Scalable Residual Laplacian Network for HEVC-compressed Video Restoration

CLAUDIO ROTA, MARCO BUZZELLI, SIMONE BIANCO, and RAIMONDO SCHETTINI,
University of Milano-Bicocca, Milan, Italy

We present a novel Convolutional Neural Network that exploits the Laplacian decomposition technique, which is typically used in traditional image processing, to restore videos compressed with the High-Efficiency Video Coding (HEVC) algorithm. The proposed method decomposes the compressed frames into multi-scale frequency bands using the Laplacian decomposition, it restores each band using the ad-hoc designed Multi-frame Residual Laplacian Network (MRLN), and finally recomposes the restored bands to obtain the restored frames. By leveraging the multi-scale frequency representation of compressed frames provided by the Laplacian decomposition, MRLN can effectively reduce the compression artifacts and restore the image details with a reduced computational cost. In addition, our method can be easily instantiated in various versions to control the tradeoff between efficiency and effectiveness, representing a versatile solution for scenarios with constrained computational resources. Experimental results on the MFQEv2 benchmark dataset show that our method achieves the state-of-the-art performance in HEVC-compressed video restoration with a lower model complexity and shorter runtime with respect to existing methods. The project page is available at https://github.com/claudiom4sir/LaplacianVCAR.

CCS Concepts: • **Computing methodologies** → **Image processing**; **Artificial intelligence**; **Appearance and texture representations**;

Additional Key Words and Phrases: Video restoration, laplacian decomposition, compression artifact reduction, deep learning

ACM Reference format:
Claudio Rota, Marco Buzzelli, Simone Bianco, and Raimondo Schettini. 2025. Scalable Residual Laplacian Network for HEVC-compressed Video Restoration. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 6, Article 164 (July 2025), 22 pages.
https://doi.org/10.1145/3727147

1 Introduction

With the increasing popularity of video content, the need for video compression techniques has become more significant. The massive quantity of video content produced and consumed by users every day makes it fundamental to limit memory occupation and to efficiently use the transmission

This work was partially supported by the MUR under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy.
Authors' Contact Information: Claudio Rota (corresponding author), University of Milano-Bicocca, Milan, Italy; e-mail: claudio.rota@unimib.it; Marco Buzzelli, University of Milano-Bicocca, Milan, Italy; e-mail: marco.buzzelli@unimib.it; Simone Bianco, University of Milano-Bicocca, Milan, Italy; e-mail: simone.bianco@unimib.it; Raimondo Schettini, University of Milano-Bicocca, Milan, Italy; e-mail: raimondo.schettini@unimib.it.

ACM 1551-6865/2025/7-ART164
https://doi.org/10.1145/3727147

bandwidth in video streaming applications. In recent years, the **High-Efficiency Video Coding (HEVC)** [33] standard has emerged as a prominent video compression approach due to its ability to provide improved compression efficiency compared to its predecessors [28]. However, one significant drawback is the presence of compression artifacts in HEVC-compressed videos, such as blocking and ringing effects, due to lossy quantization and block-based coding, which severely degrade the **Quality of Experience (QoE)** [37]. In addition, it was noticed that HEVC compression introduces quality fluctuation [44] due to the different frame coding configurations, which also negatively impacts the QoE [17]. Therefore, designing methods for the restoration of compressed videos can provide added value to the sharing and consumption of digital videos.

Recently, **Convolutional Neural Networks (CNNs)** have shown significant progress in video restoration tasks, such as denoising [6, 38], deblurring [23, 46], and super-resolution [1, 24]. In the context of HEVC-compressed video restoration, early works mainly focused on single-frame restoration [26, 42]. Due to the lack of proper mechanisms for taking advantage of information from multiple frames, their performance was limited. Other works exploited multiple frames to obtain better restoration results and reduce the quality fluctuation [10, 12, 27]. Despite the advancements in this field, there are still limitations that must be addressed. Existing methods receive a sequence of compressed frames as input, and use a CNN to exploit the complementary information contained in multiple frames to restore the target. However, these methods do not discriminate the various frequency components within compressed frames, and instead treat them equally, thus failing to remove the artifacts at specific frequency bands, and failing to correctly restore image details. In addition, they typically require a considerable amount of computational resources and long runtime, which makes them impractical to be used in scenarios where efficiency is crucial, such as video streaming and conferencing.

In this work, we present a method for the restoration of HEVC-compressed videos addressing the existing limitations. We design our method to leverage the explicit multi-scale frequency band representation provided by the Laplacian decomposition technique [5] and the complex pattern understanding capability of CNNs. Specifically, we first decompose the compressed frames into multi-scale frequency bands using the Laplacian decomposition. Then, we restore each band using a CNN, named **Multi-frame Residual Laplacian Network (MRLN)**, which progressively removes the compression artifacts from each frequency band to restore the uncompressed signal. Finally, we recompose the restored bands to obtain the restored frames. Compared to previous works that implicitly learn the multi-scale representation of compressed frames, we explicitly use the frequency representation provided by the Laplacian decomposition [10, 20, 47]. As a consequence, we can use a simpler CNN architecture design to reduce the compression artifacts in each band and restore frame details with a reduced computational cost. The design of our method is based on the intuition that decomposing the compressed frames into multi-scale frequency bands using the Laplacian decomposition provides three key advantages: (1) it creates a robust and structured representation of compressed frames, simplifying the detection and removal of compression artifacts; (2) it enables frequency-specific restoration, where each band can be processed according to the effect of the compression on the different frequencies; and (3) it introduces a negligible overhead for its computation, as it relies on simple operations like Gaussian filtering and subtractions.

Inspired by the principles of model scaling [36], we design our method to be parametrically scalable, so that it can be adapted to various scenarios where the computational resources may vary. We can instantiate different versions of our method with increasing complexity and better restoration performance by controlling only a few parameters, allowing to tune the tradeoff between efficiency and effectiveness. As shown in Figure 1, the different versions of our method can process frames at 416×240 pixel resolution in a range from 30 to 54 **Frames per Second (FPS)** on an NVidia

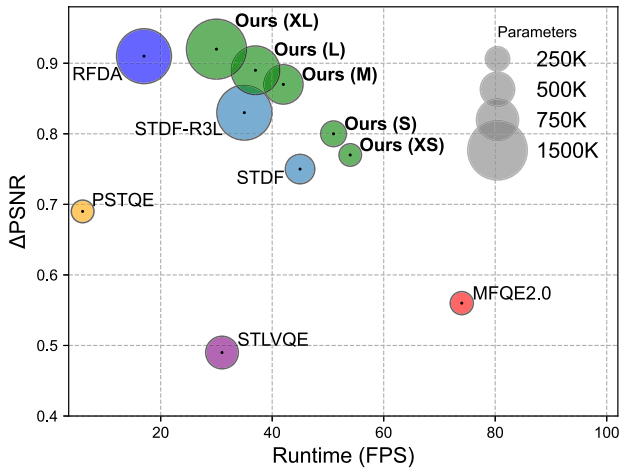


Fig. 1. Comparison between the proposed method and state-of-the-art methods for HEVC-compressed video restoration in terms of ΔPSNR and runtime (FPS). The different versions of our method (green bubbles) lie in the Pareto front of optimality when compared to the other methods. Here, runtime is computed at 416×240 pixel resolution on an NVidia GeForce GTX 1080 GPU. FPS, Frames per Second.

GeForce GTX 1080 GPU. In addition, they lie on the Pareto front of optimality when compared to other state-of-the-art methods for HEVC-compressed video restoration [9, 10, 12, 29, 47].

The main contributions of this work are the following: (1) we propose a method to restore HEVC-compressed videos that explicitly leverages the robust multi-scale frequency representation provided by the Laplacian decomposition, enabling a more precise artifact removal and enhanced preservation of details; (2) instead of restoring the entire frame, we show that specializing a CNN in restoring its decomposition into frequency bands allows considerably improving the performance adding a negligible overhead and without increasing the model complexity; (3) we present a parametrically scalable design for our method, which allows easily instantiating it in different versions to control the tradeoff between efficiency and effectiveness. This makes our method versatile and suitable for applications where computational requirements vary; (4) we conduct extensive experiments on the MFQEv2 benchmark dataset to show that the proposed method achieves state-of-the-art performance in HEVC-compressed video restoration while using fewer network parameters and requiring a shorter runtime compared to existing methods.

The remainder of the article is organized as follows: Section 2 presents an overview of related works concerning single-frame and multi-frame methods for compressed video restoration; Section 3 provides motivations and describes the proposed multi-frame method for compressed video restoration; Section 4 reports experimental results on the MFQEv2 benchmark dataset [12], including qualitative and quantitative comparisons of the proposed method with state-of-the-art methods; Section 5 concludes this article outlining future research directions.

2 Related Works

2.1 Single-frame Video Restoration

In the context of HEVC-compressed video restoration, various works focused on restoring single frames [15, 26, 40, 42, 43]. Wang et al. [40] introduced the Deep CNN-based Auto Decoder, a neural network exploiting frame spatial information to reduce the distortions of compressed videos. Yang et al. [43] proposed the Decoder-side Scalable CNN, which was later improved by Yang et al.

[42]. He et al. [15] exploited some prior knowledge about the HEVC algorithm to guide the video restoration process. Lin et al. [26] proposed an adaptive switching mechanism to select a specific model depending on both video contents and the distortions to be restored.

Since these methods only process single frames, they ignore temporal information reaching only limited performance [30].

2.2 Multi-frame Video Restoration

Based on the observation that frames in a limited temporal interval are likely to represent the same objects in the scene, other works exploit temporal information from multiple frames to obtain better restoration performance [9, 10, 12, 18–20, 25, 27, 44, 47]. Yang et al. [44] observed that HEVC-compression introduces quality fluctuation within compressed frames, and consequently developed a method, named **Multi-frame Quality Enhancement (MFQE)**, that exploits peak-quality frames, i.e. frames with lower compression, to improve the quality of the others. Guan et al. [12] later proposed MFQE2.0, which improves MFQE under different aspects, such as peak-quality frame detection and quality enhancement approach. Deng et al. [9] developed the first method applying **Deformable Convolutional Networks (DCN)** [8] in both spatial and temporal dimensions. Their method, named **Spatio-temporal Deformable Fusion (STDF)**, first aligns frames using DCN and then uses a sequence of stacked convolutions for frame restoration. Zhao et al. [47] further improved STDF by proposing a more sophisticated method, called **Recurrent Fusion Deformable Attention (RFDA)**, which combines a recurrent architecture with the attention mechanism [13]. Huo et al. [20] proposed a recurrent method that avoids the use of optical flow for frame alignment and computes attention maps to impose more attention on the edges and textures of compressed frames. Huang et al. [18] developed a method specifically designed to be applied to compressed animation and game videos with a time-domain information cross-fusion module and a detail recovery module based on the attention mechanism. Ding et al. [10] proposed the **Patch-wise Spatial-Temporal Quality Enhancement (PSTQE)**, a network that first extracts spatial and temporal features from a sequence of input frames and then uses an attention mechanism [13] to distill relevant information. Luo et al. [27] designed a method, called **Spatio-temporal Detail Retrieval (STDR)**, that integrates the alignment features of different receptive fields for more accurate deformable offsets, leading to a better use of temporal information. Schiopu and Munteanu [31] developed Attention-based Shared weights Quality Enhancement CNN by incorporating three key elements: the attention mechanism [13] for feature map refinement, the weight sharing concept to reduce model complexity, and multi-scale processing for better feature fusion. Motivated by the need for efficient processing, Chen et al. [7] presented MFQE (Fast-MFQE) integrating an image pre-processing module to minimize redundant information, a spatio-temporal fusion attention module for effectively merging information across video frames, and a feature reconstruction module designed to enhance frame quality efficiently. Jiang et al. [21] proposed **Spatial-Temporal Attention-guided Enhancement Network (STAGE-Net)**, which uses dynamic filter processing instead of optical flow estimation to reduce the overall computational complexity. In addition, they adopted a self-attention mechanism [39] to improve the visual quality of enhanced video frames with bitrate constraints. Huang et al. [19] proposed FastCNN to achieve fast video restoration, using an efficient alignment module and prior compression information. Recently, Qu et al. [29] proposed a lightweight and fast method called **Spatio-temporal Look-up table Video Quality Enhancement (STLVQE)** for online video quality enhancement tasks. Using look-up tables, STLVQE can extract spatio-temporal information from the video with reduced time consumption. Zhang et al. [45] proposed Hierarchical Frequency-based Upsampling and Refining neural network, which uses an implicit frequency upsampling module and hierarchical and iterative refinement module: the first module uses DCT-domain prior to accurately reconstructs the DCT-domain loss,

whereas the second module is used to refine the feature maps improving the visual quality of the final output. Ehrlich et al. [11] proposed leveraging the bitstream structure of compressed videos, such as motion estimation vectors and peak-quality frame information, and integrating this information into a CNN. Li et al. [25] proposed Enhanced Compressed Video Super-Resolution to perform super-resolution of compressed videos enhanced using a single model.

3 Methodology

3.1 Analysis of HEVC Compression on Frequency Bands

The main intuition for the proposed method emerges from the analysis of how HEVC compression affects the frequency bands of video frames. HEVC introduces artifacts and degrades the quality of the frames by altering their frequency content due to lossy quantization and block-based coding. This degradation is not uniform across all frequency bands, but it differently affects certain bands causing changes in their distribution and a significant loss in their energy. We demonstrate this by analyzing the effect of the HEVC compression on frequency bands of frames from the MFQEv2 dataset [12] compressed with HM16.5 under **Low Delay P (LDP)** at different **Quantization Parameters (QPs)**. Note that higher QPs indicate higher compression. We decompose each compressed frame into high-, mid-, and low-frequency bands using the Laplacian decomposition and then compute the ΔEnergy, i.e. the energy loss, of each frequency band. Given a specific frequency band, its energy is obtained by summing up its squared values, and ΔEnergy refers to the percentage of energy loss with respect to the uncompressed frames. Figure 2 shows the histograms of different frequency bands considering two QPs. The distribution of high-frequency bands (top) is more affected than those of mid- and low-frequency bands (bottom). The **Standard Deviation (STD)** of the distribution of uncompressed frequency bands is higher than the one of compressed bands, showing the loss of some frequency components. In addition, the frequency distribution is altered more by compression with higher QPs (right).

We also show that not all frequency bands have the same importance in improving the frame quality. We show this by replacing specific frequency bands of the compressed frames with the ones of the uncompressed frames. Such an operation can be seen as the frequency band restoration process performed by a CNN that restores from specific bands. The results reported in Table 1 show that, at a given QP, the energy loss in high-frequency bands is considerably higher than in the other frequency bands. Similarly, replacing the frequency bands that suffered higher energy loss leads to obtaining the best improvement in video quality. These results suggest that the frequency bands contained in the compressed frames should be processed differently. For these reasons, we propose exploiting the decomposition into frequency bands provided by the Laplacian decomposition and using a CNN to restore each band component by targeting the compression artifacts that can be found within the considered band.

3.2 Overview of the Proposed Method

Given a sequence of $2N + 1$ compressed frames $I^{LQ}_{[t-N:t+N]}$, the middle frame I^{LQ}_t is the frame to be restored, while the others represent its temporal neighborhood and are used to provide spatio-temporal information during the restoration process of I^{LQ}_t [30]. At the given timestep t, the proposed method produces a restored frame $\hat{I}^{HQ}_t$, which must be as close as possible to the original uncompressed frame I^{HQ}_t.

The proposed method is shown in Figure 3. The compressed frame I^{LQ}_t to be restored is first decomposed into multi-scale frequency bands $LP^{LQ}_{t,[L-1:0]}$ using the Laplacian decomposition. Then, the MRLN uses the spatio-temporal information extracted from the sequence of compressed frames

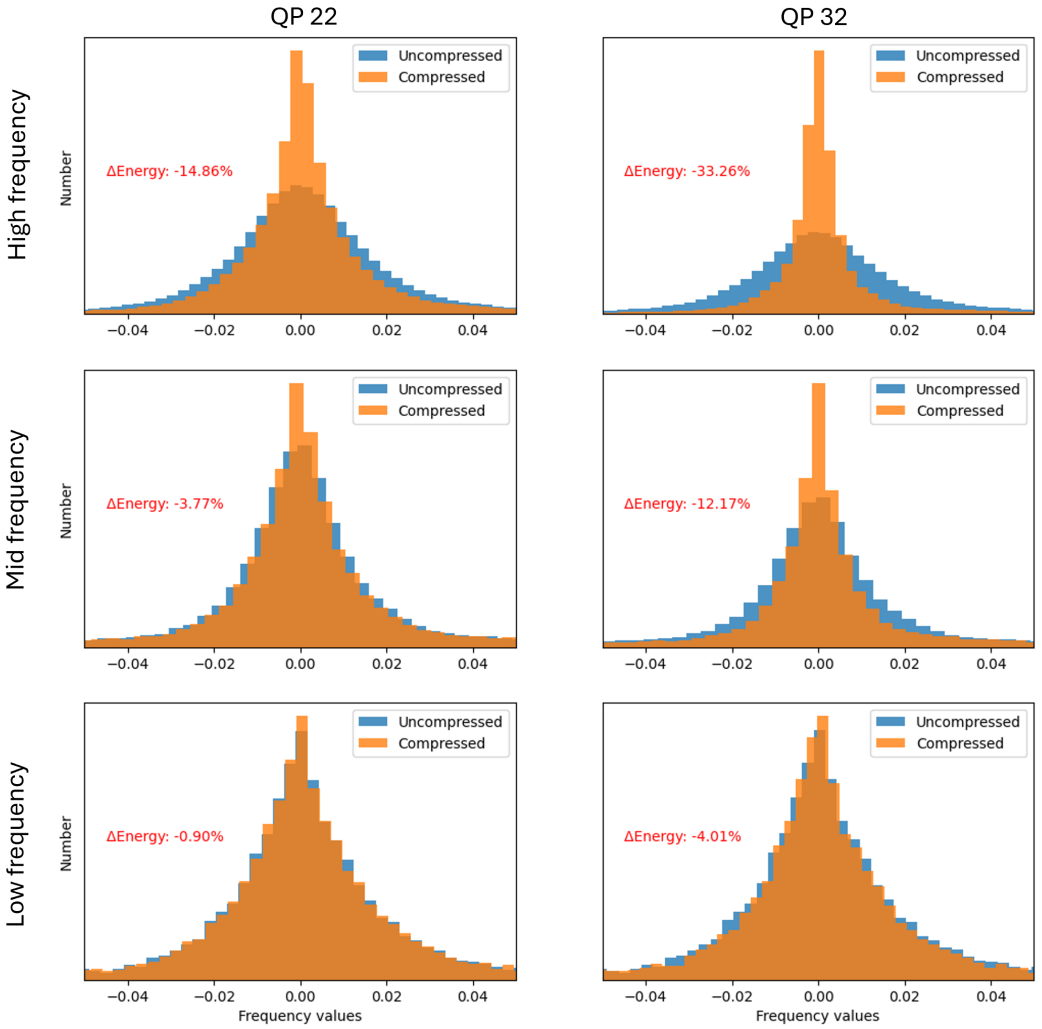


Fig. 2. Histograms of frequency bands of compressed and uncompressed frames. The standard deviation of the distribution of compressed frequency bands (orange) is lower than the one of uncompressed bands (blue), revealing the loss of some frequency components. The distributions of high-frequency bands are more affected than the others. The distributions are altered more using higher QP values. The energy of a frequency band is the sum of its squared values, while ΔEnergy is the percentage of energy loss with respect to the uncompressed frames. This example is computed on frame *200* of *BasketballDrive* sequence [12].

$I^{LQ}_{[t-N:t+N]}$ and the frequency bands $LP^{LQ}_{t,[L-1:0]}$ of the compressed frame I^{LQ}_t to produce the restored frequency bands $\hat{LP}^{HQ}_{t,[L-1:0]}$, i.e. the frequency bands of the restored frame $\hat{I}^{HQ}_t$. Finally, the restored frequency bands $\hat{LP}^{HQ}_{[L-1:0]}$ are recomposed to obtain the restored frame $\hat{I}^{HQ}_t$ using the Laplacian reconstruction. By explicitly leveraging the multi-scale frequency representation provided by the Laplacian decomposition, we can align the MRLN processing levels with the corresponding Laplacian decomposition levels. This alignment enables frequency-specific processing, allowing MRLN to apply distinct transformations to process each frequency band. As a result, the restoration

Table 1. Analysis of the HEVC Compression Effect on Different Frequency Bands of Compressed Frames

	QP 22			QP 27			QP 32		
	ΔEnergy	ΔPSNR	ΔSSIM	ΔEnergy	ΔPSNR	ΔSSIM	ΔEnergy	ΔPSNR	ΔSSIM
High freq.	−8.71	7.48	0.036	−15.59	6.31	0.054	−24.00	5.41	0.078
Mid freq.	−2.50	1.28	0.012	−5.23	1.54	0.023	−9.66	1.84	0.041
Low freq.	−0.67	0.02	0.003	−1.38	0.46	0.007	−2.80	0.67	0.020

The quality improvement (ΔPSNR and ΔSSIM) is obtained by replacing a specific band of compressed frames with the one of uncompressed frames. High-frequency bands suffer higher energy loss. Such energy loss increases as QP increases. Replacing the frequency bands that suffered higher energy loss leads to the best quality improvement.

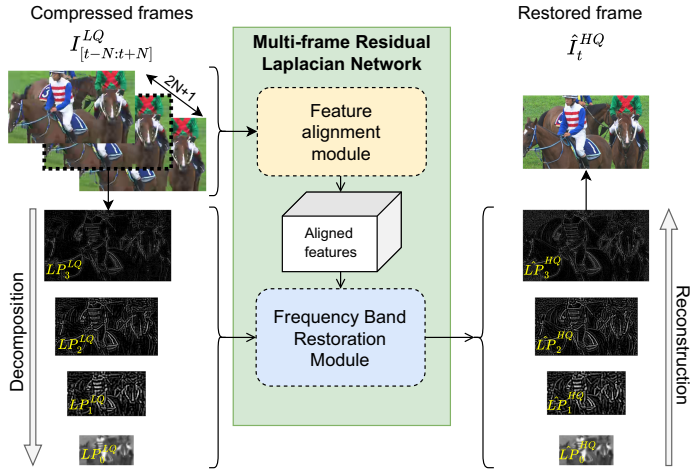


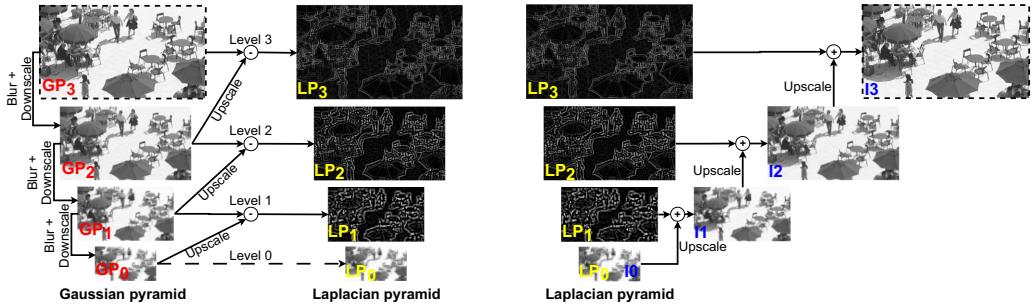
Fig. 3. Overview of the proposed method. The compressed frame I^{LQ}_t is decomposed into frequency bands $LP^{LQ}_{t,[L:0]}$ using the Laplacian decomposition. Then, the Multi-frame Residual Laplacian Network (MRLN) removes the compression artifacts from $LP^{LQ}_{t,[L:0]}$ to obtain the restored frequency bands $\hat{LP}^{HQ}_{[L:0]}$. Finally, $\hat{LP}^{HQ}_{[L:0]}$ are recomposed to obtain the restored frame $\hat{I}^{HQ}_t$ using the Laplacian reconstruction. The compressed frame I^{LQ}_t to be restored is highlighted with a dashed border.

process becomes more effective, as each MRLN processing level adapts to the distinct characteristics and patterns of compression artifacts that affect a specific frequency band.

3.3 Frequency Band Decomposition and Reconstruction

The Laplacian decomposition is a technique to decompose an image into multi-scale frequency bands. It is based on the idea that images can be represented by a sum of frequency bands at different resolution scales. We use the Laplacian decomposition of the compressed frames to perform an analysis of the compression artifacts at multiple resolution scales.

Figure 4 shows the Laplacian decomposition and the respective reconstruction process using $L = 4$ levels. During the decomposition process, shown in Figure 4(a), the input image is progressively decomposed into frequency bands. The Gaussian pyramid *GP* is computed first: the input image is blurred using a low-pass filter (e.g., a Gaussian filter) and downscaled to half of the resolution. This process is repeated for a given number of iterations (three in the figure). With this operation, high-frequency components are progressively separated from the image, leaving a lower resolution



(a) Decomposition (b) Reconstruction

Fig. 4. Overview of the Laplacian decomposition and reconstruction process. The decomposition process decomposes the input image into multi-scale frequency bands. The reconstruction process recomposes the bands to obtain the input image. The input image is reported with dashed borders.

image representation that retains only the coarse structures. Once the Gaussian pyramid GP is obtained, the Laplacian pyramid LP can be constructed as follows. The lowest level of the Laplacian pyramid LP_0 corresponds to the lowest level of the Gaussian pyramid GP_0. Then, the upper level of the Laplacian pyramid LP_1 is obtained by upscaling GP_0 and subtracting it from GP_1. This process is repeated for the same number of iterations as for the Gaussian pyramid. At the end of the decomposition process, the Laplacian pyramid LP is obtained. During the reconstruction process, illustrated in Figure 4(b), the frequency bands contained in the Laplacian pyramid LP are progressively recomposed to obtain the original input image. The first reconstruction level I_0 is represented by the lowest level of the Laplacian pyramid LP_0. Then, the upper reconstruction level I_1 is obtained by upscaling I_0 and adding it to LP_1. This process is repeated for each level l in the Laplacian pyramid LP until the input image is reconstructed.

In the proposed method, the Laplacian decomposition (Figure 4(a)) is used on the compressed frame I_t^{LQ} to obtain the frequency bands $LP_{t,[L-1:0]}^{LQ}$, while the Laplacian reconstruction (Figure 4(b)) is used on the restored frequency bands $\hat{LP}_{t,[L-1:0]}^{HQ}$ to gradually recompose them and obtain the restored frame $\hat{I}_t^{HQ}$.

3.4 MRLN

The MRLN is ad-hoc designed to produce the restored frequency bands $\hat{LP}_{t,[L-1:0]}^{HQ}$ by analyzing the sequence of compressed frames $I_{[t-N:t+N]}^{LQ}$ and the frequency bands $LP_{t,[L-1:0]}^{LQ}$ of the compressed frame I_t^{LQ} to be restored. MRLN is composed of two modules: the Frame Alignment module and the **Frequency Band Restoration Module (FbRm)**.

The Frame Alignment module receives the sequence of compressed frames $I_{[t-N:t+N]}^{LQ}$ as input and produces the corresponding aligned feature maps as output. It exploits the STDF module [9], which uses an encoder–decoder architecture followed by a deformable convolutional layer [8] to perform the alignment. Readers can refer to the work by Deng et al. [9] for further details about this module.

The FbRm receives the aligned feature maps and the frequency bands $LP_{t,[L-1:0]}^{LQ}$ of the compressed frame I_t^{LQ} to be restored as input and produces the restored frequency bands $\hat{LP}_{t,[L-1:0]}^{HQ}$ as output. As shown in Figure 5, the FbRm is implemented using an encoder–decoder architecture composed of multi-scale processing levels. Each level contains a tunable number of convolutional units. Each

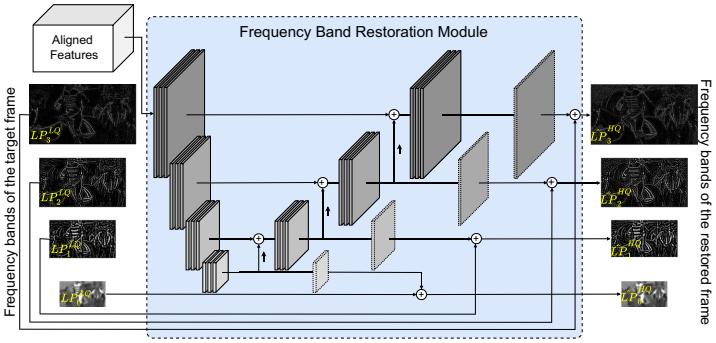


Fig. 5. The Frequency band Restoration module. The different levels are highlighted by colors: the higher the darker. Each unit is composed of a convolutional layer and a ReLU activation. The dotted units are convolutional layers to convert features into frequency residuals, thus they do not have any activation. Downscaling is performed using strided convolutions, while upscaling is achieved either using upconvolutions or bicubic interpolation.

convolutional unit contains a convolutional layer followed by the ReLU activation, except for the convolutional units to convert features into frequency residuals (dotted borders in the figure) that do not have any activation. Downscaling is performed using strided convolutions, while upscaling can be achieved by using either upconvolutions or bicubic interpolation (we discuss this in Section 3.6). The level l analyzes and progressively refines the aligned feature maps to remove the compression artifacts at its resolution scale to obtain the corresponding restored frequency bands $\hat{LP}^{HQ}_{t,l}$.

Given the frequency bands $LP^{LQ}_{t,[L-1:0]}$ of the compressed frame I^{LQ}_t, the FbRm computes the residual representation of the restored frequency bands $\hat{LP}^{HQ}_{t,[L-1:0]}$ as follows:

$$\hat{LP}^{HQ}_{t,l} = LP^{LQ}_{t,l} + \text{FbRm}_{t,l} \tag{1}$$

where $\text{FbRm}_{t,l}$ is the output of FbRm at level l when restoring the compressed frame I^{LQ}_t.

3.5 Loss Function

We aim to specialize each processing level of the MRLN in removing the compression artifacts at its corresponding resolution scale. Given the frequency bands $LP^{HQ}_{t,[L-1:0]}$ of the original uncompressed frame I^{HQ}_t and the frequency bands $\hat{LP}^{HQ}_{t,[L-1:0]}$ of the restored frame $\hat{I}^{HQ}_t$ produced by MRLN using the sequence of compressed frames $I^{LQ}_{[t:N,t+N]}$, we progressively recompose the frequency bands using the Laplacian reconstruction and compute a loss function between each reconstruction level $\hat{I}^{HQ}_{t,l}$ and $I^{HQ}_{t,l}$ [48]. The resulting loss function $\mathcal{L}$, which we refer to as Laplacian loss, is defined as:

$$\mathcal{L} = \frac{1}{L}\sum_{l=0}^{L-1} \|I^{HQ}_{t,l} - \hat{I}^{HQ}_{t,l}\|^2_2 \tag{2}$$

where L is the number of processing levels, $I^{HQ}_{t,l}$ is the reconstruction at level l of the original uncompressed frame I^{HQ}_t, $\hat{I}^{HQ}_{t,l}$ is the reconstruction at level l of the restored frame $\hat{I}^{HQ}_t$, and $\|\cdot\|^2_2$ is the MSE. Note that $\hat{I}^{HQ}_{t,L} = \hat{I}^{HQ}_t$, i.e. the highest level in the Laplacian reconstruction corresponds to the restored frame. Thus, when $l = L-1$, the Laplacian loss computes the MSE between the restored and the uncompressed frames, which corresponds to the standard reconstruction loss.

Table 2. Configuration Details of the Different Versions of the Proposed Method

Name	Frame Alignment module	Frequency band Restoration module			Params	Orientation
	Conv. kernels	Conv. kernels	Conv. layers	Upscaling method		
XS	24	24	2	Bicubic interp.	213K	Efficiency
S	24	32	2	Bicubic interp.	278K	Efficiency
M	32	48	2	Upconvolutions	654K	Balance
L	32	64	2	Upconvolutions	1,000K	Effectiveness
XL	32	64	6	Upconvolutions	1,517K	Effectiveness

The convolutional kernels and the number of convolutions refer to each level in the multi-frame residual Laplacian network.

Applying the loss to the Laplacian reconstruction levels, as opposed to the frequency band representation, allows considering the whole frame content during the comparison since each reconstruction level represents an approximation of the restored frame. Consequently, it avoids tuning the weights for the contribution of the frequency bands at each scale. It also helps the network discriminate the distortions introduced by compression artifacts from the structures of the actual frame content.

3.6 Model Complexity Control

Controlling the model complexity is fundamental to balancing the tradeoff between efficiency and effectiveness when deploying deep learning-based methods in application scenarios having different needs and varying resource availability. Inspired by [36], in the proposed method, we balance this tradeoff by controlling: (1) the number of convolutional units within each level of the MRLN; (2) the number of kernels in each convolutional unit; and (3) the approach used for the upscaling operation (e.g., bicubic interpolation or learnable upconvolutions).

In this way, we can keep the main structure of the network fixed, and instantiate various versions by simply changing these three parameters. Some proposed model configurations are reported in Table 2. For simplicity, we give different names to these models depending on the resulting number of parameters. The eXtra-Small (XS) and Small (S) configurations are efficiency-oriented and characterized by a low computational complexity. The Medium (M) configuration represents a balanced tradeoff between efficiency and effectiveness. Finally, the Large (L) and eXtra-Large (XL) configurations are effectiveness-oriented, as they are more complex and can learn better transformations to separate the frequency bands to better remove the compression artifacts, leading to better restoration results.

4 Experiments

4.1 Setup

We conduct all the experiments using a machine equipped with Ubuntu 22.04 LTS, Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz, 32 GB of RAM, and an NVidia GeForce GTX 1080 GPU.

4.1.1 Dataset. Following previous works [10, 12, 27, 47], we conduct the experiments on the MFQEv2 dataset [12], which contains 108 training sequences and 18 test sequences coming from the datasets of Xiph.org,[1] Video Quality Experts Group (VQEG),[2] and Joint Collaborative Team on Video Coding (JCT-VC) [3]. The video sequences have different resolutions, ranging from 352×240 pixels to $2{,}560 \times 1{,}600$ pixels. In particular, the test sequences are divided into five classes depending

[1]Xiph.org. Xiph.org video test media. https://media.xiph.org/video/derf.
[2]VQEG. Video Quality Experts Group. https://vqeg.org/video-datasets-and-organizations.aspx.

on the respective resolution: Class A (2,650 × 1,600), Class B (1,920 × 1,080), Class C (832 × 480), Class D (416 × 240), and Class E (1,280 × 720). We apply HEVC compression to the videos using HM16.5 [33] under LDP configuration at five different QPs, i.e. 22, 27, 32, 37, and 42.

4.1.2 Network Settings. As reported in Table 2, we design five different model configurations. For efficiency-oriented models, i.e. XS and S, we use plain bicubic interpolation as the upsampling method because it allows considerably reducing the inference time. Conversely, for balance-oriented and effectiveness-oriented models, i.e. M, L and XL, we adopt upconvolutions as they demonstrate better restoration performance in our experiments. We set the number of levels in the MRLN to 4 for all the models. The exploration of other cardinalities is reported in the ablation study in Section 4.5.

4.1.3 Training Details. We train all our models for a total of 1,000 epochs. We set the batch size to 8 and use the Adam optimizer [22] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is set to 1e-4, and it is decreased by a factor of 10 after 400 and 800 epochs, respectively. We increase the number of training samples by using randomly cropping patches of size 64 × 64 pixels, augmented with random horizontal and vertical flipping. The temporal window size is set to 7 (i.e., three previous and three subsequent frames are used). The optimization process is guided using the Laplacian loss function described in Equation (2). We train the models at QP 37 from scratch and fine-tune the models obtained after 600 epochs on the other QPs for the remaining 400 epochs.

4.1.4 Evaluation Metrics. Following previous works [9, 12, 44], we evaluate the restoration quality using ΔPSNR and ΔSSIM, which measure the increase in **Peak Signal-to-Noise Ratio (PSNR)** [16] and in **Structural Similarity Index (SSIM)** [41] of the restored frames with respect to the compressed frames. ΔPSNR is computed as PSNR($\hat{I}^{HQ}$, I^{HQ}) − PSNR(I^{LQ}, I^{HQ}), where I^{LQ} is the compressed frame, $\hat{I}^{HQ}$ is the restored frame, and I^{HQ} is the original uncompressed frame. ΔSSIM is computed accordingly. We use **Bjontegaard Delta Rate (BD-Rate)** [2] reduction to evaluate the rate-distortion performance, which is computed using HEVC as a reference. For the quality fluctuation assessment, we use the **Peak-Valley Difference (PVD)** [44] of the frame-level PSNR, which computes the average difference between peak values and their nearest valley values, and the frame-level PSNR STD [44]. We compute these metrics on the luminance component (Y channel) in YUV/YCbCr space.

We evaluate the efficiency performance using **FLOating Point operations (FLOPs)**, model parameters, and runtime, which is measured in terms of milliseconds required to restore a single frame.

4.2 Contribution of Frequency Band Restoration on the Restoration Quality

We investigate the impact of restoring specific frequency bands of the compressed frames on the restoration quality. In this experiment, we consider the results obtained by our XS model on the MFQEv2 [12] testset compressed at QP 37. The distributions of the obtained ΔPSNR and ΔSSIM values, defined in Section 4.1.4 and filtered through Kernel Density Estimation [32], are shown in Figure 6.

In Figure 6(a), we report the results obtained by restoring the frequency bands at individual levels of the Laplacian pyramid. The restoration of the frequency bands at the lowest level, i.e. level 0, leads to a quality improvement of 0.11 and 0.067 in ΔPSNR and ΔSSIM, respectively. Restoring the frequency bands at higher levels leads to better quality improvement. The best quality improvement is achieved by restoring the frequency bands at the highest level, i.e. level 3, improving the metrics by 0.4 and 0.8, respectively.

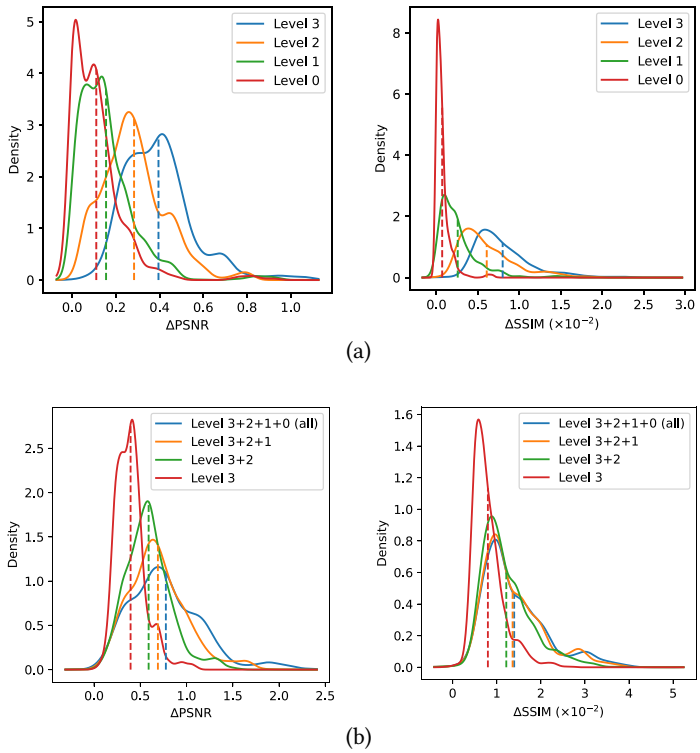


Fig. 6. Distribution of ΔPSNR and ΔSSIM ($\times 10^{-2}$) values obtained when specific frequency bands are restored by our XS model. In (a), the frequency bands at a specific level of the Laplacian pyramid are restored. In (b), the frequency bands at specific levels of the Laplacian pyramid are progressively restored, starting from the highest level. The dotted lines represent the mean value of the distribution.

In Figure 6(b), we report the results obtained by progressively restoring the frequency bands until a specific level of the Laplacian pyramid, starting from the highest level, i.e. level 3. Concerning ΔPSNR, restoring only the frequency bands at level 3 contributes by 51% to the average improvement obtained by restoring the frequency bands at all the levels. Restoring also the frequency bands at level 2 increases the contribution by 25%, and further restoring level 1 increases it by 13%. The remaining 11% is given by restoring also level 0. Instead, for ΔSSIM, restoring only the frequency bands at level 3 contributes by 57% to the average improvement, while restoring the other lower levels increases it by 30% and 10%, respectively. Here, level 0 only contributes by 3%.

4.3 Performance of the Different Versions of Our Method

We compare the performance of the different versions of our method, as described in Sections 3.6 and 4.1.2, in terms of restoration quality and efficiency performance. The results are reported in Table 3. We can observe that the restoration quality increases as the number of model parameters increases, at the expense of efficiency. The S model has only 65 K more parameters than the XS model (i.e., 31% more parameters), while their efficiency is very similar. The S model obtains a considerable improvement in restoration quality with respect to the XS model: 0.04 in both ΔPSNR and ΔSSIM, on average, considering all the QPs. Instead, comparing the L and XL models, with the latter having 517 K more parameters than the former (i.e., 51.7% more parameters), we can observe an improvement in restoration quality of only 0.02 in both ΔPSNR and ΔSSIM. This suggests that

Table 3. Performance of the Different Versions of Our Method

Name	Params	Quantization parameter (QP)					Resolution									
		QP 22	QP 27	QP 32	QP 37	QP 42	416 × 240		832 × 480		1,280 × 720		1,920 × 1,080		2,560 × 1,600	
							GFLOPs	Runtime	GFLOPs	Runtime	GFLOPs	Runtime	GFLOPs	Runtime	GLOPs	Runtime
XS	213K	0.62/0.38	0.70/0.62	0.74/0.95	0.77/ 1.41	0.78/1.98	8.41	19	45.00	73	100.94	174	227.11	423	448.60	950
S	278K	0.66/0.40	0.75/0.65	0.79/1.00	0.80/1.45	0.82/2.04	10.52	20	56.29	77	126.26	183	284.08	448	561.15	1,016
M	654K	0.72/0.44	0.81/0.71	0.86/1.09	0.87/1.55	0.86/2.16	21.32	24	104.05	96	255.82	228	575.59	547	1136.97	1,200
L	1,000K	0.75/0.45	0.85/0.74	0.89/1.12	0.89/1.60	0.88/2.21	31.70	27	169.58	107	380.37	254	855.84	608	1690.55	1,330
XL	1,517K	0.77/0.46	0.87/0.76	0.92/1.16	0.92/1.64	0.90/2.23	46.73	33	249.99	132	560.73	313	1261.65	750	2091.34	1,642

Restoration performance is reported as ΔPSNR/ΔSSIM ($\times 10^{-2}$). The higher the better. Efficiency is reported as model parameters, GFLOPs, and runtime. Runtime is expressed in milliseconds.

further increasing the complexity of the model may not bring any significant improvement in restoration quality. As shown, the M model represents a balanced tradeoff between effectiveness and efficiency.

4.4 Comparison with State-of-the-Art Methods

We compare the proposed method with state-of-the-art methods for HEVC-compressed video restoration. In particular, we use MFQE2.0 [12], PSTQE [10], STDF [9], Fast-MFQE [7], STLVQE [29], MFQE [44], STDF-R3L [9], RFDA [47], STDR [27], and STAGE-Net [21]. We divide these methods into two groups depending on the number of parameters. The low-complexity group includes MFQE2.0 [12], PSTQE [10], STDF [9], Fast-MFQE [7], STLVQE [29], and our XS model. The number of parameters of these models ranges from about 200 K to 450 K. The high-complexity group includes MFQE [44], STDF-R3L [9], RFDA [47], STDR [27], and our XL model, having a number of parameters ranging from 1,200 K to 1,500 K. We include STAGE-Net [21] in the high-complexity group, even though its number of parameters, which is 6,990 K, is considerably higher than the other methods in this group.

4.4.1 Restoration Quality. The frame restoration quality results are reported in Table 4.

Following previous works [9, 12, 44], for each method, we report the detailed performance obtained on each sequence of the MFQEv2 [12] testset compressed at QP 37, and the average performance for the other QPs obtained by averaging the results of all the sequences at a given QP. The QP is reported in the first column of the table.

Considering the average performance of the methods within the low-complexity group, we can see that our XS model outperforms all the others. More in detail, at QP 37, our XS model obtains the best ΔPSNR value on 13 out of 18 sequences, while in the case of ΔSSIM, this number increases to 17. It obtains an average quality improvement of 0.04 and 0.09 in ΔPSNR and ΔSSIM, respectively, compared to STDF [9], which represents the second-best method. In addition, the performance degradation of our XS model as QP decreases from 42 to 22 is less abrupt than those of other methods.

Regarding the high-complexity group, STDR [27] is the best-performing method. We can see that our XL model is, on average, the second-best method in terms of ΔPSNR, outperforming STDR [27] on two sequences, i.e. *Kimono* and *RaceHorses*. Considering ΔSSIM, our XL model achieves superior performance than STAGE-Net [21] at QP 37. On average, our XL model outperforms RFDA [47]. We can notice that RFDA [47] obtains good results at QP 37, but shows a considerable performance degradation when evaluated on other QPs. In contrast, the performance of our XL model is more stable as QP varies.

We present in Figure 7 a qualitative comparison among our XS and XL models and the state-of-the-art methods using some frames from the sequences *BQMall*, *BasketballDrill*, and *RaceHorses*

Table 4. Restoration Quality Comparison with State-of-the-Art Methods in Terms of ΔPSNR/ΔSSIM ($\times 10^{-2}$) on the Test Sequences at Different QPs

QP	Class	Sequence	Low-complexity						High-complexity					
			MFQE2.0 [12]	PSTQE [10]	STDF [9]	Fast-MFQE [7]	STLVQE [29]	Ours (XS)	MFQE [44]	STDF-R3L [9]	RFDA [47]	STDR [27]	STAGE-Net [21]	Ours (XL)
37	A	*Traffic*	0.59/1.02	0.64/1.04	0.65/1.04	0.61/**1.23**	0.41/0.94	**0.70**/1.17	0.50/0.90	0.74/1.22	0.80/1.28	**0.85**/**1.34**	0.79/1.21	0.81/1.32
		PeopleOnStreet	0.92/1.57	1.08/1.68	**1.18**/1.82	0.97/1.67	0.68/1.35	**1.18**/**1.90**	0.80/1.37	1.25/2.02	1.44/2.22	**1.53**/**2.34**	1.21/1.99	1.39/2.16
	B	*Kimono*	0.55/1.18	0.69/1.36	0.77/1.47	0.66/1.23	0.38/1.05	**0.95**/**1.63**	0.50/1.13	0.91/1.73	1.02/1.86	1.05/**1.91**	0.82/1.59	**1.08**/1.80
		ParkScene	0.46/1.23	0.49/1.21	0.54/1.32	0.53/1.33	0.33/0.80	**0.59**/**1.43**	0.39/1.03	0.61/1.45	0.64/1.58	0.70/**1.69**	**0.71**/1.39	0.67/1.62
		Cactus	0.50/1.00	0.62/1.15	0.70/1.23	0.64/1.16	0.43/0.85	**0.73**/**1.24**	0.44/0.88	0.77/1.42	0.83/1.49	**0.85**/**1.52**	0.79/1.50	0.82/1.42
		BQTerrace	0.40/0.67	0.50/0.87	**0.58**/0.93	0.52/0.86	0.42/0.83	0.55/**0.96**	0.27/0.48	0.63/1.05	0.65/1.06	**0.72**/**1.23**	0.64/1.09	0.67/1.19
		BasketballDrive	0.47/0.83	0.60/1.04	0.66/1.07	**0.74**/0.91	0.49/1.00	**0.74**/**1.17**	0.41/0.80	0.80/1.34	0.87/1.40	**0.94**/**1.50**	0.79/1.12	0.92/1.38
	C	*RaceHorses*	0.39/0.80	0.40/0.88	0.48/1.09	**0.53**/0.93	0.39/0.94	0.49/**1.17**	0.34/0.55	0.52/1.16	0.48/1.23	0.55/1.53	0.59/1.38	**0.63**/**1.56**
		BQMall	0.62/1.20	0.74/1.44	0.90/1.61	0.72/1.23	0.50/1.25	**0.92**/**1.63**	0.51/1.03	0.90/1.86	1.09/1.97	**1.19**/**2.12**	0.95/1.81	1.12/1.99
		PartyScene	0.36/1.18	0.51/1.46	**0.60**/1.60	0.44/1.31	0.34/1.15	0.58/**1.78**	0.22/0.73	0.67/1.92	0.66/1.88	**0.79**/**2.24**	0.72/1.79	0.71/2.13
		BasketballDrill	0.58/1.20	0.66/1.27	0.70/1.26	0.63/1.26	0.57/1.37	**0.78**/**1.49**	0.48/0.90	0.79/1.46	0.88/1.67	**0.99**/**1.89**	0.83/1.52	0.88/1.67
	D	*RaceHorses*	0.59/0.60	1.44/1.43	0.73/1.75	0.68/1.47	0.44/1.08	**0.74**/**1.79**	0.51/1.13	0.81/1.94	0.85/2.11	**0.95**/**2.44**	0.81/1.96	**0.95**/2.26
		BQSquare	0.34/0.65	0.79/**1.14**	**0.91**/1.13	0.47/0.68	0.37/0.75	0.65/0.97	−0.01/0.15	0.80/1.21	1.05/1.39	**1.28**/**1.72**	0.96/1.31	0.83/1.12
		BlowingBubbles	0.53/1.70	0.62/1.95	**0.68**/1.96	0.61/1.89	0.37/1.23	0.67/**2.08**	0.39/1.20	0.74/2.24	0.78/2.40	**0.86**/**2.67**	0.81/1.69	0.76/2.31
		BasketballPass	0.73/1.55	0.85/1.75	0.95/1.82	0.88/1.67	0.48/1.28	**0.99**/**1.95**	0.63/1.38	1.05/2.11	1.12/2.23	**1.26**/**2.52**	1.05/1.79	1.21/2.35
	E	*FourPeople*	0.73/0.95	**0.95**/1.12	0.92/1.07	0.87/0.97	0.69/1.02	0.90/**1.13**	0.66/0.85	1.00/1.29	**1.13**/1.36	1.12/**1.37**	1.01/1.08	1.03/1.27
		Johnny	0.60/0.68	0.75/**0.85**	0.69/0.73	0.71/0.73	**0.93**/0.64	0.75/0.84	0.55/0.55	0.88/1.02	**0.90**/0.94	0.89/**0.98**	0.87/0.91	0.87/0.86
		KristenAndSara	0.75/0.85	0.93/0.91	0.94/0.89	0.86/0.88	0.74/0.84	**0.97**/**0.97**	0.66/0.75	1.05/1.11	**1.19**/**1.15**	1.18/1.14	1.05/1.03	1.12/1.06
		Average	0.56/1.09	0.69/1.25	0.75/1.32	0.67/1.18	0.49/1.03	**0.77**/**1.41**	0.46/0.88	0.83/1.53	0.91/1.62	**0.98**/**1.79**	0.87/1.47	0.92/1.64
42		Average	0.59/1.65	0.69/1.86	0.71/1.87	-	-	**0.78**/**1.98**	0.44/1.30	0.74/1.99	0.82/2.20	**0.95**/**2.47**	-	0.90/2.23
32		Average	0.52/0.68	0.67/0.83	0.73/0.87	0.63/0.69	-	**0.74**/**0.95**	0.43/0.58	0.86/1.07	0.87/1.07	**0.99**/**1.24**	0.81/1.22	0.92/1.16
27		Average	0.49/0.42	0.63/0.52	0.67/0.53	0.55/0.48	-	**0.70**/**0.62**	0.40/0.34	0.74/0.59	0.82/0.68	**0.97**/**0.81**	0.71/**0.81**	0.87/0.76
22		Average	0.46/0.27	0.55/0.29	0.57/0.30	-	-	**0.62**/**0.38**	0.31/0.19	0.65/0.34	0.76/0.42	**0.87**/0.48	0.65/**0.71**	0.77/0.46

For each row, best results per complexity group in bold, second-best results underlined. The higher the better.

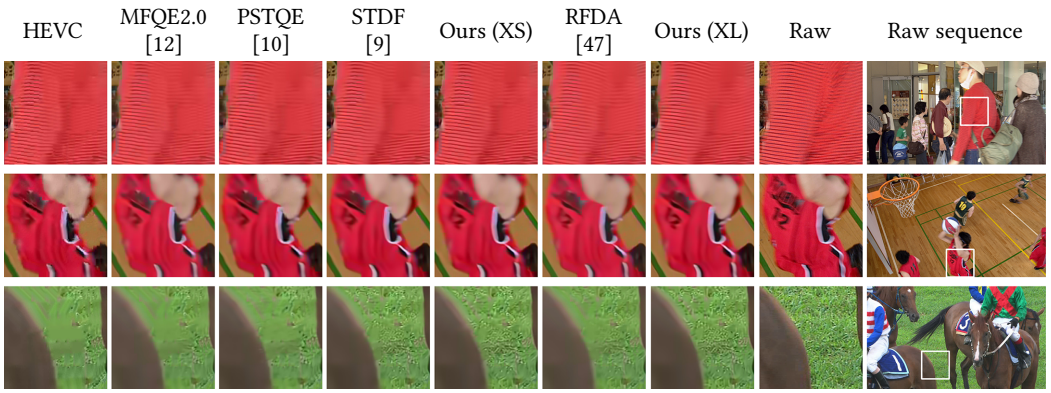


Fig. 7. Qualitative comparison with state-of-the-art methods on sequences compressed at QP 37. Frames from *BQMall* (first row), *BasketballDrill* (first row), and *RaceHorses* (third row). Zoom-in for a better view.

compressed at QP 37. The artifacts introduced by HEVC can be observed in the first column of the figure. We can see that MFQE2.0 [12] and PSTQE [10] do not properly restore the compressed frames, leaving them with notable artifacts, while STDF [9] fails in restoring textured regions. Our XS model produces sharper results, and it is better at restoring textures. Our XL model further improves these results, restoring frames better than RFDA [47].

4.4.2 Rate-Distortion Performance. The results related to the rate-distortion performance measured using BD-Rate [2] reduction with HEVC as a reference are shown in Table 5. In the low-complexity group, our XS model achieves an average 19.07% BD-Rate reduction, considerably outperforming the other methods. More in detail, it performs better on 14 out of 18 sequences. Instead, in the high-complexity group, STDR [27] is the best-performing method, while our XL model obtains the second-best performance by a close margin (2.32) compared to STDR [27], outperforming the latter on two sequences.

Figure 8 provides examples of rate-distortion curves on *Kimono*, *BasketballDrive*, and *Traffic* sequences. We can see that our XL model obtains better PSNR at a given bitrate than the other methods, immediately followed by our XS model.

4.4.3 Quality Fluctuation. Table 6 reports the results of the quality fluctuation assessment. All the analyzed methods reduce the quality fluctuation introduced by HEVC. Our XS model achieves better performance both in terms of PVD [44] and STD [44] compared to the other methods in the low-complexity group, which means that it produces more stable results. Instead, in the context of high-complexity methods, our XL model performs worse than RFDA [47] and STDR [27] at QP 27 and QP 32, it obtains the same PVD results as RFDA [47] at QP 37, and it outperforms RFDA [47] at QP 42.

Figure 9 shows an example of the quality fluctuation of two sequences compressed at QP 42 and restored by MFQE2.0 [12], PSTQE [10], and STDF [9], our XS and XL models. In both the sequences, our XL model places above the others, showing better restoration quality and reduced fluctuation, immediately followed by our XS model.

4.4.4 Efficiency Performance. The comparison of the efficiency performance is shown in Table 7. Within the low-complexity group, our XS model is the most lightweight method, relying on only 213 K parameters. We can also see that it has the lowest GFLOP count considering each tested resolution. Regarding the runtime, the fastest method is MFQE2.0 [12], which however has the

Table 5. Rate-Distortion Performance Comparison with State-of-the-Art Methods in Terms of BD-Rate Reduction (%) on the Test Sequences with HEVC as a Reference

Class	Sequence	Low-complexity				High-complexity				
		MFQE2.0 [12]	PSTQE [10]	STDF [9]	Ours (XS)	MFQE [44]	STDF-R3L [9]	RFDA [47]	STDR [27]	Ours (XL)
A	*Traffic*	16.98	18.57	18.92	**20.32**	14.56	21.19	22.70	**25.96**	22.95
	PeopleOnStreet	15.08	17.80	17.26	**18.80**	13.71	17.42	21.22	**22.77**	21.88
B	*Kimono*	13.34	16.20	19.09	**21.95**	12.60	17.96	22.32	23.47	**24.99**
	ParkScene	13.66	15.89	17.08	**18.76**	12.04	18.10	19.85	**23.21**	21.33
	Cactus	14.84	18.37	20.03	**21.37**	12.78	21.54	21.78	**25.54**	24.32
	BQTerrace	14.72	19.73	20.36	**22.26**	10.95	24.71	24.41	**32.21**	26.89
	BasketballDrive	11.85	15.31	16.86	**18.15**	10.54	16.75	20.24	22.68	**22.86**
C	*RaceHorses*	9.61	9.24	10.37	**11.21**	8.83	**15.62**	14.29	14.14	15.17
	BQMall	13.50	15.73	19.11	**19.56**	11.11	21.12	21.62	**26.08**	24.06
	PartyScene	11.28	16.49	17.52	**17.62**	6.67	22.24	21.22	**25.98**	21.20
	BasketballDrill	12.63	15.20	14.80	**15.26**	10.47	15.94	18.06	**19.65**	18.71
D	*RaceHorses*	11.55	**12.97**	10.37	11.21	10.41	15.26	17.57	**19.25**	19.14
	BQSquare	11.00	23.72	**24.20**	23.74	2.72	33.36	31.65	**39.94**	29.97
	BlowingBubbles	15.20	18.57	19.52	**19.62**	10.73	23.54	22.89	**26.73**	22.44
	BasketballPass	13.43	16.01	17.36	**17.57**	11.70	18.42	20.42	**22.32**	21.82
E	*FourPeople*	17.50	**20.75**	19.94	20.46	14.89	22.91	22.84	**26.70**	23.44
	Johnny	18.57	21.34	21.52	**22.36**	15.94	24.55	23.87	**29.22**	25.26
	KristenAndSara	18.34	18.40	21.98	**23.02**	15.06	23.64	24.47	**28.68**	26.26
	Average	14.06	17.24	18.13	**19.07**	11.41	20.79	21.75	**25.25**	22.93

For each row, best results per complexity group in bold, second-best results underlined. The higher the better.

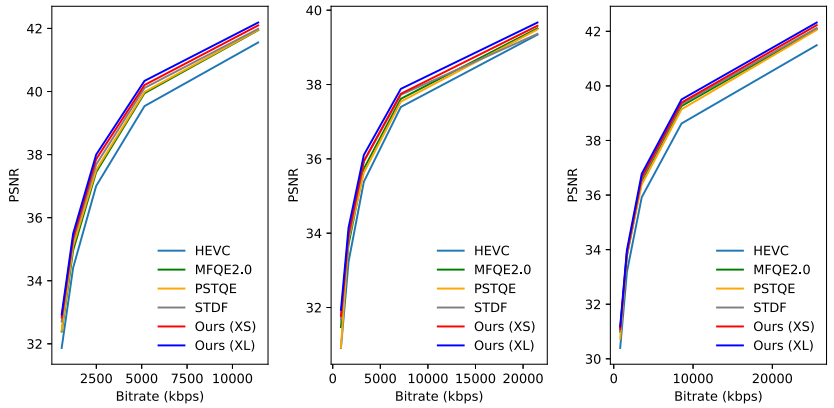


Fig. 8. Rate-distortion curve comparison between HEVC, MFQE2.0 [12], PSTQE [10], STDF [9], and our XS and XL models on *Kimono* (left), *BasketballDrive* (center), and *Traffic* (right). The higher the better. Zoom-in for a better view.

worst effectiveness performance. Our method requires 92% and 18% less time than PSTQE [10] and STDF [9], respectively, to restore a single frame at $1{,}920 \times 1{,}080$ pixel resolution. Concerning the high-complexity group, our XL model has 1,517 K parameters: 271 K fewer parameters than MFQE [44] and 193 K more parameters than STDR [27]. The GFLOP count of our XL model is smaller than STDF-R3L [9] and comparable with RFDA [47]. Nevertheless, it is about 43% faster than RFDA [47] in restoring frames.

Table 6. Quality Fluctuation Comparison with State-of-the-Art Methods in Terms of Average PVD/STD

Method	QP 27	QP 32	QP 37	QP 42
HEVC	1.07/0.83	1.38/0.82	1.42/0.79	1.21/0.74
MFQE2.0 [12]	0.77/0.74	0.98/0.70	0.96/0.67	0.74/0.62
PSTQE [10]	0.70/0.64	0.97/0.65	0.89/0.63	0.72/0.62
STDF [9]	0.68/**0.63**	0.96/0.62	**0.75**/0.61	0.60/**0.59**
Ours (XS)	**0.64**/**0.63**	**0.94**/**0.61**	**0.75**/**0.60**	**0.58**/**0.59**
MFQE [44]	0.84/0.81	1.00/0.77	1.05/0.73	0.82/0.69
RFDA [47]	0.59/0.45	0.77/0.41	0.71/0.39	0.64/0.36
STDR [27]	**0.50**/**0.38**	**0.68**/**0.36**	**0.67**/**0.36**	**0.55**/**0.31**
Ours (XL)	0.61/0.62	0.88/0.60	0.71/0.60	0.56/0.59

For each column, best results per complexity group in bold, second-best results underlined. The lower the better. The first group of methods corresponds to the low-complexity group, while the second one to the high-complexity group.

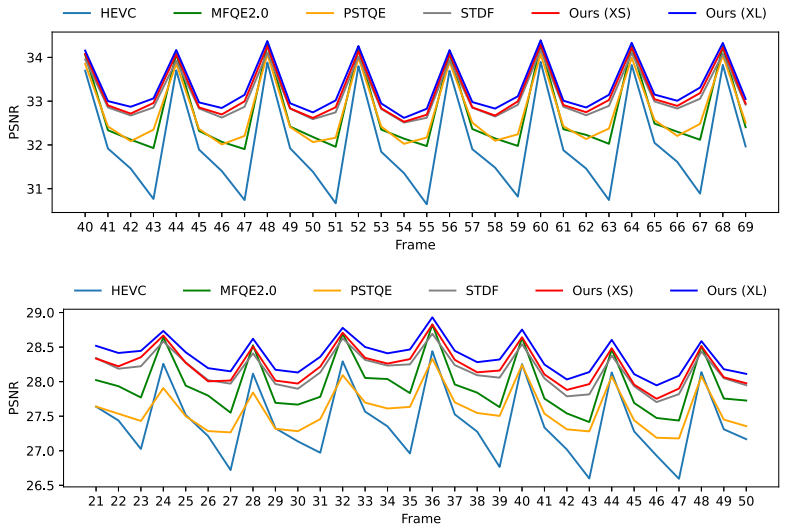


Fig. 9. PSNR curves showing the frame quality fluctuation of HEVC, MFQE2.0 [12], PSTQE [10], STDF [9], and our XS and XL models on *Kimono* (top) and *BQMall* (bottom). The higher the better.

4.5 Ablation Study

We set up ablation experiments to understand the contribution of the different components of the proposed method. The ablation results referring to our XS model, evaluated on sequences compressed at QP 37, are reported in Table 8. In these experiments, the ablated model variants we evaluate have almost the same number of parameters as our XS model (about −2 K/+6K parameters), which is achieved by increasing/decreasing the number of kernels within the convolutional units. Thus, the observed difference in performance is not dependent on this aspect. For a fair comparison, we train all the ablated model variants from scratch, following the setup described in Section 4.1.3.

Table 7. Efficiency Comparison with State-of-the-Art Methods in Terms of Model Parameters, GFLOPs, and Runtime

Method	Params	Resolution									
		416 × 240		832 × 480		1,280 × 720		1,920 × 1,080		2,650 × 1,600	
		GFLOPs	Runtime	GFLOPs	Runtime	GFLOPs	Runtime	GFLOPs	Runtime	GFLOPs	Runtime
MFQE2.0 [12]	255 K	17.42	13	93.18	54	209.01	125	470.27	287	928.92	584
PSTQE [10]	217 K	57.46	172	307.19	692	689.02	1,589	1550.29	5,066	3062.30	15,191
STDF [9]	365 K	19.87	22	106.28	90	238.38	213	536.36	516	1059.48	1,146
Fast-MFQE [7]	243 K	-	-	-	-	-	-	-	-	-	-
STLVQE [29]	448 K	14.23	32	77.67	125	171.54	280	383.27	629	754.98	1,232
Ours (XS)	213K	8.41	19	45.00	73	100.94	174	227.11	423	448.60	950
MFQE [44]	1,788 K	-	-	-	-	-	-	-	-	-	-
STDF-R3L [9]	1,275 K	64.98	28	347.62	116	779.71	267	1754.34	629	3465.36	1,335
RFDA [47]	1,270 K	45.47	58	243.46	231	546.16	541	1229.05	1,275	2428.08	2,723
STDR [27]	1,324 K	-	-	-	-	-	-	-	-	-	-
STAGE-Net [21]	6,990 K	-	-	-	-	-	-	-	-	-	-
Ours (XL)	1,517K	46.73	33	249.99	132	560.73	313	1261.65	750	2091.34	1,642

Runtime is expressed in milliseconds. The first group of methods corresponds to the low-complexity group, while the second one to the high-complexity group.

Table 8. Ablation Study on Network Design, Loss Function, and Number of Processing Levels

Exp.	Network design for MRLN		Loss function		Proc. levels	ΔPSNR	ΔSSIM	Params	GFLOPs	Runtime
	Encoder-decoder	Laplacian dec.	MSE loss	Laplacian loss						
E1	✓		✓		4	0.69	1.34	213K	225.48	418
E2	✓	✓	✓		4	0.75	1.40	213K	227.11	423
E3	✓	✓		✓	4	0.77	1.41	213K	227.11	423
E4	✓	✓		✓	3	0.76	1.38	211K	250.34	433
E5	✓	✓		✓	5	0.74	1.37	219K	215.63	419

The results refer to variants of our XS model applied to sequences compressed at QP 37. Restoration performance is reported as ΔPSNR/ΔSSIM ($\times 10^{-2}$). The higher the better. Efficiency is reported as model parameters, GFLOPs, and runtime. GFLOPs and runtime are computed at 1,920 × 1,080 frame resolution. Runtime is expressed in milliseconds. E3 corresponds to our XS model configuration.

The baseline model (E1) is represented by an encoder–decoder model that directly produces the restored frames and is trained using a plain MSE loss function, which is applied to the highest level of the decoder.

In E2, we evaluate the contribution of exploiting the different processing levels of the decoder to produce the frequency bands of the restored frames, which are then recomposed using the Laplacian reconstruction. We achieve this by computing the Laplacian decomposition of the compressed target frame and using the processing levels of the decoder to compute the residual representation of the restored frequency bands. Compared to directly producing the restored frames (E1), producing the restored frequency bands and recomposing them leads to a reconstruction quality improvement of 0.06 in both ΔPSNR and ΔSSIM values. This experiment shows the advantage of introducing the Laplacian decomposition in the restoration process, which enables a considerable improvement in performance with a small impact on model complexity (+0.72% GFLOPs) and only adding a negligible overhead in runtime (+1.20% milliseconds).

In E3, we evaluate the contribution of using the Laplacian loss (Equation (2)) instead of plain MSE loss to specialize each processing level in removing the compression artifacts at its corresponding resolution scale. The use of the Laplacian loss leads to a restoration quality improvement of 0.02 and 0.01 in ΔPSNR and ΔSSIM, respectively, with respect to E2 where we use the plain MSE loss. This experiment shows that further specializing each processing level of MRLN in removing the compression artifacts at its corresponding resolution scale allows improving the quality of the results.

In E4 and E5, we evaluate the contribution of using different numbers of processing levels. Using too many levels would produce frequency bands with too little signal, making it difficult to identify the compression artifacts. On the other hand, using too few levels would prevent the network from capturing relevant frame components, limiting its effectiveness. The results show that using four processing levels, i.e., $L = 4$, leads to the best performance, while decreasing the levels to three (E4) or increasing them to five (E5) causes a drop in performance, especially in ΔSSIM. Note that GFLOPs and runtime in E4 and E5 show an unexpected behavior. This is because changing the number of kernels (e.g., from 24 to 28 in E4 or from 24 to 22 in E5) within convolutional units has a higher impact on computational complexity than changing the number of processing levels (e.g., from 4 to 3 in E4 or from 4 to 5 in E5).

5 Conclusion

In this work, we presented a method that combines the Laplacian decomposition technique with CNNs to effectively and efficiently reduce the artifacts in HEVC-compressed videos. The proposed method leverages the capabilities of the Laplacian decomposition to decompose compressed frames into distinct multi-scale frequency bands, which are restored by the MRLN and finally recomposed to obtain the restored frames. The proposed method is parametrically scalable and can be easily instantiated in different versions to control the tradeoff between efficiency and effectiveness, providing a versatile approach to be used in various scenarios. Specifically, we presented and studied five versions.

Experimental results showed that our efficiency-oriented XS model outperforms state-of-the-art methods with similar model complexity in terms of restoration performance. In addition, it has a faster runtime. Our effectiveness-oriented XL model achieves state-of-the-art performance when compared with methods with similar model complexity, processing videos in a shorter time.

In the future, exploiting the coding information provided by the HEVC encoder could be an interesting direction to further improve our work [14, 34, 35]. In addition, we plan to extend the investigation of the proposed method to other codec standards, such as H.266/VVC [4].

References

[1] Arbind Agrahari Baniya, Tsz-Kwan Lee, Peter W. Eklund, Sunil Aryal, and Antonio Robles-Kelly. 2023. Online video super-resolution using information replenishing unidirectional recurrent model. *Neurocomputing* 546 (2023), 126355. DOI: https://doi.org/10.1016/j.neucom.2023.126355

[2] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. *ITU SG16 Doc. VCEG-M33*.

[3] Frank Bossen. 2013. Common test conditions and software reference configurations. *JCTVC-L1100* 12, 7 (2013), 1.

[4] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764. DOI: https://doi.org/10.1109/TCSVT.2021.3101953

[5] Peter J. Burt and Edward H. Adelson. 1987. The Laplacian pyramid as a compact image code. In *Readings in Computer Vision*. Elsevier, 671–679. DOI: https://doi.org/10.1016/B978-0-08-051581-6.50065-9

[6] Huaian Chen, Yi Jin, Kai Xu, Yuxuan Chen, and Changan Zhu. 2022. Multiframe-to-multiframe network for video denoising. *IEEE Transactions on Multimedia* 24 (2022), 2164–2178. DOI: https://doi.org/10.1109/TMM.2021.3077140

[7] Kemi Chen, Jing Chen, Huanqiang Zeng, and Xueyuan Shen. 2023. Fast-MFQE: A fast approach for multi-frame quality enhancement on compressed video. *Sensors* 23, 16 (2023), 7227. DOI: https://doi.org/10.3390/s23167227

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 764–773. DOI: https://doi.org/10.1109/ICCV.2017.89

[9] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. 2020. Spatio-temporal deformable convolution for compressed video quality enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10696–10703. DOI: https://doi.org/10.1609/aaai.v34i07.6697

[10] Qing Ding, Liquan Shen, Liangwei Yu, Hao Yang, and Mai Xu. 2021. Patch-wise spatial-temporal quality enhancement for HEVC compressed video. *IEEE Transactions on Image Processing* 30 (2021), 6459–6472. DOI: https://doi.org/10.1109/TIP.2021.3092949

[11] Max Ehrlich, Jon Barker, Namitha Padmanabhan, Larry Davis, Andrew Tao, Bryan Catanzaro, and Abhinav Shrivastava. 2024. Leveraging bitstream metadata for fast, accurate, generalized compressed video quality enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1517–1527. DOI: https://doi.org/10.1109/WACV57701.2024.00154

[12] Zhenyu Guan, Qunliang Xing, Mai Xu, Ren Yang, Tie Liu, and Zulin Wang. 2021. MFQE 2.0: A new approach for Multi-Frame quality enhancement on compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 3 (2021), 949–963. DOI: https://doi.org/10.1109/TPAMI.2019.2944806

[13] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R. Martin, Ming-Ming Cheng, and Shi-Min Hu. 2022. Attention mechanisms in computer vision: A survey. *Computational Visual Media* 8, 3 (2022), 331–368. DOI: https://doi.org/10.1007/s41095-022-0271-y

[14] Huiguo He, Hongyang Chao, and Jian Yin. 2022. Compression loss-based spatial-temporal attention module for compressed video quality enhancement. *Neurocomputing* 501 (2022), 75–87. DOI: https://doi.org/10.1016/j.neucom.2022.05.111

[15] Xiaoyi He, Qiang Hu, Xiaoyun Zhang, Chongyang Zhang, Weiyao Lin, and Xintong Han. 2018. Enhancing HEVC compressed videos with a partition-masked convolutional neural network. In *2018 25th IEEE International Conference on Image Processing*, 216–220. DOI: https://doi.org/10.1109/ICIP.2018.8451086

[16] Alain Horé and Djemel Ziou. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, 2366–2369. DOI: https://doi.org/10.1109/ICPR.2010.579

[17] Sudeng Hu, Hanli Wang, and Sam Kwong. 2012. Adaptive quantization-parameter clip scheme for smooth quality in H.264/AVC. *IEEE Transactions on Image Processing* 21, 4 (2012), 1911–1919. DOI: https://doi.org/10.1109/TIP.2011.2176347

[18] Jiawang Huang, Jinzhong Cui, Mao Ye, Shuai Li, and Yu Zhao. 2022. Quality enhancement of compressed screen content video by cross-frame information fusion. *Neurocomputing* 493 (2022), 486–496. DOI: https://doi.org/10.1016/j.neucom.2021.12.092

[19] Zhijie Huang, Jun Sun, and Xiaopeng Guo. 2023. FastCNN: Towards fast and accurate spatiotemporal network for HEVC compressed video enhancement. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 3 (2023), 1–22. DOI: https://doi.org/10.1145/3569583

[20] Yongkai Huo, Qiyan Lian, Shaoshi Yang, and Jianmin Jiang. 2021. A recurrent video quality enhancement framework with multi-granularity frame-fusion and frame difference based attention. *Neurocomputing* 431 (2012), 34–46. DOI: https://doi.org/10.1016/j.neucom.2020.12.019

[21] Nanfeng Jiang, Weiling Chen, Jielian Lin, Tiesong Zhao, and Chia-Wen Lin. 2023. Video compression artifacts removal with spatial-temporal attention-guided enhancement. *IEEE Transactions on Multimedia* 26 (2023), 5657–5669. DOI: https://doi.org/10.1109/TMM.2023.3338087

[22] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980

[23] Chen Li, Li Song, Rong Xie, and Wenjun Zhang. 2022. L0 structure-prior assisted blur-intensity aware efficient video deblurring. *Neurocomputing* 483 (2022), 195–209. DOI: https://doi.org/10.1016/j.neucom.2022.02.013

[24] Dingyi Li, Zengfu Wang, and Jian Yang. 2022. Video super-resolution with inverse recurrent net and hybrid local fusion. *Neurocomputing* 489 (2022), 40–51. DOI: https://doi.org/10.1016/j.neucom.2022.03.019

[25] Feng Li, Yixuan Wu, Anqi Li, Huihui Bai, Runmin Cong, and Yao Zhao. 2024. Enhanced video super-resolution network towards compressed data. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 7 (2024), 1–21. DOI: https://doi.org/10.1145/3651309

[26] Weiyao Lin, Xiaoyi He, Xintong Han, Dong Liu, John See, Junni Zou, Hongkai Xiong, and Feng Wu. 2020. Partition-aware adaptive switching neural networks for post-processing in HEVC. *IEEE Transactions on Multimedia* 22, 11 (2020), 2749–2763. DOI: https://doi.org/10.1109/TMM.2019.2962310

[27] Dengyan Luo, Mao Ye, Shuai Li, Ce Zhu, and Xue Li. 2023. Spatio-temporal detail information retrieval for compressed video quality enhancement. *IEEE Transactions on Multimedia* 25 (2023), 6808–6820. DOI: https://doi.org/10.1109/TMM.2022.3214775

[28] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiow Keng Tan, and Thomas Wiegand. 2012. Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC). *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (2012), 1669–1684. DOI: https://doi.org/10.1109/TCSVT.2012.2221192

[29] Zefan Qu, Xinyang Jiang, Yifan Yang, Dongsheng Li, and Cairong Zhao. 2024. Online video quality enhancement with spatial-temporal look-up tables. In *European Conference on Computer Vision*. Springer, 449–465. DOI: https://doi.org/10.1007/978-3-031-73220-1_26

[30] Claudio Rota, Marco Buzzelli, Simone Bianco, and Raimondo Schettini. 2023. Video restoration based on deep learning: A comprehensive survey. *Artificial Intelligence Review* 56, 6 (2023), 5317–5364. DOI: https://doi.org/10.1007/s10462-022-10302-5

[31] Ionut Schiopu and Adrian Munteanu. 2022. Deep learning post-filtering using multi-head attention and multiresolution feature fusion for image and intra-video quality enhancement. *Sensors* 22, 4 (2022), 1353. DOI: https://doi.org/10.3390/s22041353

[32] David W. Scott. 2015. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons. DOI: https://doi.org/10.1002/9780470316849

[33] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas 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. DOI: https://doi.org/10.1109/TCSVT.2012.2221191

[34] Weiheng Sun, Xiaohai He, Honggang Chen, Ray E. Sheriff, and Shuhua Xiong. 2020. A quality enhancement framework with noise distribution characteristics for high efficiency video coding. *Neurocomputing* 411 (2020), 428–441. DOI: https://doi.org/10.1016/j.neucom.2020.06.048

[35] Weiheng Sun, Xiaohai He, Chao Ren, Shuhua Xiong, and Honggang Chen. 2022. A quality enhancement network with coding priors for constant bit rate video coding. *Knowledge-Based Systems* 258 (2022), 110010. DOI: https://doi.org/10.1016/j.knosys.2022.110010

[36] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.

[37] Thiow Keng Tan, Rajitha Weerakkody, Marta Mrak, Naeem Ramzan, Vittorio Baroncini, Jens-Rainer Ohm, and Gary J. Sullivan. 2016. Video quality evaluation methodology and verification testing of HEVC compression performance. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 1 (2016), 76–90. DOI: https://doi.org/10.1109/TCSVT.2015.2477916

[38] Gregory Vaksman, Michael Elad, and Peyman Milanfar. 2021. Patch craft: Video denoising by deep modeling and patch matching. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2137–2146. DOI: https://doi.org/10.1109/ICCV48922.2021.00216

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł. Ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 6000–6010.

[40] Tingting Wang, Mingjin Chen, and Hongyang Chao. 2017. A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC. In *2017 Data Compression Conference (DCC)*, 410–419. DOI: https://doi.org/10.1109/DCC.2017.42

[41] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*13, 4 (2004), 600–612. DOI: https://doi.org/10.1109/TIP.2003.819861

[42] Ren Yang, Mai Xu, Tie Liu, Zulin Wang, and Zhenyu Guan. 2019. Enhancing quality for HEVC compressed videos. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 7 (2019), 2039–2054. DOI: https://doi.org/10.1109/TCSVT.2018.2867568

[43] Ren Yang, Mai Xu, and Zulin Wang. 2017. Decoder-side HEVC quality enhancement with scalable convolutional neural network. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 817–822. DOI: https://doi.org/10.1109/ICME.2017.8019299

[44] R. Yang, M. Xu, Z. Wang, and T. Li. 2018. Multi-frame quality enhancement for compressed video. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6664–6673. DOI: https://doi.org/10.1109/CVPR.2018.00697

[45] Qianyu Zhang, Bolun Zheng, Xingying Chen, Quan Chen, Zunjie Zhu, Canjin Wang, Zongpeng Li, Xu Jia, and Chengang Yan. 2024. Hierarchical frequency-based upsampling and refining for HEVC compressed video enhancement. *IEEE Transactions on Circuits and Systems for Video Technology* (2024), 1–1. DOI: https://doi.org/10.1109/TCSVT.2024.3517840

[46] Xiaoqin Zhang, Runhua Jiang, Tao Wang, Pengcheng Huang, and Li Zhao. 2021. Attention-based interpolation network for video deblurring. *Neurocomputing* 453 (2021), 865–875. DOI: https://doi.org/10.1016/j.neucom.2020.04.147

[47] Minyi Zhao, Yi Xu, and Shuigeng Zhou. 2021. Recursive fusion and deformable spatiotemporal attention for video compression artifact reduction. In *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, New York, NY, 5646–5654. DOI: https://doi.org/10.1145/3474085.3475710

[48] Simone Zini and Marco Buzzelli. 2022. Laplacian encoder-decoder network for raindrop removal. *Pattern Recognition Letters* 158 (2022), 24–33. DOI: https://doi.org/10.1016/j.patrec.2022.04.016

Received 21 June 2024; revised 19 March 2025; accepted 21 March 2025