other hand, the demand for video resolution is constantly increasing, from 1080p to 2K, 3K, 4K, as well as 8K. The extremely large amount of video data has to be compressed by video compression standards, such as H.264/AVC [1] and H.265/HEVC [2], to fit the available storage and network bandwidth. As the compression ratio increases, the encoder significantly reduces the bit rate while introducing undesirable artifacts that severely degrade the quality of experience (QoE). The introduced artifacts also impair the performance of downstream video-oriented tasks (e.g., action recognition [3], [4], object tracking [5], [6], video understanding [7], [8], [9] and frame interpolation [10], [11]). Therefore, it is necessary to enhance the quality of the compressed video and improve the efficiency of video streaming [12].

Convolutional neural networks (CNNs) have achieved milestones in the task of video quality enhancement (VQE). The CNN-based approaches can generally be classified into two categories: single-frame based methods [13], [14], [15], [16], [17], [18], [19] and multi-frame based methods [20], [21], [22], [23], [24], [25], [26]. The single-frame based video enhancement method is equivalent to image enhancement, which explores the contextual information within the frame/image by CNNs to reduce compression artifacts and improve the visual quality. However, the temporal correlations between adjacent frames in the video are ignored, which severely limits the performance. In multi-frame based methods, the temporal information between adjacent frames are explored. Since there are motions between adjacent frames, the inter-frame information cannot be used directly. Some works use the optical flow to compensate the motion between frames. For example, [20], [21] use dense optical flow for motion compensation. However, the optical flow calculated from compressed video tends to be inaccurate. Thus, some work [23] utilizes deformable convolution (DCN) to capture the dependencies between multiple adjacent frames and the DCN-based approaches have made great progress in this task. Then, [24] proposed the Recursive Fusion (RF) module based on [23], which saves the temporal information of previously enhanced video frames for correlation modeling, implicitly expanding the temporal range and achieving superior results. However, deformable convolutional alignment modules in DCN-based approaches are difficult to train, and its instability in training often leads to offset overflow, which ultimately reduces the efficiency of correlation modeling.

In order to capture the long-range correlations between frames, we propose the Transformer based Spatio-Temporal feature Fusion (TSTF) module, which introduces Vision
Transformer into the VQE task for its strong capability to learn long-range dependencies between image patches and its adaptability to image content. Since the computational complexity of the traditional Vision Transformer [27] grows quadratically with the increase of image resolution, we build our model based on improved Swin Transformer [28] in this work, along with the autoencoder structure. The window-based Transformer and multi-scale encoder-decoder structure with skip connections can improve the inference efficiency and reduce GPU consumption. The proposed method is proved to facilitate the mining of spatio-temporal information, as well as correlation modeling of temporal information among multiple frames. As shown in Fig. 1, the compressed frame 210 is enhanced with the information from frames 207 to 213 (only frame 208 to 212 are drawn for illustration). It can be seen that the green line on the floor (below the athlete’s arm) is totally or partially occluded in frames from 207 to 209, and becomes gradually clear from frame 210 to 213. In order to recover the green line in frame 210, proper correlation among pixels should be modeled. The results show that our method achieves better recovery result in this region, which verifies its effective correlation modeling of temporal information among multiple frames.

On the other hand, a multi-scale channel attention enhancement module is proposed to calculate channel attention at multiple scales, which aggregates the multi-scale temporal information from the output of TSTF. Existing methods overlook the significance of multiscale enhancement, which is essential for capturing and enhancing both local and global features in video frames. In TSTF, the large-scale feature maps have a small receptive field, which can model the spatio-temporal correlation of small motions at pixel level. While the small-scale feature maps have a large receptive field and are capable of modeling spatio-temporal correlations of large motions. The proposed multi-scale channel attention enhancement module can efficiently fuse the multi-scale inter-frame information and finally generates the high-quality video frame.

The main contributions of this paper are summarized as follows:
- A novel network for enhancing compressed videos is proposed, which is entirely based on a Transformer-based architecture.
- A Transformer-based spatio-temporal information exploration module is proposed, which is capable of long-range and adaptive correlation modeling between video frames.
- A multi-scale channel-wise attention based quality enhancement module is designed to integrate information from both large and small motions.

The rest of the paper is organized as follows. In Section II, the deep learning based compressed video enhancement methods and Vision Transformer are reviewed. Section III describes the task of VQE, the proposed TVQE method and the training scheme. Section IV presents the experiments and results. Finally, conclusions are provided in Section V.

II. RELATED WORK

In this section, we first review recent works on deep learning-based quality enhancement of compressed video, including single-frame based methods and multi-frame based methods. Then, a brief overview on Vision Transformer is provided.

A. Single-Frame Based Video Enhancement Method

Single-frame video enhancement task is equivalent to image enhancement. Earlier works [13], [15], [17], [18], [29], [30], [31], [32], [33] mainly focus on the quality enhancement of JPEG compressed images. Specifically, AR-CNN [13] first uses a convolutional neural network for image enhancement,
and learns the nonlinear mapping between the original image and the compressed image with four convolutional layers. Subsequently, several works [15], [16], [17], [34], [35], [36] propose using deeper networks to further improve the performance. With batch normalization and residual learning emerged, DnCNN [17] effectively solves the problem of gradient disappearance in deep image enhancement networks. NLRN [31] and RNAN [33] propose the residual non-local attention mechanism to capture long-range dependencies between pixels. In addition to exploiting the information in the spatial domain, methods such as [14], [18], [32] exploit the relevant information in the frequency domain to further improve the subjective visual quality. In particular, [37], [38], [39], [40], [41] utilize the prior knowledge to improve the enhancement performance. For example, DS-CNN [41] and QE-CNN [40] use different methods to deal with intra-frame coding (e.g., AI) and inter-frame coding (e.g., LDP, LDB, RA).

B. Multi-Frame Based Video Enhancement Method

Multi-frame video enhancement methods utilize the spatio-temporal information of multiple adjacent frames for enhancement. Yang et al. [20] proposed the Multi-Frame Quality Enhancement (MFQE 1.0) method, which first uses SVM to distinguish high and low quality frames, and then uses two adjacent high quality frames to enhance the low quality frame by optical flow guided motion estimation. As an enhanced version of MFQE 1.0, MFQE 2.0 [21] proposes an end-to-end quality enhancement network, which pre-trained a bidirectional long short-term memory (BiLSTM) based model to detect peak quality frame (PQF). The QE-subnet is also advanced by introducing the multi-scale strategy, batch normalization and dense connection. However, the video is compressed, and the compressed video can be severely distorted by various compression artifacts, so the estimated optical flow is often inaccurate and unreliable, resulting in ineffective motion compensation. To this end, Deng et al. proposed a sliding window based method STDF [23], which utilizes deformable convolution to avoid explicit calculation of optical flow. This method innovatively proposed to perform feature alignment of moving objects on input multi-frame images through deformable convolution. Based on STDF, RFDA [24] proposes the recursive fusion (RF) module, which not only utilizes the reference frames within the current time window, but also exploits the temporal information of previously enhanced video frames. By implicitly expanding the time window, RFDA leverages a larger range of temporal information for better spatio-temporal compensation. However, the computational complexity of RF module is huge. STDR [42] proposes a Multi-path Deformable Alignment (MDA) module based on STDF, which integrates the alignment features of different receptive fields to obtain more accurate deformation offsets. Luo et al. [43] proposed a recurrent deformable fusion method. Instead of aligning multiple frames to the target frame simultaneously, it aligns each pair of the target frame and adjacent frame according to the timeline, which uses the compensation information between the adjacent frames more efficiently.

C. Vision Transformers

Transformer is a deep neural network based on self-attention mechanism and parallel processing. Transformer [44] first emerged in the field of NLP. The successful application of Transformer in NLP has made relevant scholars begin to investigate its application in the field of computer vision [27], [45]. Image Transformer [46] was the first to migrate the Transformer architecture to the field of computer vision. Subsequently, Dosovitskiy et al. [27] proposed the Visual Transformer (ViT), and ViT completely replaces the Transformer structure with the convolutional structure to deal with the classification task, and achieves results beyond CNN on extremely large-scale datasets [47], [48], [49], [50]. However, the self-attention mechanism calculates the global similarity, and its computational complexity grows quadratically with the expansion of spatial resolution. To improve operational efficiency, an efficient and effective Vision Transformer called Swin Transformer was proposed in [28]. Based on the shift window mechanism, Swin Transformer achieves state-of-the-art performance in image classification [27], [28], [51], object detection [45], [52], image segmentation [53], [54], video understanding [55], [56], image generation [57] and point clouds processing [58], [59]. Zamir et al. proposed Restomer [60], which computes self-attention across channels rather than spatial dimensions, and its complexity grows linearly with image resolution. Thus, Restomer achieves state-of-the-art performance for high-resolution images restoration.

III. METHODOLOGY

Given a compressed video consisting of T frames \( V = [X_1, X_2, \ldots, X_t, \ldots, X_T] \), where \( X_t \in \mathbb{R}^{H \times W} \) represents the compressed frame at time \( t \) and \( W \) is the height and width of \( X_t \). The task of compressed video enhancement is to generate an enhanced video \( V^e = [X^e_1, X^e_2, \ldots, X^e_t, \ldots, X^e_T] \) from the input compressed video \( V \).

The overall framework of the proposed method is shown in Fig. 2, which consists of two modules: (a) Transformer based Spatio-Temporal feature Fusion (TSF) module and (b) Multi-scale Channel-wise Attention based Quality Enhancement (MCQE) module. The TSF module explores the spatio-temporal information across multiple frames by modeling the correlations among these frames. After TSF, the information between channels in the feature map is further fused by MCQE, and finally generate the residual of the enhanced frame. For each compressed frame \( X_t \), its \( R \) preceding frames and \( R \) succeeding frames are used to exploit correlated temporal information. With the input \( V_t = \{X_{t-R}, \ldots, X_t, \ldots, X_{t+R}\} \), the whole process can be expressed as:

\[
X^m_t = T(V_t; \phi),
\]

\[
I_t = M(X^m_t; \psi),
\]

\[
X^e_t = I_t + X_t,
\]

where \( T(\cdot) \) denotes the process of TSF, and \( M(\cdot) \) denotes the process of MCQE. \( X^m_t \) is the output of TSF, which is a set including four scales features \( (TX^m_t, SX^m_t, MX^m_t, LX^m_t) \). \( I_t \) is the output of MCQE, and \( X^e_t \) is the final output. \( \phi \) and \( \psi \) represent the parameters to be learned in the TSF and MCQE.
modules, respectively. The residual learning is used to improve the training efficiency.

A. Transformer Based Spatio-Temporal Feature Fusion Module

Our proposed TSTF is able to explore wider spatio-temporal information than the DCN-based method. As the DCN-based method uses local offsets to build inter-frame correlations, which only matches critical information within a local range between adjacent frames. While the critical information outside the range is ignored. The proposed TSTF characterizes different scales of motion by using different scales of features within a fixed size window, thanks to the U-net structure. The small-scale features contain global spatio-temporal information, which enables modeling the spatio-temporal correlation of large motions. While the large-scale features are able to capture the spatio-temporal information of small motions at the pixel level. Thus, TSTF is able to capture the spatio-temporal association of both large and small motions between adjacent frames.

The TSTF module is an autoencoder, consisting of the Patch Partition layer, adaptive Swin Transformer Block (ASwin-TB) and Pixel Shuffle layer. First, the target frame and the adjacent reference frames are partitioned into non-overlapping patches by the Patch Partition layer. For the consideration of computing speed, the Patch Partition layer downsamples the features and restores it to the original resolution at the final stage by the Pixel Shuffle layer. Between the Patch Partition layer and the Pixel Shuffle layer, the spatio-temporal information is aggregated with our improved ASwin-TB.

The original Swin Transformer only supports input image of fixed-resolution. For the same window size, input images with different resolution lead to different segmentation. So, when the input image size changes, the overall feature map size and the number of segmented windows will also change, which in turn affects the calculation of masks. Therefore, we propose the improved ASwin-TB, which pads the length and width of the feature map to an integer multiple of the window size to support input image of any resolution. In TSTF, each patch after segmentation is treated as a token and then calculated the spatio-temporal attention. In the encoder, Patch Merging Layer increases the number of channels while features are downsampled, and ASwin-TB further enhances the features. For \( V_t = [X_{t-R}, \ldots, X_t, \ldots, X_{t+R}] \), the whole Encoder process can be expressed as

\[
E_1 = E_{stage1}(V_t), \quad (4)
\]
\[
E_2 = E_{stage2}(E_1), \quad (5)
\]
\[
E_3 = E_{stage3}(E_2), \quad (6)
\]

where \( E_{stage} \) denotes the combination of Patch Merging and ASwin-TB, and 1, 2, 3 represent each stage of the encoder.

Corresponding to the encoder, the decoder uses a Patch Expanding Layer to upsample the deep features. For each scale, the low-level features in the encoder are connected to the high-level features in the decoder through skip connections, to reduce the loss of spatial information caused by down sampling. The whole Encoder process can be expressed as

\[
TX^m_t = Dstage1(E_3) + E_3, \quad (7)
\]
\[
SX^m_t = Dstage2(TX^m_t) + E_2, \quad (8)
\]
\[
MX^m_t = Dstage3(SX^m_t) + E_1, \quad (9)
\]
\[
LX^m_t = UP(MX^m_t) \quad (10)
\]

where \( Dstage \) denotes the combination of Patch Expanding and ASwin-TB, \( UP \) represents the upsampling layer of Pixel Shuffle, and 1, 2, 3 represent each stage of the decoder.

By using TSTF, the compressed frame at time \( t \) can aggregate the temporal information from the adjacent reference frames and generate four different scales of features, from small to large \( (TX^m_t, SX^m_t, MX^m_t, LX^m_t) \) in order.

B. Multi-Scale Channel-Wise Attention Based Quality Enhancement Module

To efficiently fuse the temporal information between channels and generate the residual map for the target frame, we propose an efficient multi-scale channel-wise attention module (MCQE). In TSTF, the encoder and decoder structure produces multi-scale information. As the resolution of feature map decreases, the number of channels doubles, allowing for a larger receptive field within a fixed window size. Thus for feature map of low resolution, the channel information of

---

Fig. 2. The framework of our proposed TVQE method, which consists of the Transformer based Spatio-Temporal feature Fusion (TSTF) Module and the Multi-scale Channel-wise Attention based Quality Enhancement (MCQE) Module. The TSTF module is designed to exploit spatio-temporal correlation between multiple frames. After TSTF, the multi-scale information between channels in the feature map is further fused by the MCQE module, and finally generate the enhanced frame.
large inter-frame motion can be effectively extracted. While with the increased resolution, the channel information of the feature map contains information at the pixel level, enabling the correlation modeling of small movements. In MCQE, the channel attention of each small-scale feature from the TSTF is calculated and then fused with the next level of large-scale features. Finally, features containing multi-scale information is calculated and then fused with the next level of large-scale attention and then sums with MXm. Pixel Shuffle attention, and after convolution layer to get the final residual map. The whole × in the temporal feature map is further fused and enhanced.

The proposed MCQE module is shown in Fig. 2-(b). The decoder of TSTF contains four different scales of features (TXm, SXm, MXm, LXm). To efficiently fuse multi-scale features, the channel attention is first computed for small-scale features. Then, they are fused with the next level of features by summation operation. Finally, the texture features at different scales are fused into the enhanced residual map. Specifically, the minimum scale feature TXm calculates the channel-level attention, and after Pixel Shuffle upsampling, it is summed with SXm to obtain SXm. Accordingly, SXm calculates the attention and then sums with MXm to obtain MXm. MXm calculates the attention and then sums with LXm to obtain LXm. Then, LXm fuses the information of four different scales in the decoder from TSTF and undergoes a final calculation of channel attention. The temporal information between channels in the temporal feature map is further fused and enhanced.

Finally, the number of channels is reduced to 1 by a 3 × 3 convolution layer to get the final residual map. The whole process are as follows:

\[ TX_t^m = M(TX_t^m), \]  
(11)  
\[ SX_t^m = UP(CA(TX_t^m + SX_t^m)), \]  
(12)  
\[ MX_t^m = UP(CA(SX_t^m + MX_t^m)), \]  
(13)  
\[ LX_t^m = UP(CA(MX_t^m + LX_t^m)), \]  
(14)  
\[ I_t = Conv(CA(LX_t^m)). \]  
(15)

where CA denotes Channel Attention block, UP denotes the upsample layer of Pixel Shuffle and Conv is the convolution layer.

As shown in Fig. 3, the Channel Attention block consists of two parts: Multi-Head Transposed Attention (MTA) and Feed-Forward Network (FN). The MTA computes the cross-covariance between channels instead of spatial dimensions, which reduces the computational overhead of the network. The input feature Xm is fed into one Convolution layer to generate Q ∈ RHW×C, K ∈ RHW×C, V ∈ RHW×C. Then use the reshape operation to get \( \hat{Q} \in R^{HW \times C}, \hat{K} \in R^{HW \times C}, \hat{V} \in R^{HW \times C} \). Finally, MTA calculates the dot product of \( \hat{Q} \) and \( \hat{K} \) to generate the channel attention map M with size C × C, which can be expressed as:

\[ M = \text{Softmax}(\hat{Q} \cdot \hat{K}) \]  
(16)

To get more accurate residual information, we utilize FN to enhance the details.

C. Training Scheme

For frame Xt at time t, we use a two-stage training strategy to enhance its quality. In the first stage, we use Charbonnier Loss [61] to optimize the parameters of TVQE. In the second stage, we use \( L_2 \) Loss to further fine-tune the model for a better visual result. Finally, the loss functions are defined as below:

\[ L_{\text{charb}} = \sqrt{(X_t^r - X_t^{raw})^2 + \epsilon}, \]  
(17)  
\[ L_{\text{mse}} = \|X_t^r - X_t^{raw}\|_2^2, \]  
(18)  
\[ L = \alpha \times L_{\text{charb}} + \beta \times L_{\text{mse}}, \]  
(19)

where \( X_t^r \) denotes the enhanced video frame at time t, \( X_t^{raw} \) denotes the original uncompressed frame (ground truth), and \( \epsilon \) is a constant set to 10\(^{-6}\). \( \alpha \) and \( \beta \) are the weights of the loss.

IV. EXPERIMENT

A. Datasets

Following [21], [23], we use the MFQE 2.0 [21] and LDV [25] datasets for training, and JCT-VC [64] dataset for testing. All sequences were compressed by HEVC HM 16.5\(^1\) with LDP (Low-Delay-P) configuration. Five QPs (quantization parameters), i.e., 22, 27, 32, 37,42 at different compression bit rates are used for experiments.

1) MFQE 2.0: It contains totally 128 sequences, in which training set contains 108 sequences. The sequences are acquired from Xiph.org [65] and VQEG [66], with resolutions ranging from SIF (352 × 240) to WQXGA (2560 × 1600).

2) LDV: It is proposed in NTIRE 2021 challenge [25] with 240 sequences, which consists of training set, validation set and test set. We use 200 sequences from the training set as training data and 40 sequences from the validation and test sets for validation, all sequences have the same resolution of 960 × 536.

3) JCT-VC: The test set has 18 sequences, delivered by JCT-VC (Joint Collaborative Team on Video Coding) for evaluating the performance of our model. There are totally five resolutions ranging from 240p (416 × 240) to WQXGA (2560 × 1600), named as Class A to E.

B. Implementation Details

For network structure, the window size is set to 8 in TSTF. In the first to third stages of the encoder in TSTF, the number of ASwin-TB modules is set to [2, 2, 2], and the number of attention heads is set to [2, 2, 2]. The Patch embedding dimension is set to 48, and the MLP-ratio is 1.

\(^1\)https://hevc.hhi.fraunhofer.de/trac/hevc/milestone/HEM-16.5

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE I
OBJECTIVE RESULTS OF ΔPSNR (dB) / ΔSSIM (× 10^-2) ON JCT-VC DATASET AT 5 DIFFERENT QPS. THE BEST AND SECOND BEST PERFORMANCE ARE IN BOLD AND UNDERLINED, RESPECTIVELY

| QP | Approach | AR-CNN [13] | RNAN [33] | MFQE2.0 [21] | STDF-RML [23] | RFDA [24] | MRDN [62] | FastCNN [63] | Ours TVQEL |
|----|----------|------------|----------|-------------|--------------|----------|----------|----------|----------|
|    | Metrics  | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM | PSNR / SSIM |
| A  | Traffic OnStreet | 0.24 / 0.47 | 0.40 / 0.86 | 0.59 / 1.02 | 0.73 / 1.15 | 0.80 / 1.28 | 0.72 / 1.16 | 0.76 / 1.25 | 0.88 / 1.44 |
| B  | Kinomo ParkScene Coctus | 0.22 / 0.65 | 0.33 / 0.98 | 0.55 / 1.18 | 0.83 / 1.61 | 1.02 / 1.86 | 0.82 / 1.65 | 1.00 / 1.68 | 0.99 / 1.82 |
| C  | RaceHorses BQMall PartyScene BasketballDrive | 0.22 / 0.43 | 0.39 / 0.99 | 0.39 / 0.80 | 0.55 / 1.35 | 0.48 / 1.23 | 0.60 / 1.48 | 0.57 / 1.29 | 0.61 / 1.59 |
| D  | RaceHorses BSSquare BlowingBubbles BasketballPass | 0.27 / 0.55 | 0.42 / 1.02 | 0.59 / 1.43 | 0.83 / 2.08 | 0.85 / 2.11 | 0.83 / 2.09 | 0.85 / 2.18 | 0.86 / 2.30 |
| E  | FourPeople Johnny KristenAndSara | 0.37 / 0.50 | 0.70 / 0.97 | 0.75 / 0.95 | 0.94 / 1.17 | 1.13 / 1.36 | 0.94 / 1.18 | 1.05 / 1.25 | 1.16 / 1.42 |

In the training stage, we crop 128 × 128 patches randomly from the compressed video and the corresponding raw video as training samples. Random flips and rotations are also used for data augmentation. The batch size is set to 32. We train the model using the Adam optimizer (β1 = 0.9, β2 = 0.999, ϵ = 10^-8). The learning rate is 10^-4 throughout the training process. In the first stage of training, α is set to 1 and β is set to 0 in Eq. (19). In the second stage, α is set to 0 and β is set to 1. All experiments are performed on the NVIDIA TITAN RTX.

For testing, we evaluate the results using ΔPSNR and ΔSSIM, as well as BD-rate. All tests are performed on the Y-channel in YUV space.

C. Comparison With State of the Art Methods

To demonstrate the effectiveness of our method, we compare the proposed method with seven state-of-the-art methods, including single-frame based methods (AR-CNN [13], RNAN [33], DnCNN [17], DCAD [39]), DS-CNN [40]) and multi-frame based methods (MFQE 1.0 [20], MFQE 2.0 [21], STDF [23], RFDA [24], MRDN [62] and FastCNN [63]).

1) Objective Results: Table I presents the Objective results of our method and seven state-of-the-art methods on ΔPSNR and ΔSSIM. It can be seen that our method outperforms the current state-of-the-art methods on most sequences when QP=37. Besides, our method TVQE outperforms the seven models in terms of the average ΔPSNR and ΔSSIM for all QPs. Meanwhile, the gain of our method TVQE on SSIM is higher than PSNR obviously. Such as QP=37, our method TVQE gains 7.7% on ΔPSNR and 12.3% on ΔSSIM over the second best method RFDA, which indicates that our method TVQE provides better visual effects.

As for the single-frame based methods, RNAN proposes non-local attention blocks to obtain the remote dependency of the feature map, and finally achieves the best performance among all single-frame based methods. It gains about 69% over DnCNN, which reflects the superiority of the Transformer-based method. However, the single-frame based method cannot use temporal information and has limited performance. Our method computes spatial attention and channel attention over multiple frames, and achieves ΔPSNR of 0.98, which is about 123% compared to RNAN.

As for the multi-frame based methods, MFQE 2.0 calculates the explicit optical flow of compressed video and achieves an average ΔPSNR of 0.56. STDF proposes deformable convolution to align video frames, which solves the problem of inaccurate optical flow estimation, and achieves an average ΔPSNR of 0.83. RFDA utilizes the RF (Recursive Fusion) module to exploit temporal information within a longer time range, and obtains ΔPSNR of 0.91. Our method TVQE utilizes the long-range modeling property of Transformer to exploit the temporal information, which further increases the PSNR with an average ΔPSNR of 0.98, which demonstrates the effectiveness of our method.

2) Comparison of Model Complexity: Table III shows the Parameters, FLOPs, GPU consumption, Inferred speed and ΔPSNR of our method, STDF [23] and RFDA [24]. As can
TABLE II
RESULTS OF BD-RATE REDUCTION (%) AT QP = 22, 27, 32, 37 AND 42 WITH THE HEVC AS ANCHOR. THE BEST AND SECOND BEST PERFORMANCE ARE IN BOLD AND UNDERLINED, RESPECTIVELY

| Sequence          | AR-CNN [13] | DnCNN [17] | DCAD [39] | DS-CNN [40] | MQPE 1.0 | MQPB 2.0 | STDFT-R3L [23] | RFDA [24] | FastCNN [63] | Ours          |
|-------------------|-------------|------------|-----------|-------------|-----------|-----------|----------------|-----------|-------------|---------------|
| **A**             |             |            |           |             |           |           |                 |           |             |               |
| Traffic           | 7.40        | 8.54       | 9.97      | 9.18        | 14.56     | 16.98     | 21.19          | 22.70     | **26.48**    | **24.51**     |
| PeopleOnStreet    | 6.99        | 8.28       | 9.68      | 8.67        | 13.71     | 15.08     | 17.42          | 21.11     | **27.29**    | **23.44**     |
| **B**             |             |            |           |             |           |           |                 |           |             |               |
| Kimono            | 6.07        | 7.33       | 8.44      | 7.81        | 12.60     | 13.34     | 17.96          | 22.32     | **26.83**    | **23.37**     |
| ParkScene         | 4.47        | 5.04       | 5.68      | 5.42        | 12.04     | 13.66     | 18.10          | 19.85     | **21.95**    | **20.39**     |
| Cactus            | 6.16        | 6.80       | 8.69      | 8.78        | 12.78     | 14.84     | 21.54          | 21.78     | **24.20**    | **23.22**     |
| BQTerrace         | 6.86        | 7.62       | 9.98      | 8.67        | 10.95     | 14.72     | 24.71          | 24.41     | **25.62**    | **28.68**     |
| BasketballDrive   | 5.83        | 7.33       | 8.94      | 7.89        | 10.54     | 11.85     | 16.75          | 20.24     | **20.54**    | **21.67**     |
| **C**             |             |            |           |             |           |           |                 |           |             |               |
| RaceHorses        | 5.07        | 6.77       | 7.62      | 7.48        | 8.83      | 9.61      | **15.62**      | 14.29     | 15.16       | 13.69         |
| BQMall            | 5.60        | 7.01       | 8.65      | 7.64        | 11.11     | 13.50     | 21.12          | 21.62     | 19.85       | **21.65**     |
| PartyScene        | 1.88        | 4.02       | 4.88      | 4.08        | 6.67      | 11.28     | **22.24**      | 21.11     | 18.75       | **23.21**     |
| Basketball Drill  | 4.67        | 8.02       | 9.80      | 8.22        | 10.47     | 12.63     | 15.94          | 18.06     | 17.16       | **21.24**     |
| **D**             |             |            |           |             |           |           |                 |           |             |               |
| RaceHorses        | 5.61        | 7.22       | 8.16      | 7.35        | 10.41     | 11.55     | 15.26          | 17.57     | **18.97**    | 16.88         |
| BQSquare          | 0.68        | 5.49       | 6.11      | 3.94        | 2.72      | 11.00     | 33.36          | 31.65     | 26.26       | **35.36**     |
| Blowing Bubbles   | 3.19        | 5.10       | 6.13      | 5.55        | 10.73     | 15.20     | 23.54          | 22.89     | 20.54       | **24.05**     |
| Basketball Pass   | 5.11        | 7.03       | 8.35      | 7.49        | 11.70     | 13.43     | 18.42          | 20.42     | 18.93       | **20.97**     |
| **E**             |             |            |           |             |           |           |                 |           |             |               |
| FourPeople        | 8.42        | 10.12      | 12.21     | 11.13       | 14.89     | 17.50     | **22.91**      | 22.84     | 21.98       | **24.56**     |
| Johnny            | 7.66        | 10.91      | 13.71     | 12.19       | 15.94     | 18.57     | **24.55**      | 23.87     | 23.84       | **29.44**     |
| Kristen And Sara  | 8.94        | 10.65      | 12.93     | 11.49       | 15.06     | 18.34     | 23.64          | 24.47     | 24.27       | **28.04**     |
| **Average**       | 5.59        | 7.36       | 8.89      | 7.85        | 11.41     | 14.06     | 20.79          | 21.73     | **22.05**    | **23.58**     |

TABLE III
COMPARISON OF PARAMETERS, FLOPS, GPU CONSUMPTION, INFERRED SPEED AND ΔPSNR BETWEEN OUR METHOD AND SOME MAINSTREAM METHODS. FOR A FAIR COMPARISON, ALL METHODS WERE RE-TESTED ON THE NVIDIA TITAN RTX. THE FLOPS AND FRAMES PER SECOND (FPS) ARE EVALUATED AT RESOLUTION 720P. THE GPU CONSUMPTION IS EVALUATED AT RESOLUTION 1080P. THE BEST AND SECOND BEST PERFORMANCE ARE IN BOLD AND UNDERLINED, RESPECTIVELY

| Method          | Parameter(M) | FLOPs(T) | GPU(GB) | FPS    | ΔPSNR |
|-----------------|--------------|----------|---------|--------|-------|
| STDF (23)       | 1.27         | 0.77     | 5.8     | 6.4    | 0.83  |
| RFDA (24)       | 1.27         | 0.53     | 10.8    | 6.5    | 0.91  |
| TVQF(Proposed)  | 2.09         | 0.12     | 8.4     | 7.0    | 0.98  |

be seen, although our model is based on Transformer, it still has a fast inference speed (FPS) and low FLOPs, thanks to the encoder-decoder structure and skip connections. At the same time, our method is hardware friendly compared to RFDA [24] as it requires less GPU memory. More specifically, comparing to STDF at 720p resolution, our method is 9.4% faster at inference speed (from 6.4 to 7.0, in Table III), and with a 84.4% reduction in FLOPs (from 0.77 to 0.12, in Table III), as well as a 18.1% improvement in average ΔPSNR performance when QP= 37 (from 0.83 to 0.98, in Table I). RFDA is based on STDF by adding RF module, and thus consumes more GPU resources. Comparing to RFDA, our method outperforms RFDA in terms of inference speed and GPU consumption at all resolutions. Specifically, under 1080p resolution, the inference speed is improved by 7.7% (from 6.5 to 7.0, in Table III) and FLOPs is reduced by 77.4% (from 0.53 to 0.12, in Table III). Overall, although our proposed method has a larger parameters compared to the CNN-based methods, it brings faster inference speed and better performance with much lower computational complexity and moderate GPU consumption.

3) Quality Fluctuation: High quality fluctuation will damage the user’s viewing experience. Thus, quality fluctuation is evaluated in this section following the settings in [21]. The PSNR of each frame in two sequences are plotted in Fig. 4, where the horizontal axis represents the frame index, and the vertical axis represents the frame quality. It can be observed that the HEVC compressed sequences have severe quality fluctuations (i.e., quality differences between high quality frames and adjacent low quality frames). Compared to both STDF and RFDA, our method provides better PSNR and smaller quality fluctuations, effectively improving the QoE.

4) Rate-Distortion Performance: Fig. 6 presents the rate distortion curves for the four sequences. It can be seen that our method outperforms other methods on both sequences with huge motion (e.g., Class C, Basketball Drill) and smooth motion (e.g., Class E, Johnny). In addition, we also calculate the BD-rate reduction of PSNR on five QPs (= 22, 27, 32, 37, 42) in Table II. Our method provides an average BD-rate reduction of 23.58%, which is better than the state-of-the-art CNN-based method RFDA with 21.73%. It demonstrates that TVQE exhibits a better rate distortion performance, which can provide superior visual effects under the same bit-rate.
5) Subjective Results: Fig. 5 gives the subjective results for the five sequences. It can be seen that the single-frame based quality enhancement methods AR-CNN [13] and DnCNN [17] do not make use of temporal information, so the enhanced video frames still have serious compression artifacts (e.g., block effect, ringing effect). With the help of temporal information, CNN-based multi-frame quality enhancement methods MFQE 2.0 [21] and STDF [23] provide better visual effects with the help of reference frames, but the locality of the convolution operation prevents these methods from taking full advantage of the temporal information of reference frames, resulting in enhanced video frames that are too smooth and lack of detailed texture. RFDA [24] further implicitly expands the temporal range with RF to better recover details, but the RF module consumes large computational resources and decreases the inference speed. TVQE is based on Transformer, which has better remote correlation capability than convolution, thus resulting in better exploration of spatio-temporal information and finer recovery of textures. For example in Fig. 5, the player’s fingers in BasketballDrill, the rope on the horse in Racehorses, the textures on the railings in BQSquare and BQTerrace, and the shadow of the sneaker in BasketballPass can be better recovered by our method than other methods.

6) Results on VVC Test Sequences: To demonstrate the effectiveness of our proposed method on VVC compressed video sequences, we test both our method TVQE and RFDA method on videos compressed with VVC and QP = 37. Table IV shows that TVQE yields better results than RFDA on VVC compressed sequences. The proposed method achieves an average ∆PSNR / ∆SSIM of 0.04 / 0.10 (up to 0.05 / 0.27), which proves the applicability of the TVQE method on the latest VVC compression standard.
D. Ablation Study

In this section, we perform ablation experiments as well as specific analysis of the proposed method. We take STDF as baseline and replace the modules from different models to analyze their effects. For a fair comparison, all models are retrained by the same training approach as the proposed method. The results of FLOPs and inference speed FPS are obtained at 720p resolution. The GPU consumption is obtained at 1080p resolution, and ΔPSNR / ΔSSIM takes the average results of the test sequences in ClassA-E at QP=37.

1) Effectiveness of TSTF: To illustrate the effectiveness of the TSTF module, we compare the proposed TSTF with the baseline STDF (from STDF-R3L) and STFF (from RFDA). As shown in Table V, we replaced the STDF module with STFF and TSTF (the second and third row), where the TSTF provides best performance while having the fastest inference speed and second lowest GPU consumption. Specifically, compared to the baseline STDF, the Transformer-based TSTF is able to explore global temporal information within a time window, with an improvement of ΔPSNR by 0.10 (from 0.83 to 0.93) and ΔSSIM (×10−2) by 0.24 (from 1.51 to 1.75). Moreover, benefit from the autoencoder structure and skip connections, TSTF has a 6.3% speedup (from 6.4 to 6.8) inference speed compared to STDF and 8.6% reduction in GPU consumption (from 5.8 to 5.3). Compared with STFF, TSTF does not have to utilize additional information outside the time window, resulting in 16.7% lower GPU consumption (from 10.8 to 9.0). 67.2% lower FLOPs from 0.61 to 0.20 and 4.6% higher inference express (from 6.5 to 6.8), which demonstrates the effectiveness of the TSTF module.

2) Effectiveness of MCQE: To illustrate the effectiveness of the MCQE module, we replace the baseline quality enhancement module QE with MCQE (fourth to sixth rows). As shown in the results (third and sixth lines), MCQE provides a ΔPSNR gain of 0.05 (from 0.93 to 0.98) and a ΔSSIM gain of 0.07 (from 1.75 to 1.82) compared to the CNN-based QE. Meanwhile, the multi-scale channel attention resulted in a further 40% reduction in FLOPs (from 0.20 to 0.12) and 6.7% reduction in memory consumption (from 9.0 to 8.4). In addition, the inference speed improved from 6.8 to 7.0 and outperformed all methods, which demonstrates the effectiveness of the MCQE module.

3) Influence of the Number of ASwin-TB: Table VI shows the impact of the number of ASwin-TB units on the TSTF module at different resolutions. Through comparison, progressive growth increases the parameter volume by 41.1% (from 2.09 to 2.95) and the FLOPs by 5.2% (from 0.116 to 0.122), while also resulting in faster inference speeds. However, the performance gain shows a significant decline. Therefore, it can be concluded that information mining at each resolution is crucial.

4) Window Size Impact: From the analysis above, we know that ASwin-TB based on Transformer is more suitable for mining spatio-temporal information in videos than Convolutions. We further investigate the impact of different window sizes, and the results are shown in Table VII. It can be seen from the table that as the window size increases, the model’s parameters and complexity remain the same. With a larger window size (from 7 to 8), a single window covers more information, enabling the model to capture a larger range of spatio-temporal information in videos, resulting in a higher ΔPSNR gain. However, there is a bottleneck in performance gain, as we found that when the window size is set to 9, the ΔPSNR performance is at the same level as that of a window size of 8, while the ΔSSIM gain decreases. Additionally, larger windows consume more GPU and have lower inference speeds. Therefore, in this paper, we choose the ASwin-TB module with a window size of 8 as the basic unit to construct TSTF.

V. Conclusion

In this paper, we propose an end-to-end Transformer based network for compressed video enhancement (TVQEC), which mainly consists of two modules, TSTF module and MCQE module. The TSTF module can efficiently explore temporal information within the time window, while the MCQE module can well fuse the temporal information. The proposed method outperforms the CNN-based methods both in terms of performance and inference speed. The proposed modules can be used as plug-ins in other applications, such as video super-resolution and video interpolation, to explore and fuse temporal information more effectively.
Li Yu received the B.S. degree from Soochow University, Suzhou, China, in 2012, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2017. From 2017 to 2018, she was a Postdoctoral Researcher with the Department of Signal Processing, Tampere University of Technology, Tampere, Finland. Since 2018, she has been a Faculty Member with the Nanjing University of Information Science and Technology, Nanjing, China. Her research interests include video streaming, video coding, image and video processing, computer vision, and deep learning.

Wenshui Chang received the B.S. and M.S. degrees from the School of Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interests primarily focus on video coding, computer vision, and deep learning.

Shiyu Wu is currently pursuing the master’s degree with the School of Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include video restoration, semi-supervised learning, and machine learning.

Moncef Gabbouj (Fellow, IEEE) is a Professor of Information Technology with the Department of Computing Sciences, Tampere University, Finland. He was an Academy of Finland Professor. His research interests include big data analytics, multimedia analysis, artificial intelligence, machine learning, pattern recognition, nonlinear signal processing, video processing, and coding. He is the Finland Site Director of the NSF IUCRC funded Center for Big Learning. He is a Fellow of the Asia–Pacific Artificial Intelligence Association. He is member of the Academia Europaea, the Finnish Academy of Science and Letters, and the Finnish Academy of Engineering Sciences.