Journal Pre-proof

CVEGAN: A perceptually-inspired GAN for Compressed Video Enhancement

Di Ma, Fan Zhang, David R. Bull

PII: S0923-5965(24)00028-6
DOI: https://doi.org/10.1016/j.image.2024.117127
Reference: IMAGE 117127

To appear in: *Signal Processing: Image Communication*

Received date : 24 June 2022
Revised date : 15 December 2023
Accepted date : 30 March 2024

Please cite this article as: D. Ma, F. Zhang and D.R. Bull, CVEGAN: A perceptually-inspired GAN for Compressed Video Enhancement, *Signal Processing: Image Communication* (2024), doi: https://doi.org/10.1016/j.image.2024.117127.

CVEGAN: A Perceptually-inspired GAN for Compressed Video Enhancement

Di Ma, Fan Zhang and David R. Bull

Department of Electrical and Electronic Engineering, University of Bristol, Bristol, BS8 1UB, UK

Abstract

We propose a new Generative Adversarial Network for Compressed Video frame quality Enhancement (CVEGAN) . The CVEGAN generator benefits from the use of a novel Mul^{2}Res block (with multiple levels of residual learning branches), an enhanced residual non-local block (ERNB) and an enhanced convolutional block attention module (ECBAM). The ERNB has also been employed in the discriminator to improve the representational capability. The training strategy has also been re-designed specifically for video compression applications, to employ a relativistic sphere GAN (ReSphereGAN) training methodology together with new perceptual loss functions. The proposed network has been fully evaluated in the context of two typical video compression enhancement tools: post-processing (PP) and spatial resolution adaptation (SRA). CVEGAN has been fully integrated into the MPEG HEVC and VVC video coding test models (HM 16.20 and VTM 7.0) and experimental results demonstrate significant coding gains (up to 28% for PP and 38% for SRA compared to the anchor) over existing state-of-the-art architectures for both coding tools across multiple datasets based on the HM 16.20. The respective gains for VTM 7.0 are up to 8.0% for PP and up to 20.3% for SRA.

Keywords: CVEGAN, GAN, perceptual loss, ReSphereGAN, Mul^{2}Res, ERNB, ECBAM, HEVC, VVC.

1. Introduction

Over the past decade, machine learning methods based on deep neural networks have provided revolutionary advances across various computer vision applications, in particular for image/video processing and understanding. More recently these have been successfully applied to picture compression, both in terms of enhancing individual tools within conventional codecs, and also in providing new end-to-end compression via auto-encoder architectures [1].

In the context of learning-based video coding, two major classes exist - CNN-enhanced coding tools for hybrid conventional codecs [2–8] and end-to-end optimised deep video frameworks [9–14]. Although the latter offers the potential to achieve competitive coding performance, they still cannot outperform the latest standard video coding methods, such as VVC and AV1. Among CNN-enhanced coding tools, there is a distinct class of methods which provide superior performance by employing Convolutional Neural Network (CNN) processes to enhance the quality of the video reconstructed at the decoder. Typical examples include post-processing (PP) and video format adaptations (with CNN format restoration) [15, 16]. These approaches often employ relatively simple network structures and utilise pixel-wise loss functions, which cannot produce the reconstruction results with optimal visual quality. More recently, compression enhancement tools have been improved by employing Generative Adversarial Networks (GANs) [17–19], in which the adversarial training methodologies have been used to jointly train two networks (the generator and discriminator) in order to achieve better perceptual reconstruction quality. After training, only the generator is integrated into those CNN-based coding tools (e.g., PP) to enhance visual quality. However, it is noted that, in these works, the employed loss functions have typically combined pixel-wise distortions, simple quality metrics and feature map differences with artificially configured weights, which do not offer optimal correlation with visual quality.

In this context, we propose a novel GAN architecture for Compressed Video Enhancement (CVEGAN). The main contributions of this work are summarised below.

(1) (**Mul^{2}Res blocks**) We present a novel block structure, Mul^{2}Res, which employs multiple levels of multiple residual learning branches with various kernel sizes. As far as we know, this is the first use of a nested residual learning structure with various kernel sizes; it improves overall enhancement performance compared to conventional residual learning blocks with a fixed kernel size.

(2) (**ERNB and ECBAM**) We employ enhanced residual non-local blocks (ERNBs) and enhanced convolutional block attention modules (ECBAMs), which have been modified from non-local blocks [20] and convolutional block attention modules [21] respectively. Through extracting non-local features (ERNB) and applying channel and spatial attention mechanisms (ECBAM), the representational capability of the network has been further improved.

(3) (**ReSphereGAN training strategy**) We have designed a new training methodology, Relativistic SphereGAN (ReSphereGAN), which embodies an enhanced version of the original SphereGAN [22]. Compared to conventional GAN algorithms, this new adversarial training strategy exploits higher-order statistics of feature points in the hypersphere to achieve better training performance.

(4) (**Perceptual loss**) We also propose a novel perceptual loss function which optimises video quality during training. This linearly combines the logarithms of four existing losses, where the combining weights have been determined using data from eight subjective video quality databases, hence making the proposed loss function correlate better with perceptual quality.

The rest of the paper is organised as follows. Section 2 overviews the recent advances in video coding standards and the state of the art in deep video compression. Section 3 describes the the proposed network architectures in detail, while the new training methodologies (including the perceptual loss function) are presented in Section 4. In Section 5, experimental configurations are summarised which have been used to evaluate the proposed algorithms. Section 6 reports experimental results with analysis and discussion. Finally, conclusions and future work are outlined in Section 7.

2. Related Work

With dramatic increases in numbers of users and available content, the introduction of new immersive formats and higher user quality expectations, video content is now by far the greatest consumer of global internet bandwidth. These increased demands are challenging all delivery technologies, not least video compression which plays a key role in the trade-off between limited network capacity and elevated user video quality expectations.

2.1. Standard Video Codecs

The latest video coding standard, Versatile Video Coding (VVC) [23], approved by ISO and ITU in 2020, provides 30-40% bitrate savings [24] over its predecessor, High Efficiency Video Coding (HEVC) [25]. A competitor of HEVC and VVC is the open source and royalty-free codec, AV1, which was released by the Alliance for Open Media (AOM) in 2018. AV1 also achieves significant coding gains compared to HEVC [26]. Despite their evident coding gains, none of these codecs have widely exploited machine learning methods to optimise their architecture or tools. In contrast several learning-based approaches have been published in the open literature which achieve very promising results compared to these standards.

2.2. Deep Learning-based Video Compression

Existing learning-based coding algorithms can be classified into two primary categories. The first relates to end-to-end training and optimisation using autoencoder type architectures for image [9–11, 27, 28] and video [12–14, 29–31] compression tasks . Although solutions in this category are not yet competitive with the latest standardised codecs, such as VVC and AV1, they demonstrate significant potential for the future.

A second class contains algorithms (primarily based on CNNs) that are designed to enhance individual coding tools within a standard codec configuration. Such approaches have been used to optimise tools including: intra prediction [2, 32], motion estimation [3, 33], transforms [4, 34], quantisation [5], entropy coding [6, 35], post-processing (PP) [36, 37], in-loop filtering (ILF) [7, 38] and format adaptation [8, 39, 40].

Among these CNN-based coding tools, there is one group of methods, which stand out in offering higher coding gains compared to the others [15, 16]. These perform CNN operations at the decoder of a conventional codec (e.g., HEVC HM or VVC VTM) to enhance the quality of reconstructed video frames. Typical examples include post-processing (PP) [37, 41], in-loop filtering [38, 42], spatial resolution adaptation (SRA) and bit depth adaptation [43]. While most PP approaches focus on single frame enhancement, some recent contributions have reported multi-frame enhancement methods [17, 44]. However, because these are associated with much higher computational complexity, we address only single frame enhancement here. Exploring the multi-frame enhancement method to handle and evaluate inter-frame consistency for compressed video enhancement will be our future work.

2.3. Network Architecture

Most CNN architectures reported for video enhancement have their origins in single image restoration. Their common architectural features include: (i) concatenated convolutional layers [45–52] (ii) concatenated residual blocks [53–57]; (iii) residual dense connections [8, 58–62]; (iv) cascading connections [7, 63–65]; and (v) feature review structures [7, 57, 59, 62]. More recently, new advanced structures have also been proposed including channel and spatial attention mechanisms [21, 65–67] and non-local feature extraction [20, 68, 69].

The performance of a CNN is generally related to three primary factors: *depth* (i.e. the depth of networks), *width* (i.e. the number of feature maps), and *cardinality* (which is defined in [70–72] as the size of transformation sets, i.e. the number of convolutional branches used in a convolutional block in CNN models) . Most of the network architectures described above have been designed to increase the network *depth* [20, 47, 53, 58, 59, 66, 73, 74] and *width* [55, 58, 59, 71, 74, 75], while only a few of them exploit the *cardinality* characteristic [72, 76–78]. However, *cardinality* is widely acknowledged to be a more effective way to improve overall performance and network capacity compared to the other two factors [72]. We also note that the kernel size of convolutional layers in these existing networks is usually set at a fixed value (3 in most cases), which may limit the receptive field size and hence the overall network performance [70].

2.4. Training Strategy and Loss Functions

In most deep learning-based coding enhancement tools, $\ell 1$ or $\ell 2$ loss is used to train the CNN models with the aim of minimising pixel-wise distortions. Alternative training strategies have been proposed to improve perceptual image quality, typically based on Generative Adversarial Network (GAN) architectures and loss functions combining $\ell 1/\ell 2$ loss, feature map differences (e.g. VGG19-54 [58]) and low-complexity quality metrics (e.g. MS-SSIM and SSIM). Notable examples include approaches using standard GANs [53, 79], Relativistic average GANs (RaGAN) [58, 74], conditional GANs (cGAN) [75], Patch GANs [80], and Wasserstein GAN-gradient penalty (WGAN-GP) [81]. It is noted that the cGAN model has been recently used for end-to-end image compression [27]. The generator of cGAN is firstly used to reconstruct the image using the latent representation generated by the encoder and the discriminator accepts the original and the reconstructed images as its input to distinguish which one is real or fake. The rate-distortion and cGAN adversarial losses have been combined to conduct end-to-end training and optimisation achieving good perceptual quality on the compressed images with relatively low bitrates compared to other methods.

Moreover, it is well known that $\ell 1$ and $\ell 2$ losses do not correlate well with subjective video quality [82, 83], and the combined loss functions employed in these GAN-based training strategies use artificially configured combining weights, which have never been fully evaluated in terms of their correlation with subjective video quality. These issues inevitably lead to sub-optimal training performance when the networks are utilised for compression application.

3. The Proposed Network Architectures

The proposed CVEGAN follows the basic GAN framework [84], combining a generator and a discriminator. Its architecture is described in the following subsections.

3.1. Generator Architecture

The generator, denoted as CVENet, is shown in Figure 1. This takes a 96×96 YCbCr 4:4:4 compressed image block as the input and outputs a processed image block in the same format, targeting its original uncompressed version. The kernel sizes, number of feature maps and stride values for each convolutional layer are shown in Figure 1. It is noted that the existing CNN models (single frame-based) designed for image and video restoration were mainly developed following a same backbone structure as proposed in [45]. Our core network architecture is also similar to this backbone structure. It comprises three primary stages: shallow feature extraction-Stage 1 (S1), deep feature processing-Stage 2 (S2), and final reconstruction-Stage 3 (S3) as shown in Figure 1. Different from the existing network architectures, we carefully redesigned these three stages to effectively improve the overall performance of the network for video compression enhancement purposes.

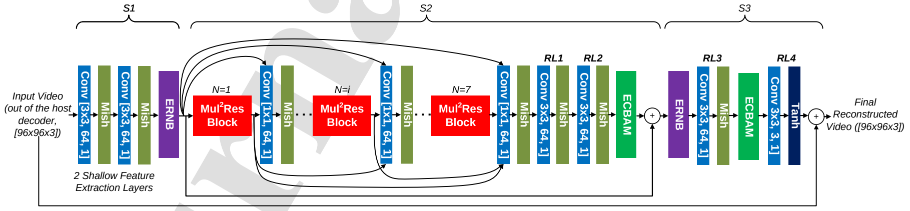


Figure 1: Illustration of the CVEGAN Generator (CVENet).

CVENet has three unique structural features: (1) Mul^{2}Res blocks (employed in S2); (2) enhanced residual non-local blocks (ERNB) used in S1 and S3; (3) enhanced convolutional block attention modules (ECBAM) utilised in Mul^{2}Res blocks and S3. These are described below. It can be seen that a Mish activation

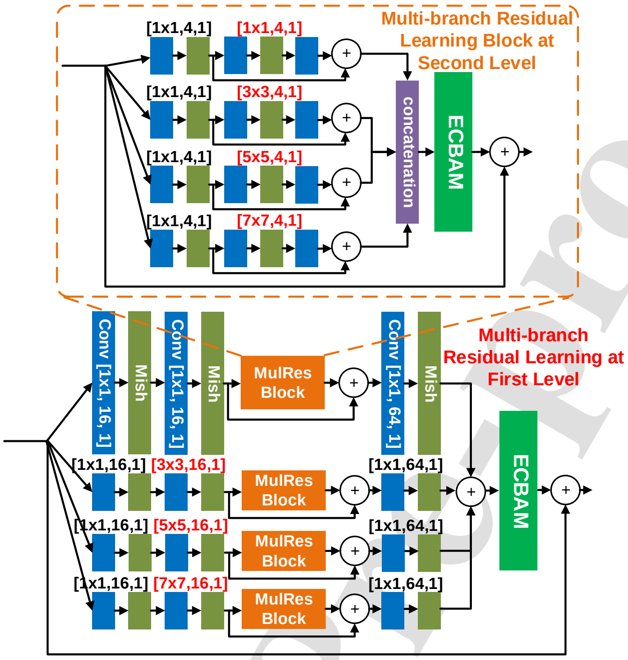


Figure 2: Illustration of an Mul^{2}Res Block.

function [85] has been employed after all convolutional operations except for the final one. Mish has been previously reported to offer better performance than other commonly used functions such as ReLU, leaky ReLU (LReLU) and parametric ReLU (PReLU) [85].

In order to further improve information flow between Mul^{2}Res blocks, cascading connections [63] (shown as black curves in Figure 1) are utilised between the input of the first Mul^{2}Res block and the 1×1 convolutional layer after each of the Mul^{2}Res blocks, and between the output of each Mul^{2}Res block (except the final one) and the 1×1 convolutional layer after each of the subsequent Mul^{2}Res blocks.

Mul^{2}Res Block: The new Mul^{2}Res Block structure contains multiple levels of multiple residual learning branches (novel nested residual learning structures) to exploit the *cardinality* characteristic of networks. Figure 2 illustrates the Mul^{2}Res Block structure used in CVEGAN, which has four residual learning branches at two different levels. The number of branches and levels can be adapted for different applications based on the computational resources available. At the first level, the input of the Mul^{2}Res block is fed into four residual learning branches. Each branch has a convolutional layer with various kernel sizes (1×1, 3×3, 5×5 and 7×7). This diversifies the feature maps extracted by the convolutional layers using various receptive field sizes. The ECBAM is also employed at both levels before the final skip connection. The Mul^{2}Res block at the second level also has four residual learning branches. The primary differences include (i) the MulRes Blocks at the first level are replaced by residual blocks, each of which contains two convolutional layers (with various kernel sizes for different branches) and a Mish activation function; (ii) the output from four branches are concatenated before feeding into the ECBAM.

The proposed structure with multiple levels of multiple branches (the novel nested residual learning structures) has been designed to increase the network's *cardinality*. This supports different kernel sizes which enable operations with various receptive fields and is expected to offer improved performance for content with complex textures [72, 86]. Moreover, ECBAM has been utilised in the Mul^{2}Res blocks to extract informative features from the source data in order to further improve the reconstruction ability of the network [66]. As shown in Figure 2, we employed convolutional layers with kernel size of 1×1 (alongside element-wise addition) and the concatenation operation at the first and second levels of the Mul^{2}Res block (as shown in Figure 2) respectively to keep the number of output feature maps consistent. The spatial size of feature maps output from each branch is not changed due to the use of a fixed stride value of 1 in the model.

ERNB: Non-local operations have been initially designed to capture longer-range dependencies between pixels within the input image block or feature maps. It can effectively improve the representational ability of the networks. However, the non-local feature fusion stage in the original algorithm was implemented using an embedded Gaussian function incorporated with a large matrix multiplication which is not a trainable process [20, 68]. This inevitably leads to the higher computational complexity, especially when the large feature maps are processed. In this context, we have developed an enhanced residual non-local block (ERNB) based on the original non-local operations proposed in [20, 68], as show in Figure 3. This employs concatenation operations and residual blocks to achieve feature fusion, avoiding large matrix multiplication, facilitating training of the feature fusion process and improving information flow of the network. A further modification is employment of a long skip connection to produce the ERNB output, which can further stabilise the non-local learning.

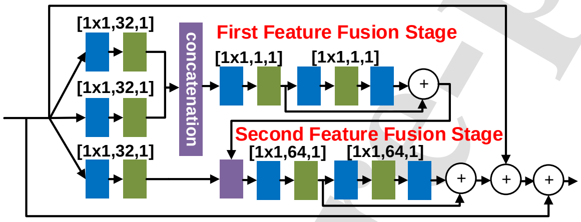


Figure 3: Illustration of an ERNB.

ECBAM: Inspired by attention mechanisms proposed recently [20, 21, 66], we have designed an enhanced convolutional block attention module (ECBAM) for CVEGAN (Figure 4). This follows the basic structure of the convolutional block attention module (CBAM) [21], which comprises a channel and a spatial attention module. Rather than directly producing output from the second matrix multiplication as in [21], we have added a concatenation operation with a convolutional layer to achieve non-linear feature fusion. This has been previously reported to improve information flow and overall performance of the network [59].

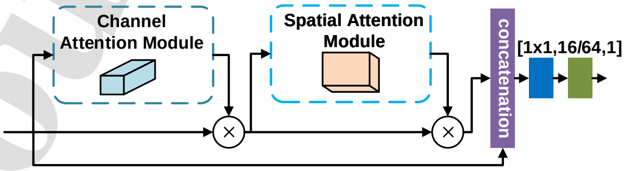


Figure 4: Illustration of an ECBAM. ⊗: matrix multiplication.

3.2. Discriminator Architecture

Figure 5 illustrates the architecture of the CVEGAN discriminator, which is based on SRGAN [53]. This network was only used in the training process, which helps to improve the overall performance of the final model. The primary differences include (i) we employ two ERNBs - one after the first convolutional layer, and the other before the final convolutional layer - these extract non-local features and improve the representational capability. (ii) we remove the final dense layer in SRGAN to output high (1024) dimensional feature points rather than a single 1D scalar.

4. Training Methodology

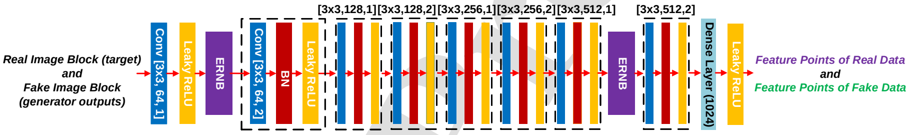


Figure 5: Illustration of the CVEGAN's Discriminator.

We follow a similar training strategy to that described in [53, 58], which consists of two stages: (i) CVENet is trained using a combined perceptual loss function (Section 4.1) to obtain an initial model for the second training step; (ii) CVENet is then trained jointly with the discriminator with a new training method ReSphereGAN (Section 4.2).

4.1. Perceptually-inspired Loss Function

As discussed in Section 2.4, existing loss functions combining pixel-wise losses, feature map differences and low-complexity quality metrics, do not always correlate well with perceived quality. To address this we have employed a loss function comprising a linear combination of the elementary transforms of six commonly used losses, $\ell1$ (denoted as L_1), $\ell2$ (L_2), gradient loss[1] [75, 87] (L_3), VGG19-54 loss [58] (L_4)[2], Structural Similarity Index (SSIM) loss [88] (L_5) and Multi-scale SSIM (MS-SSIM) loss [89] (L_6):

$$L_{\text{test}} = \sum_{i=1}^{6} a_i f(L_i). \tag{1}$$

Here, $f(\cdot)$ represents an elementary transformation [90], which can be either a constant, a power, a root, an exponential, a logarithmic, a trigonometric, an inverse trigonometric, a hyperbolic or an inverse hyperbolic function[3]. a_i represents the linear combination weights, where a_6 is always set to 1 in order to relatively simplify the parameter exhaustive search process. The range of all these six single losses is between 0 and 1. We have excluded non-linear combinations and combinations of different transformed losses due to their high computational and training complexity.

We have used an eight-fold cross validation method [91] to train the proposed combined loss function (equation (1)) based on eight publicly available subjective video quality databases. These include: the Netflix public database (70 test sequences) [92], BVI-HD (192) [82], CC-HD (108) [26], CC-HDDO (90) [93], MCL-V (96) [94], SHVC (32) [95], IVP (100) [96], and VQEG-HD3 (72) [97]. All of these contain video sequences compressed using commonly used video codecs (H.264, HEVC, AV1, VVC or MPEG-2).

The eight databases were divided into two sub groups - seven training datasets and one for testing. An exhaustive search was performed among all tested transformation functions and their corresponding weighting parameters,

[1] This calculates the mean of absolute differences between the horizontal and vertical gradients of the original and restored images. It is used to minimise the gradient discrepancy to reconstruct the high frequency details in the restored images.

[2] Here, the original and the restored (output from the neural network) images are input to a pre-trained 19-layer VGG network to produce high level features from the 4th convolutional layer (just before the 5th max-pooling layer). The root-mean-square-error between these two are then obtained as the final loss for training.

[3] It is noted that the elementary transformations are used here as they are differentiable [90] and easily to be calculated.

the best of which was selected for this split to achieve the highest average correlation between the combined loss values and subjective scores for all seven training datasets. The Spearman rank order correlation coefficient (SROCC) [1] is employed to quantify the correlation performance. The search range of each parameter is between 0 and 1, with an interval of 0.1.

Table 1: Cross-validation results over eight training-testing trails.

Loss Function	L_1	L_2	L_3	L_4	L_5	L_6
Average SROCC	0.5984	0.6478	0.3991	0.7085	0.5720	0.7168
Loss Function	L_7	L_8	L_9	L_{10}	L_{11}	Ours
Average SROCC	0.5430	0.5993	0.6008	0.6198	0.6591	**0.8067**

To avoid possible content bias due to a single training-testing split, we performed this cross-validation for all eight splits. Table 1 presents the average SROCC performance on the test datasets among all eight splits for the trained loss functions, which are compared to the results from 11 commonly used loss functions in training image restoration and enhancement CNNs, L_1-L_6; linear combination of $\ell 1$ loss, gradient loss and VGG19-54 perceptual loss [98] (L_7); linear combination of $\ell 1$ loss and SSIM loss [73] (L_8); linear combination of $\ell 1$ loss and VGG19-54 perceptual loss [58, 64, 74, 80, 99] (L_9); linear combination of $\ell 1$ loss, gradient loss, SSIM and MS-SSIM losses and VGG19-54 perceptual loss [75] (L_{10}); linear combination of MSE and VGG19-54 perceptual loss [53] (L_{11}). It can be observed that our trained loss functions have an average SROCC value of 0.8067, which is significantly higher than those for other tested loss functions (L_1-L_{11}).

The transformation function used for all optimal loss functions across all eight training-testing splits, is the natural logarithm $ln(\cdot)$. We use the median values of the corresponding combination parameters, and normalise these to ensure $\sum_{i=1}^{6} a_i = 1$. The final combined loss function $\mathcal{L}_P$ used to train CVEGAN is given below:

$$\begin{aligned}\mathcal{L}_P = \;& 0.3 \cdot ln(\ell 1) + 0.2 \cdot ln(\text{SSIM\_loss}) + \\ & 0.1 \cdot ln(\ell 2) + 0.4 \cdot ln(\text{MSSSIM\_loss}) + \\ & 0 \cdot ln(\text{VGG19-54\_loss}) + 0 \cdot ln(\text{Gradient\_loss})\end{aligned} \tag{2}$$

It is noted that the equation (1) is a generic format of the loss function. After conducting training process as discussed above, all combining weights and elementary transformations are determined, and the final perceptual loss function is shown as equation (2). There are six terms in equation (1), but for two of them (VGG19-54 loss and gradient loss), the optimal values are zero based on the cross validation results. It should be noted that $\mathcal{L}_P$ remains within the range 0 to 1, and is differentiable, which is required to support back-propagation during training. Since the test sequences were generated using a wide range of video codecs, this loss function should generalise well across image and video compression applications. $\mathcal{L}_P$ has been used here for training the CVENet (generator) during the first training stage.

4.2. ReSphereGAN

The proposed Relativistic SphereGAN training methodology is a modified version of the SphereGAN [22], which is based on an *integral probability metric* (IPM) and has achieved superior performance compared to other commonly used GAN algorithms [22, 100]. As illustrated in Figure 6, the original SphereGAN compares the geodesic distances between the north pole **N** and fake/real feature points in the n-dimensional Euclidean feature space through an inverse stereographic projection. Here n is the feature point number of fake and real data produced by the CVEGAN discriminator, which has a default value of 1024. It is noted that the main benefits of converting the plane to the hypersphere include [22, 100]: (1) exploiting higher-order statistics of feature points based on the geodesic distances in the hypersphere, hence providing better training performance; and (2) the geodesic distance function in the hypersphere is bounded which makes GAN adversarial training more stable. Inspired by RaGAN [101], we also calculate the relativistic geodesic distance between the projected real and fake feature points in our loss functions at the second training stage. This modification further optimises the generator by obtaining gradient information from both real and fake data during the adversarial training process. Specifically, the loss functions for generator ($\mathcal{L}_{Re\_gen}$) and discriminator ($\mathcal{L}_{Re\_disc}$) (in the second training stage) are given below:

$$\begin{aligned}\mathcal{L}_{Re\_gen} = \quad & \mathcal{L}_P + 0.005 \cdot (-\textstyle\sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_f}))) \\ & + \textstyle\sum_{m=1}^{M} E(d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))))\end{aligned} \tag{3}$$

$$\begin{aligned}&\mathcal{L}_{Re\_disc} = \textstyle\sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_f}))) - \\ &\textstyle\sum_{m=1}^{M} E(d^m(\mathbf{N}, T(\mathbf{x_r}))) - \sum_{m=1}^{M} E(d^m(T(\mathbf{x_r}), T(\mathbf{x_f})))\end{aligned} \tag{4}$$

Here, $E(\cdot)$ represents the mean operation. $d^m(\mathbf{a}, \mathbf{b})$ is the *geometric-aware transformation function* [22], which calculates the *m-th* central geometric moment (geodesic distance) [22] in the hypersphere space between $\mathbf{a}$ and $\mathbf{b}$ (projected feature points in the hypersphere space). M is set to 3 in this work. $T(\cdot)$ stands for the *inverse of stereographic projection* [22] from the Euclidean feature space to the hypersphere space.

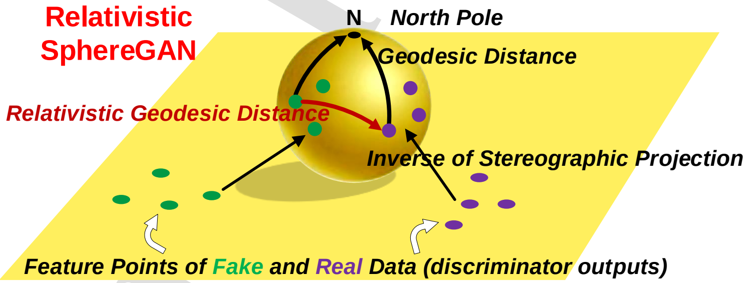

Relativistic
SphereGAN
N
North Pole
Geodesic Distance
Relativistic Geodesic Distance
Inverse of Stereographic Projection
Feature Points of Fake and Real Data (discriminator outputs)

Figure 6: Illustration of the proposed ReSphereGAN. Yellow plane and sphere represent the 1024-dimensional Euclidean feature space and hypersphere respectively. Green and purple points represent the feature points of fake and real data respectively.

$\mathbf{N}$ is denoted as the north pole in the hypersphere space, and $\mathbf{x_r}$ and $\mathbf{x_f}$ are the real and fake feature points respectively in n-dimensional Euclidean feature space. The weight combining the perceptual loss ($\mathcal{L}_P$) and the generator adversarial loss was set to 0.005 based on several previous works [58, 73, 74] on GAN-based image restoration and enhancement.

Gradient Analysis

The gradients generated from the losses during the second training stage are crucial for stabilising the GAN models. It is important to avoid vanishing gradients and explosion problems as pointed out in [70, 84]. The gradients of $d^m(\mathbf{N}, T(\mathbf{x_r}))$ and $d^m(\mathbf{N}, T(\mathbf{x_f}))$ have already been evaluated in [22], so we just analyse the gradients of the new relativistic geodesic distance $d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))$.

Lemma 1. $E(\left\|\nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))\right\|_2) < \infty$ for all m. Here ∇ represents the derivation operation, and $\|\cdot\|_2$ is the Euclidean norm.

The proof of **Lemma 1** is provided in Appendix A. **Lemma 1** indicates that the ReSphereGAN can ensure stable GAN learning with any moment m. In practice, we have noted that although ReSphereGAN may generate relatively large gradients as the original SphereGAN, they can still be calculated during training process when the Adam optimiser is used [102]. This has also been reported in [22].

5. Experiment Configuration

As mentioned in Section 2.2, we have trained and evaluated the proposed network for two coding tools: post-processing (PP) and spatial resolution adaptation (SRA). Other important tools for compression enhancement, e.g. effective bit depth adaptation, has similar workflow to PP and SRA but offers lower overall coding gains [43, 103]. These are therefore not presented here due to the limited space available.

5.1. CNN-based PP and SRA Coding Tools

The coding workflows for CNN-based PP and SRA tools are shown in Figures 7 and 8. The CNN module employed for PP provides an additional quality enhancement on the decoded content, while for SRA, the original video is spatially down-sampled (by a factor of 2) before encoding and a CNN-based super resolution operation is performed on the decoded low-resolution content for both up-sampling and quality enhancement. More detailed descriptions of PP and SRA can be found in [103].

As indicated by the yellow boxes shown in Figures 7 and 8, the generator of the proposed CVEGAN network is placed after the host decoder to directly accept the decoded frames as its input and produce reconstructed frames with higher perceptual quality. In contrast to the method in [27], the proposed GAN model is not involved in any encoding or decoding operations (e.g., to decode any latent representations or perform any operations in the latent space) - all the encoding and decoding operations are conducted in the original host encoder and decoder respectively. It is also noted that CVEGAN's discriminator is only used during the training stage to improve the performance of the generator (CVENet as shown in Figure 1). At the inference stage, only the generator - CVENet is directly invoked to enhance the decoded frames.

5.2. Training Data and Configuration

A large training database, BVI-DVC [103], has been employed to train the proposed network. This contains 800 video clips (10 bit, YCbCr 4:2:0) at various spatial resolutions from 270p to 2160p covering a wide range of content and video texture types. It has been previously reported to provide enhanced training performance over other databases for optimising CNN-based coding enhancement tools. It has recently been adopted as the default training database by JVET Ad-hoc Group 11 (Neural-network-based video coding) [104].

We followed the same procedure as detailed in [103] to generate training material for CNN-based PP and SRA coding tools using the HEVC HM 16.20 and VVC VTM 7.0 with the Random Access (RA) configurations (Main10 profile) and four base quantisation parameter (QP) values, 22, 27, 32 and 37. This produces two classes of training data for PP and SRA, each of which contains four QP sub-groups. For the SRA class, the nearest neighbour (NN) filter was utilised to up-sample the compressed low resolution video frames to the original resolution, which ensures that the same CVENet architecture can be used for both PP and SRA.

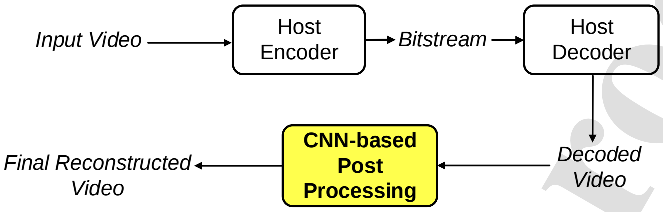


Figure 7: Coding workflow with a CNN-based PP module.

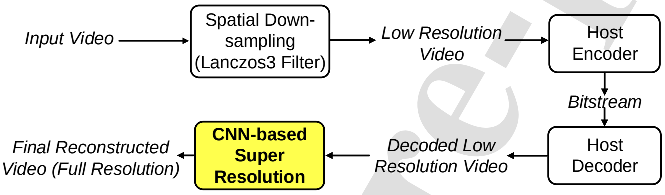

Input Video
Spatial Down-sampling (Lanczos3 Filter)
Low Resolution Video
Host Encoder
Bitstream
Host Decoder
Decoded Low Resolution Video
CNN-based Super Resolution
Final Reconstructed Video (Full Resolution)

Figure 8: Coding workflow with a CNN-based SRA module.

365 For each QP sub-group in the PP and SRA classes, the compressed (or the NN up-sampled compressed) video frames and their corresponding original counterparts were randomly selected, cropped into 96×96 blocks and converted to YCbCr 4:4:4 format. Finally, approx. 195,000 pairs of image blocks for each QP sub-group were generated.

370 CVEGAN was implemented using the TensorFlow platform (1.8.0). The training process was conducted based on the following parameters: Adam optimisation [102] with the following hyper-parameters: β_1=0.9 and β_2=0.999; batch size of 16; 200 training epochs; initial learning rate (0.0001); weight decay of 0.1 for every 100 epochs.

375 *5.3. Evaluation Datasets and Configuration*

During evaluation, each decoded frame (after NN up-sampling for SRA) was converted to YCbCr 4:4:4 format and segmented into 96×96 overlapping blocks with an overlap size of 4 pixels as CVENet (the generator) input. The network output blocks from CVENet were further aggregated following the same pattern and converted to YCbCr 4:2:0 format to form the final reconstructed frame.

Twenty-four popular and state-of-the-art CNN and GAN architectures, which have been widely used in image super-resolution, restoration and video compression, have been benchmarked in this paper. All of these have been re-implemented and trained using the same framework (TensorFlow 1.8.0) with identical training material following the same training methodology and loss functions as described in their original literature. During re-implementation, the input and output interfaces of these networks have been modified to satisfy the data format requirements.

All networks under test have been integrated into both PP and SRA coding tools and tested under JVET common test conditions (CTC) [105] using the RA configuration (Main10) with four QP values of 22, 27, 32 and 37. The original HEVC HM 16.20 and VVC VTM 7.0 were used as the host codecs and also as the benchmark anchors. The coding performance against HEVC HM and VVC VTM is calculated using the Bjøntegaard Delta [106] measurement (BD-rate) based on two quality metrics: Peak Signal-to-Noise-Ratio (PSNR, luma channel only) and Video Multimethod Assessment Fusion (VMAF, 0.6.1) [107]. PSNR is the most widely used quality metric for image and video compression, although it does not always correlate well with subjective quality scores [82, 108]. VMAF is a machine learning-based video quality assessment algorithm, increasingly used in industry, which combines various quality metrics and video features using a Support Vector Machine regressor. It has been reported to offer much better correlation performance with subjective opinions on compressed content compared to the PSNR [82, 107].

The relative computational complexity of all test networks (the generator for the case of GANs) have also been calculated and benchmarked against the simplest SRCNN [45]. The training and evaluation processes were both executed on a shared cluster, where each node contains two 14 core 2.4 GHz Intel E5-2680 V4 (Broadwell) CPUs, 128 GB of RAM, and NVIDIA P100 GPU devices.

To evaluate their performance, JVET-CTC SDR (19 test sequences) [105] has been employed as the main test dataset due to its content and resolution diversity. The average BD-rates assessed by both PSNR and VMAF were calculated for all tested networks. To further evaluate network generalisation, another two commonly used test databases, UVG (6 test sequences) [109] and AOM main test dataset (21 test sequences) *objective-1-fast* (o-1-f) [4] [110], have also been employed to test the proposed CVEGAN and the other three top performers. None of the sequences in these three test datasets were included in the training database, BVI-DVC. It should be noted that only UHD (2160p) content from these databases was used to evaluate the SRA coding tool since, as previously reported [43], lower resolutions provide only limited and inconsistent coding gains.

We have also conducted a lab-based subjective test using a double stimulus methodology on a selection of network architectures (alongside the anchor HM and CVEGAN) for both PP and SRA. We collected subjective scores from twenty-eight subjects [111] using the reconstructed videos of 12 UHD source sequences (QP 37 only). Further details of the testing configuration can be found in the Section 5.5.

5.4. Ablation Study

Five primary contributions have been tested and compared to the state of the art for PP and SRA. All ablation studies are based on the JVET-CTC SDR dataset. It is noted that the discriminator loss has been employed for each comparison in ablation study.

(1) **Mul^{2}Res Block** effectiveness has been evaluated by replacing it with other commonly used convolutional blocks for CNN-based image restoration, which include residual block (RB) [53], modified residual block (MRB) [73], residual dense block (RDB) [59], residual-in-residual dense block (RRDB) [58], residual channel attention block (RCAB) [66], Xception block [76] (*cardinality*

[4] We have excluded a few source sequences in UVG and o-1-f datasets, which have been employed in BVI-DVC as training data.

is 4), ResNeXt block [72] (*cardinality* is 4) and ResNeSt block [78] (*cardinality* is 4).

(2) **ERNB** is substituted by the original non-local block [20] to evaluate its effectiveness.

(3) **ECBAM** is replaced by the original CBAM [21] for comparison.

(4) **ReSphereGAN** training has been compared with other commonly used GAN training approaches, including standard GAN [53], Relativistic average GAN (RaGAN) [58], PatchGAN [80], conditional GAN (cGAN) [75], Wasserstein GAN-gradient penalty (WGAN-GP) [81] and the original SphereGAN [22].

(5) **Perceptual Loss Function** proposed in this paper was compared with other commonly used loss functions for GAN training (L_7-L_{11}), as described in the Section 4.1.

5.5. Subjective Test Configuration

We have conducted a lab-based subjective test on the results generated by CVEGAN and a selection of other tested network architectures. Twelve UHD (2160p) source sequences from the JVET-CTC SDR [105] and UVG [109] datasets are selected as source content in this subjective evaluation. They have been encoded by the original HEVC HM 16.20 and its two enhanced versions (QP 37 only) with CNN-based PP and SRA coding tools. Each tool further generated results using four different networks to perform CNN operations, including RNAN [20], RFB-ESRGAN [74], MSRGAN [73], and the proposed CVEGAN. The former three architectures were selected due to their relatively higher coding gains (see Table 2) compared to other benchmark networks assessed by VMAF. This results in 9 different test versions for each source sequence.

The subjective tests were conducted in a laboratory with a darkened, living room style environment. The background luminance level was set to 15% of the peak luminance of the monitor used [111]. All the video sequences were shown at their native framerates, on a SONY PVM-X550 4K OLED professional video monitor with a maximum viewing angle of 89° and an effective picture size (H×V) of 1209.6×680.4 mm. The spatial resolution of the monitor was configured as 3840×2160, and it was connected to a Windows PC running the MATLAB R2019b and Psychotoolbox 3.0. The viewing distance was set to be 1.6 times of the monitor height (718.4 mm) based on the ITU-R BT.500 [111].

In this experiment, the double stimulus continuous quality scale (DSCQS) methodology [111] was used. Twenty-eight subjects (16 male and 12 female) [111] participated in this experiment and their average age was 31.6 years. They all had normal or corrected-to-normal colour vision verified by using Snellen and Ishihara charts [111]. After viewing these two sequences, the participants were asked to rate the perceived quality of both videos, based on a continuous quality scale from 1 to 5 (1-Bad, 2-Poor, 3-Fair, 4- Good and 5-Excellent).

After the experiment, a difference score were calculated for each trial and each participant by subtracting the quality score of the distorted sequence from its corresponding reference source. Possible outliers were removed following the procedures described in [111]. A Difference mean opinion score (DMOS) were obtained for every trial by taking the mean of the difference scores. We then calculated the average DMOS among all source sequences for each test version as shown in Table 6 (in the main paper).

6. Results and Discussion

This section presents an analysis of the rate quality performance of CVEGAN and a comparison with twenty-four state-of-the-art network architectures in the context of video compression enhancement. Perceptual comparisons are also given to aid further evaluation of its effectiveness.

6.1. Compression Performance Comparisons

Table 2 summarises the compression performance generated by CVEGAN and the 24 CNN/GAN networks when they are integrated into post-processing (PP) and spatial resolution adaptation (SRA) coding tools in the context of HEVC. We can observe that for both PP and SRA coding tools, although based on PSNR, the pixel-wise distortion based quality metric, the proposed CVEGAN

Table 2: Comprehensive compression results (in terms of BD-rate based on both PSNR and VMAF) of the proposed CVEGAN and 24 benchmark networks when they are integrated into PP and SRA coding tools for HEVC compression. All of them are benchmarked on the original HEVC - negative BD-rates indicate coding gains. The result sets {i/j/k} in this table stands for the BD-rate values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test network is also provided for comparison.

Compression performance comparisons with state-of-the-art network architectures

Network	CNN-based Post-Processing			CNN-based Spatial Resolution Adaptation		
	BD-rate (%) (PSNR)	BD-rate (%) (VMAF)	Relative Complexity	BD-rate (%) (PSNR)	BD-rate (%) (VMAF)	Relative Complexity
SRCNN [45]	-1.9/–/–	-7.4/–/–	1.0×	-3.1/–/–	-21.1/–/–	1.0×
FSRCNN [46]	-1.6/–/–	-7.3/–/–	1.37×	-4.5/–/–	-20.9/–/–	1.28×
VDSR [47]	-1.9/–/–	-7.6/–/–	2.05×	-6.6/–/–	-18.3/–/–	3.79×
DRRN [54]	-10.8/–/–	-14.9/–/–	2.70×	-15.0/–/–	-33.2/–/–	5.01×
EDSR [55]	-10.0/–/–	-14.6/–/–	4.50×	-13.4/–/–	-30.1/–/–	8.33×
SRResNet [53]	-9.8/–/–	-12.7/–/–	2.45×	-13.2/–/–	-30.0/–/–	4.46×
MSRResNet [73]	-10.4/–/–	-14.2/–/–	2.46×	-14.6/–/–	-32.7/–/–	4.52×
CARN [63]	-11.2/–/–	-15.4/–/–	2.23×	-15.5/–/–	-33.5/–/–	4.15×
UDSR[50]	-11.4/–/–	-16.0/–/–	3.04×	-15.7/–/–	-33.3/–/–	5.62×
HR-EnhanceNet [51]	-11.3/–/–	-16.4/–/–	2.80×	-15.8/–/–	-33.1/–/–	5.56×
ESRResNet [58]	-11.8/–/–	-17.7/–/–	3.82×	-16.1/–/–	-33.6/–/–	7.10×
RCAN [66]	-12.1/–/–	-18.5/–/–	4.82×	-17.1/–/–	-35.1/–/–	8.98×
RDN [59]	-12.2/–/–	-17.0/–/–	3.46×	-16.6/–/–	-34.5/–/–	6.41×
RNAN [20]	-12.5/-14.1/-11.7	-19.2/-23.9/-21.5	5.78×	-17.4/-9.3/–	-34.8/-31.6/–	10.79×
ADGAN [79]	-1.3/–/–	-7.7/–/–	2.46×	-5.1/–/–	-18.5/–/–	4.56×
SRResCGAN [98]	-7.1/–/–	-10.4/–/–	1.71×	-10.3/–/–	-27.2/–/–	3.16×
SRGAN [53]	-7.4/–/–	-12.9/–/–	2.46×	-10.9/–/–	-30.2/–/–	4.52×
PCARNGAN [64]	-8.3/–/–	-16.0/–/–	2.25×	-12.3/–/–	-33.7/–/–	4.18×
RCAGAN [75]	-9.1/–/–	-16.8/–/–	3.31×	-13.7/–/–	-33.9/–/–	6.13×
MSRGAN [73]	-6.5/-8.7/-5.9	-21.1/-25.7/-23.5	2.46×	-9.1/-3.3/–	-35.6/-32.9/–	4.54×
ESRGAN [58]	-8.7/–/–	-17.9/–/–	3.82×	-12.5/–/–	-33.8/–/–	7.15×
RCAN-GAN [99]	-9.3/–/–	-18.0/–/–	4.84×	-13.9/–/–	-34.1/–/–	9.03×
PatchESRGAN [80]	-9.0/–/–	-18.1/–/–	3.82×	-12.8/–/–	-34.2/–/–	7.20×
RFB-ESRGAN [74]	-9.1/-10.8/-7.9	-18.3/-23.2/-20.4	4.58×	-12.9/-4.5/–	-34.3/-31.0/–	8.52×
CVENet (Ours)	-9.5/-11.3/-8.5	-21.3/-26.0/-23.6	2.80×	-14.2/-5.9/–	-36.4/-33.3/–	5.23×
CVEGAN (Ours)	-10.2/-11.9/-9.0	**-23.4/-27.8/-25.3**	2.80×	-14.8/-6.4/–	**-38.4/-35.5/–**	5.23×

is not the best performer among all the tested networks, it outperforms all 24 architectures based on the perceptual quality metric VMAF. Considering that VMAF offers much higher correlation with subjective scores compared to PSNR [82, 107, 112], the effectiveness of the proposed algorithm in term of video quality enhancement is evident. The additional coding gains in terms of BD-rate (based on VMAF) compared to other networks are greater than 1.8% and

Table 3: Comprehensive compression results (in terms of BD-rate based on both PSNR and VMAF) of the proposed CVEGAN and its ablation study variants when they are integrated into PP and SRA coding tools for HEVC compression. All of them are benchmarked on the original HEVC - negative BD-rates indicate coding gains. The result sets {i/j/k} in this table stands for the BD-rate values for JVET-CTC, UVG and o-1-f respectively. The relative complexity of each test variant is also provided for comparison.

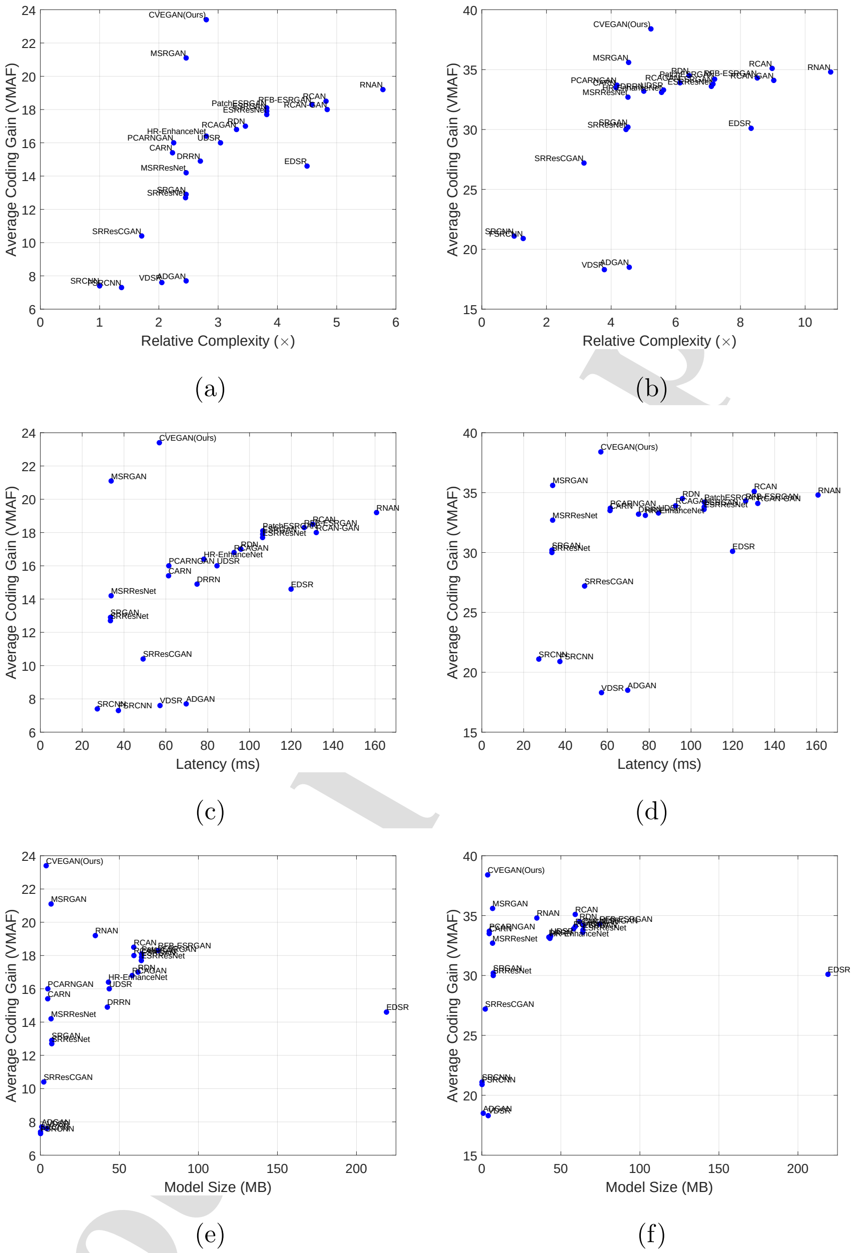
Ablation Study Variants

CVEGAN Variants		CNN-based Post-Processing			CNN-based Spatial Resolution Adaptation		
		BD-rate (%) (PSNR)	BD-rate (%) (VMAF)	Relative Complexity	BD-rate (%) (PSNR)	BD-rate (%) (VMAF)	Relative Complexity
w/o Mul^{2}Res block	w/ ResNeSt block [78]	-9.4/–/–	-22.3/–/–	2.81×	-14.3/–/–	-37.1/–/–	5.25×
	w/ RCAB block [66]	-9.0/–/–	-22.0/–/–	3.07×	-14.0/–/–	-36.6/–/–	5.72×
	w/ RRDB block [58]	-8.8/–/–	-21.6/–/–	3.16×	-13.8/–/–	-36.3/–/–	5.92×
	w/ RDB block [59]	-7.1/–/–	-20.9/–/–	2.95×	-12.3/–/–	-35.9/–/–	5.67×
	w/ ResNeXt block [72]	-6.7/–/–	-20.6/–/–	2.65×	-12.0/–/–	-35.7/–/–	4.85×
	w/ Xception block [76]	-6.6/–/–	-20.1/–/–	2.65×	-11.8/–/–	-35.2/–/–	4.92×
	w/ MRB block [73]	-6.3/–/–	-18.3/–/–	2.24×	-9.4/–/–	-34.6/–/–	4.03×
	w/ RB block [53]	-6.1/–/–	-16.6/–/–	2.24×	-8.6/–/–	-33.1/–/–	4.07×
w/o ERNB	w/ Non-local block [20]	-9.9/–/–	-22.1/–/–	3.22×	-14.5/–/–	-37.2/–/–	6.20×
w/o ECBAM	w/ CBAM block [21]	-10.0/–/–	-22.3/–/–	2.78×	-14.2/–/–	-37.0/–/–	5.22×
w/o ReSphereGAN	w/ SphereGAN [22]	-9.7/–/–	-22.1/–/–	2.80×	-14.3/–/–	-37.1/–/–	5.23×
	w/ RaGAN [58]	-9.6/–/–	-21.8/–/–	2.80×	-14.2/–/–	-36.8/–/–	5.26×
	w/ cGAN [75]	-9.6/–/–	-21.6/–/–	2.80×	-14.1/–/–	-36.7/–/–	5.26×
	w/ PatchGAN [80]	-9.4/–/–	-21.6/–/–	2.79×	-13.8/–/–	-36.8/–/–	5.31×
	w/ WGAN-GP [81]	-9.1/–/–	-21.7/–/–	2.80×	-13.2/–/–	-36.6/–/–	5.28×
	w/ Standard GAN [53]	-8.2/–/–	-21.5/–/–	2.80×	-12.6/–/–	-36.5/–/–	5.26×
w/o $\mathcal{L}_P$	w/ L_7 loss [98]	-4.5/–/–	-19.2/–/–	2.80×	-9.6/–/–	-34.1/–/–	5.21×
	w/ L_8 loss [73]	-7.5/–/–	-21.9/–/–	2.80×	-12.4/–/–	-37.0/–/–	5.23×
	w/ L_9 loss [58]	-5.7/–/–	-20.2/–/–	2.80×	-10.7/–/–	-35.4/–/–	5.21×
	w/ L_{10} loss [75]	-6.4/–/–	-21.4/–/–	2.80×	-11.5/–/–	-36.3/–/–	5.26×
	w/ L_{11} loss [53]	-5.8/–/–	-20.4/–/–	2.79×	-10.7/–/–	-35.5/–/–	5.20×

2.6% for PP and SRA respectively.

We also evaluated the models obtained after the first training step (trained with our perceptual loss function $\mathcal{L}_P$), denoted as CVENet in Table 3. Its overall performance is slightly lower than that of the final CVEGAN, with up to 2.1% and 2.2% BD-rate differences (based on VMAF) for PP and SRA respectively. This demonstrates the improvement due to the second training stage using the proposed ReSphereGAN.

Table 2 also shows the relative complexity of all test networks, which are



(a) (b) (c) (d) (e) (f)

Figure 9: Average coding gains (VMAF) versus relative complexities, latencies and model sizes for different network architectures for PP and SRA coding modules based on JVET-CTC dataset and HEVC HM 16.20: (a), (c) and (e) for PP coding module; (b), (d) and (f) for SRA coding module.

benchmarked on that of SRCNN. It is noted that the relative complexity of CVEGAN is only 2.8 times of that for SRCNN, which is relatively low compared to many network architectures including EDSR, UDSR, ESRResNet, RCAN, RDN, RNAN RCAGAN, ESRGAN, RCAN-GAN, PatchESRGAN and RFB-ESRGAN. The average coding gains (based on VMAF) versus relative complexities of different network architectures for PP and SRA coding modules based on JVET-CTC dataset and HEVC HM 16.20 are further shown in Figure 9 (a)-(b). Moreover, we also summarised the latency results[5] and model sizes[6] for different network architectures in Table 4 and Figure 9 (c)-(f) . It can be also observed that SRCNN has the shortest latency and smallest model size compared to other networks due to the simple network structures. The proposed CVEGAN has an appropriate latency and model size which are smaller than other complex networks, such as ESRResNet, RCAN, RDN, RNAN RCAGAN, RCAN-GAN, PatchESRGAN, and RFB-ESRGAN.

From the ablation study (Table 3 and Figure 10), we observe that the proposed Mul^{2}Res block, ERNB, ECBAM, ReSphereGAN and the new perceptual loss function have contributed at least 1.1%, 1.2%, 1.1%, 1.3% and 1.4% coding gains (assessed by VMAF) when compared to the tested replacements, but with similar or lower complexity. This shows the effectiveness of these new structures.

Table 5 summarises the compression results of the CVEGAN-based PP and SRA for VVC VTM 7.0. It can be observed that the proposed method provides evident coding gains on JVET tested sequences, with the average BD-rates of -8.0% and -20.3% based on VMAF for PP and SRA coding modules respectively.

Moreover, we have further analysed the overall BD-rate savings when employing different number of residual learning branches in Mul^{2}Res block (the size of the *cardinality* utilised at both first and second levels of Mul^{2}Res block as shown in Figure 2) and the different number of Mul^{2}Res blocks in CVEGAN

[5] Latency here is defined as the time taken to process one image block [70].

[6] For GAN algorithms, the generators' sizes are provided in the table.

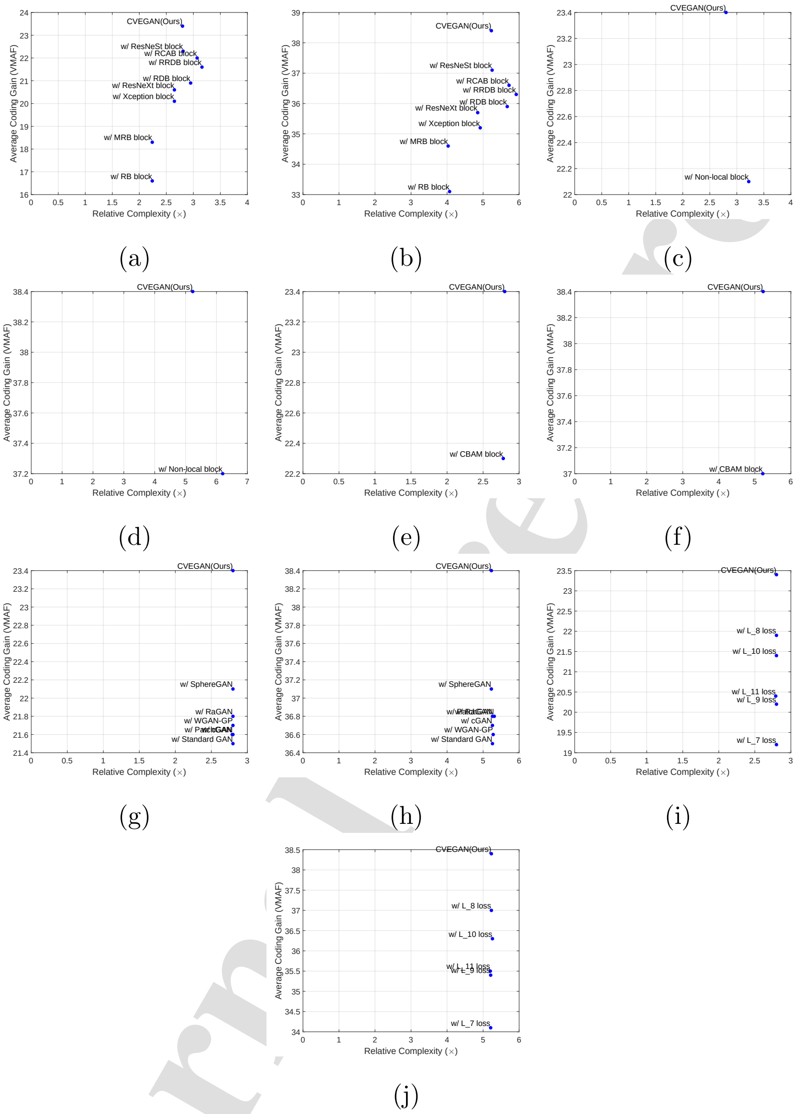


Figure 10: Average coding gains (VMAF) for different ablation studies for PP and SRA coding modules based on JVET-CTC dataset and HEVC HM 16.20: (a) ablation studies on Mul^{2}Res block (PP); (b) ablation studies on Mul^{2}Res block (SRA); (c) ablation studies on ERNB block (PP); (d) ablation studies on ERNB block (SRA); (e) ablation studies on CBAM block (PP); (f) ablation studies on CBAM block (SRA); (g) ablation studies on ReSphereGAN (PP); (h) ablation studies on ReSphereGAN block (SRA); (i) ablation studies on loss function $\mathcal{L}_P$ (PP) and (j) ablation studies on loss function $\mathcal{L}_P$ (SRA).

Table 4: Latency (millisecond-ms) and model sizes (megabyte-MB) of different network architectures. The result sets {i/j} in this table stands for the Latency results and model sizes, respectively.

	SRCNN	FSRCNN	VDSR	DRRN	EDSR
Latency (ms)/Model Size (MB)	27.24/0.119	37.32/0.146	57.21/4.04	74.91/42.4	119.86/219.0
	SRResNet	MSRResNet	CARN	UDSR	HR-EnhanceNet
Latency (ms)/Model Size (MB)	33.51/7.23	33.85/6.79	61.30/4.67	84.44/43.6	78.18/43.1
	ESRResNet	RCAN	RDN	RNAN	ADGAN
Latency (ms)/Model Size (MB)	106.24/63.9	130.21/59.1	95.88/61.8	160.72/34.8	69.73/0.901
	SRResCGAN	SRGAN	PCARNGAN	RCAGAN	MSRGAN
Latency (ms)/Model Size (MB)	49.13/2.17	33.54/7.23	61.42/4.73	92.62/58.1	33.86/6.81
	ESRGAN	RCAN-GAN	PatchESRGAN	RFB-ESRGAN	CVEGAN (Ours)
Latency (ms)/Model Size (MB)	106.29/64.0	131.91/59.2	106.33/64.1	126.12/74.5	56.89/3.69

for PP and SRA coding modules based on the HEVC HM 16.20. Figure 11 shows the overall coding gains (in terms of VMAF) using CVEGAN models with different numbers of residual learning branches (C=1, 2, 4, 6, 8, 10 and 12) and Mul^{2}Res blocks (N=3, 5, 7, 9, 11, 13 and 15) to process compressed JVET-CTC content. It can be observed that when the numbers of residual learning branches (C) and Mul^{2}Res blocks (N) increase from 1 to 4 and 3 to 7 respectively, the overall BD-rate savings effectively increase for both PP and SRA coding tools. However, when the numbers of residual learning branches and Mul^{2}Res blocks exceed 4 and 7 respectively, the overall coding gains start to decrease slightly for both two coding modules. These indicate that the size of the *cardinality* in Mul^{2}Res block and the number of Mul^{2}Res blocks employed

Table 5: Compression results of the CVEGAN-based PP and SRA for VTM 7.0.

Sequence (Class)	CNN-based PP		CNN-based SRA	
	BD-rate (PSNR)	BD-rate (VMAF)	BD-rate (PSNR)	BD-rate (VMAF)
Class A (2160p)	-2.2%	-12.1%	+4.4%	-20.3%
Class B (1080p)	-1.8%	-9.7%	–	–
Class C (480p)	-2.7%	-5.1%	–	–
Class D (240p)	-3.4%	-2.5%	–	–
Overall	-2.5%	-8.0%	+4.4%	-20.3%

in the proposed method are optimal selections.

6.2. Perceptual Comparisons

Figure 12 presents subjective comparison results among the proposed CVEGAN, the anchor HEVC and other top performing architectures for both PP and SRA[7]. The perceptual quality improvements associated with CVEGAN can be clearly observed in these examples and provide further validation of our approach.

We have conducted a lab-based subjective test on the results (for both PP and SRA coding modules) generated by CVEGAN and a selection of other tested network architectures (RNAN [20], MSRGAN [73] and RFB-ESRGAN [74]), which have achieved relatively higher coding gains according to VMAF.

Table 6 presents the average DMOS values of all evaluated sequences for CVEGAN, HEVC and the three top performing networks. The average DMOS for CVEGAN is lower than that for HEVC anchor and the other networks, providing further evidence of its effectiveness.

[7] More perceptual comparisons can be found at: `https://fan-aaron-zhang.github.io/CVE-GAN/TCSVT_CVEGAN.pdf`.

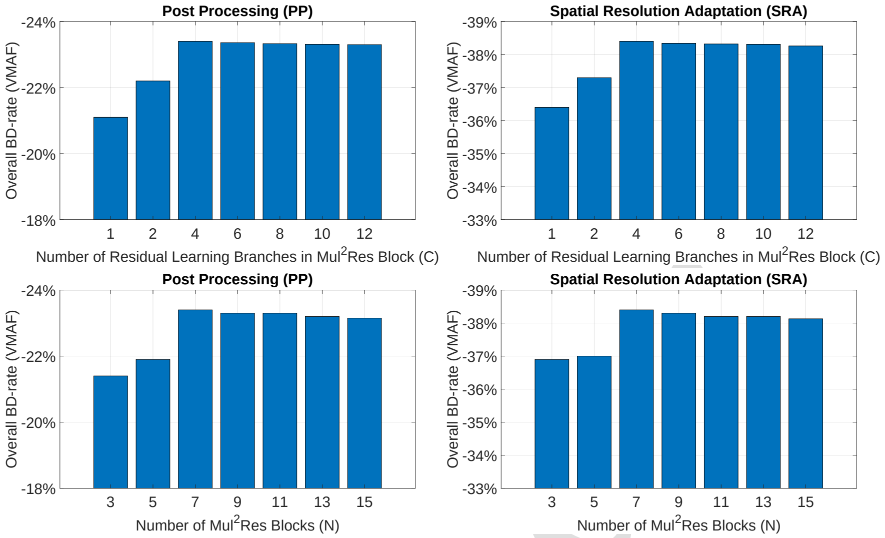


Figure 11: Overall BD-rate savings (VMAF) for different numbers of residual learning branches in Mul^{2}Res block and the number of Mul^{2}Res blocks used in the CVEGAN for PP and SRA coding modules based on the HEVC HM 16.20.

Table 6: Subjective results based on 12 UHD source sequences.

PP	Anchor (HM 16.20)	[20]	[73]	[74]	Ours
Average DMOS	1.97	1.61	1.57	1.58	**1.53**
SRA	Anchor (HM 16.20)	[20]	[73]	[74]	Ours
Average DMOS	1.97	1.38	1.40	1.42	**1.33**

7. Conclusion

In this paper, a novel GAN architecture, CVEGAN, has been proposed for compressed video quality enhancement. This network, when integrated into a conventional video coding system, has enabled significantly improved coding performance compared to many state-of-the-art architectures. This enhanced performance can be attributed to the use of several new features including a novel Mul^{2}Res blocks, ERNB, ECBAM, the new ReSphereGAN training methodology and perceptual-inspired loss functions. Future work will address

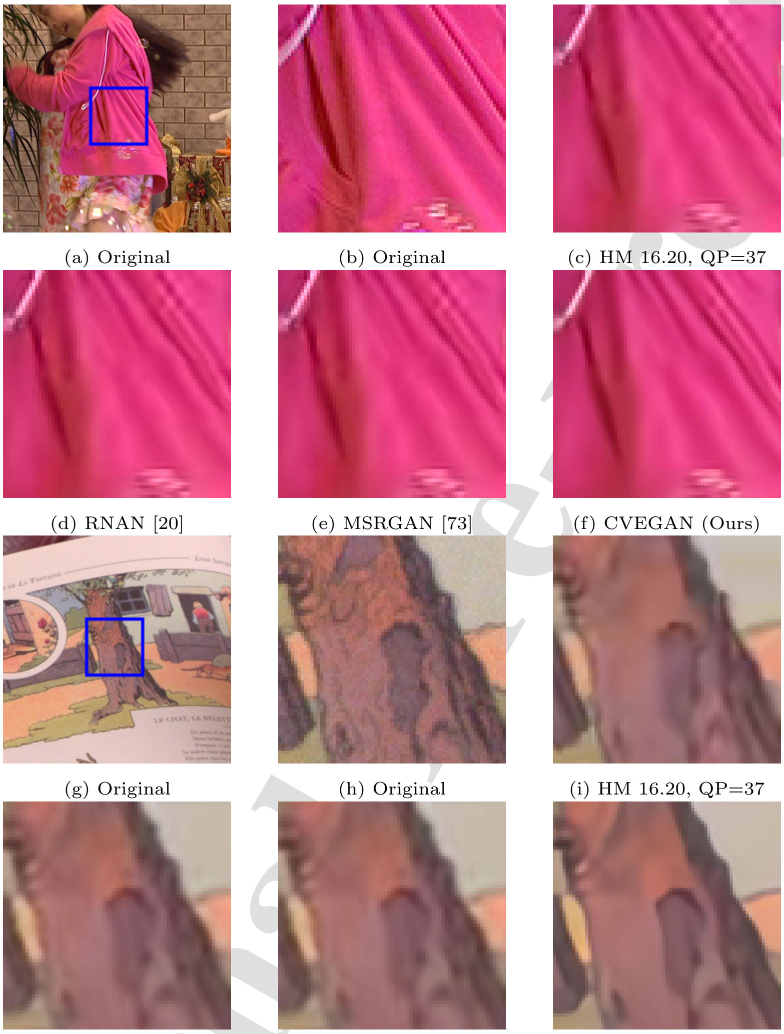


Figure 12: Two sets of example blocks cropped from the reconstructed frames generated by the anchor HM 16.20 (QP=37), RNAN [20], MSRGAN [73] and the proposed CVEGAN. The bitstreams for each example set consumed identical or similar bit rates. Row 1 corresponds to the 170th frame of the *PartyScene* sequence (for CNN-based PP) and Row 2 corresponds to the 104th frame of the *CatRobot1* sequence (for CNN-based SRA). It can be observed that the output of CVEGAN exhibits improved perceptual quality compared to HM 16.20, RNAN [20] and MSRGAN [73], with fewer blocking artefacts, more textural detail and higher contrast.

network complexity reduction, validate it on other standardised codecs and explore CVENet for distortion-oriented applications. We will also extend our approach to other application scenarios, including image restoration and super-resolution processing.

Appendix A. Mathematical Proof of Lemma 1

Lemma 1. $E(\left\|\nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}),T(\mathbf{x_f}))\right\|_2) < \infty$ for all m. Here ∇ represents the derivation operation, and $\|\cdot\|_2$ is the Euclidean norm.

Proof. Based on the definition provided in [22], the relativistic geodesic distance $d^m(T(\mathbf{x_r}),T(\mathbf{x_f}))$ can be written as:

$$\begin{aligned} & d^m(T(\mathbf{x_r}),T(\mathbf{x_f})) \\ = \quad & \arccos^m\left(\frac{\|\mathbf{x_r}\|_2^2\|\mathbf{x_f}\|_2^2-\|\mathbf{x_r}\|_2^2-\|\mathbf{x_f}\|_2^2+4\mathbf{x_r}\cdot\mathbf{x_f}+1}{(\|\mathbf{x_r}\|_2^2+1)(\|\mathbf{x_f}\|_2^2+1)}\right) \\ \equiv \quad & \arccos^m(A) \end{aligned} \tag{A.1}$$

Here $A \equiv \dfrac{\|\mathbf{x_r}\|_2^2\|\mathbf{x_f}\|_2^2-\|\mathbf{x_r}\|_2^2-\|\mathbf{x_f}\|_2^2+4\mathbf{x_r}\cdot\mathbf{x_f}+1}{(\|\mathbf{x_r}\|_2^2+1)(\|\mathbf{x_f}\|_2^2+1)}$, $A\in[-1,1]$. $\mathbf{x_r}$ and $\mathbf{x_f}$ are the real and fake feature points respectively in n-dimensional Euclidean feature space.

According to the chain rule,

$$\frac{\partial d^m(T(\mathbf{x_r}),T(\mathbf{x_f}))}{\partial\mathbf{x_r}} = \arccos^{m-1}(A)\cdot\frac{-m}{\sqrt{1-A^2}}\cdot\frac{\partial A}{\partial\mathbf{x_r}} \tag{A.2}$$

Based on the equation (A.2), the gradient of $d^m(T(\mathbf{x_r}),T(\mathbf{x_f}))$ can be further obtained following the chain rule and the product rule of derivative:

$$\begin{aligned} \nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}),T(\mathbf{x_f})) &= \frac{\partial^2 d^m(T(\mathbf{x_r}),T(\mathbf{x_f}))}{\partial\mathbf{x_r}\partial\mathbf{x_f}} \\ = \quad & m(m-1)\cdot\arccos^{m-2}(A)\cdot\frac{1}{1-A^2}\cdot\frac{\partial A}{\partial\mathbf{x_f}}\cdot\frac{\partial A}{\partial\mathbf{x_r}} - \\ & m\cdot\arccos^{m-1}(A)\cdot\frac{A}{(1-A^2)^{\frac{3}{2}}}\cdot\frac{\partial A}{\partial\mathbf{x_f}}\cdot\frac{\partial A}{\partial\mathbf{x_r}} - \\ & m\cdot\arccos^{m-1}(A)\cdot\frac{1}{\sqrt{1-A^2}}\cdot\frac{\partial^2 A}{\partial\mathbf{x_r}\partial\mathbf{x_f}} \end{aligned} \tag{A.3}$$

Based on Lemma 1 and 2 in [22, 100], we have

$$\arccos(A) < \infty,\ \frac{\partial A}{\partial\mathbf{x_f}}\frac{\partial A}{\partial\mathbf{x_r}} < \infty \text{ and} \frac{\partial^2 A}{\partial\mathbf{x_r}\partial\mathbf{x_f}} < \infty \tag{A.4}$$

According to Proposition 1 and 2 in [22] and Theorem 6.9 in [113], the geometric distance between the real and fake data feature points, $\mathbf{x_r}$ and $\mathbf{x_f}$, weakly converges to 0, for all moment values m:

$$d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) = \arccos^m(A) \rightharpoonup 0 \tag{A.5}$$

Here, ⇀ represents *weak convergence*. This indicates that $\arccos^m(A) \neq 0$, and thus $A \neq \pm 1$ [90]. Therefore we can have:

$$1 - A^2 \neq 0 \tag{A.6}$$

According to equation (A.4) and (A.6), the gradient (and its mean) of the relativistic geodesic distance is bounded for all moment values m:

$$\nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f})) < \infty, \tag{A.7}$$

$$E(\left\|\nabla_{(\mathbf{x_r},\mathbf{x_f})} d^m(T(\mathbf{x_r}), T(\mathbf{x_f}))\right\|_2) < \infty \tag{A.8}$$

□

Acknowledgment

The authors acknowledge funding from UK EPSRC (EP/L016656/1 and EP/M000885/1), MyWorld Strength in Places Programme and the NVIDIA GPU Seeding Grants.

References

[1] D. R. Bull, F. Zhang, Intelligent image and video compression: communicating pictures, Academic Press, 2021.

[2] C.-H. Yeh, Z.-T. Zhang, M.-J. Chen, C.-Y. Lin, HEVC intra frame coding based on convolutional neural network, IEEE Access 6 (2018) 50087–50095.

[3] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, W. Gao, Enhanced motion-compensated video coding with deep virtual reference frame generation, IEEE Transactions on Image Processing 28 (10) (2019) 4832–4844.

[4] S. Jimbo, J. Wang, Y. Yashima, Deep learning-based transformation matrix estimation for bidirectional interframe prediction, in: 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), IEEE, 2018, pp. 726–730.

[5] M. M. Alam, T. D. Nguyen, M. T. Hagan, D. M. Chandler, A perceptual quantization strategy for HEVC based on a convolutional neural network trained on natural images, in: Applications of Digital Image Processing XXXVIII, Vol. 9599, International Society for Optics and Photonics, 2015, p. 959918.

[6] C. Ma, D. Liu, X. Peng, F. Wu, Convolutional neural network-based arithmetic coding of DC coefficients for HEVC intra coding, in: 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, 2018, pp. 1772–1776.

[7] D. Ma, F. Zhang, D. Bull, MFRNet: a new CNN architecture for post-processing and in-loop filtering, IEEE Journal of Selected Topics in Signal Processing 15 (2) (2020) 378–387.

[8] D. Ma, F. Zhang, D. R. Bull, GAN-based effective bit depth adaptation for perceptual video compression, in: 2020 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2020, pp. 1–6.

[9] J. Ballé, V. Laparra, E. P. Simoncelli, End-to-end optimized image compression, in: Proceedings of the International Conference on Learning Representations (ICLR), 2017, pp. 1–27.

[10] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, D. Zhao, An end-to-end compression framework based on convolutional neural networks, IEEE Transactions on Circuits and Systems for Video Technology 28 (10) (2017) 3007–3018.

[11] D. Minnen, J. Ballé, G. D. Toderici, Joint autoregressive and hierarchical

priors for learned image compression, in: Advances in Neural Information Processing Systems, 2018, pp. 10771–10780.

[12] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, Z. Gao, DVC: an end-to-end deep video compression framework, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11006–11015.

[13] J. Lin, D. Liu, H. Li, F. Wu, M-LVC: Multiple frames prediction for learned video compression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 3546–3554.

[14] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, G. Toderici, Scale-space flow for end-to-end optimized video compression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8503–8512.

[15] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, S. Wang, Image and video compression with neural networks: A review, IEEE Transactions on Circuits and Systems for Video Technology 30 (6) (2019) 1683–1698.

[16] D. Liu, Y. Li, J. Lin, H. Li, F. Wu, Deep learning-based video coding: a review and a case study, ACM Computing Surveys (CSUR) 53 (1) (2020) 1–35.

[17] J. Wang, X. Deng, M. Xu, C. Chen, Y. Song, Multi-level wavelet-based generative adversarial network for perceptual quality enhancement of compressed video, Proceedings of the European Conference on Computer Vision (ECCV) (2020) 405–421.

[18] S. Zhang, L. Herranz, M. Mrak, M. G. Blanch, S. Wan, F. Yang, Dcn-gan: A deformable convolution-based gan with QP adaptation for perceptual quality enhancement of compressed video, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 2035–2039.

[19] L. Agnolucci, L. Galteri, M. Bertini, A. Del Bimbo, Perceptual quality improvement in videoconferencing using keyframes-based GAN, IEEE Transactions on Multimedia (2023) 1–14.

[20] Y. Zhang, K. Li, K. Li, B. Zhong, Y. Fu, Residual non-local attention networks for image restoration, in: Proceedings of the International Conference on Learning Representations (ICLR), 2019, pp. 1–18.

[21] S. Woo, J. Park, J.-Y. Lee, I. So Kweon, CBAM: convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.

[22] S. W. Park, J. Kwon, Sphere generative adversarial network based on geometric moment matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4292–4301.

[23] B. Bross, J. Chen, S. Liu, Y.-K. Wang, Versatile Video Coding (Draft 10), in: JVET-S2001. ITU-T and ISO/IEC, 2020.

[24] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, J.-R. Ohm, 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.

[25] ITU-T Rec. H.265, High Efficiency Video Coding, ITU-T Std., (2015).

[26] F. Zhang, A. V. Katsenou, M. Afonso, G. Dimitrov, D. R. Bull, Comparing VVC, HEVC and AV1 using objective and subjective assessments, arXiv preprint arXiv:2003.10282.

[27] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, L. V. Gool, Generative adversarial networks for extreme learned image compression, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 221–231.

[28] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, L. V. Gool, Practical full resolution learned lossless image compression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10629–10638.

[29] O. Rippel, S. Nair, C. Lew, S. Branson, A. G. Anderson, L. Bourdev, Learned video compression, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 3454–3463.

[30] A. Djelouah, J. Campos, S. Schaub-Meyer, C. Schroers, Neural inter-frame compression for video coding, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 6421–6429.

[31] A. Habibian, T. v. Rozendaal, J. M. Tomczak, T. S. Cohen, Video compression with rate-distortion autoencoders, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 7033–7042.

[32] J. Li, B. Li, J. Xu, R. Xiong, W. Gao, Fully connected network-based intra prediction for image coding, IEEE Transactions on Image Processing 27 (7) (2018) 3236–3247.

[33] Z. Zhao, S. Wang, S. Wang, X. Zhang, S. Ma, J. Yang, Enhanced bi-prediction with convolutional neural network for High Efficiency Video Coding, IEEE Transactions on Circuits and Systems for Video Technology 29 (11) (2018) 3291–3301.

[34] S. Puri, S. Lasserre, P. Le Callet, CNN-based transform index prediction in multiple transforms framework to assist entropy coding, in: 2017 25th European Signal Processing Conference (EUSIPCO), IEEE, 2017, pp. 798–802.

[35] R. Song, D. Liu, H. Li, F. Wu, Neural network-based arithmetic coding of intra prediction modes in HEVC, in: 2017 IEEE Visual Communications and Image Processing (VCIP), IEEE, 2017, pp. 1–4.

[36] H. Zhao, M. He, G. Teng, X. Shang, G. Wang, Y. Feng, A CNN-based post-processing algorithm for video coding efficiency improvement, IEEE Access 8 (2019) 920–929.

[37] W. Lin, X. He, X. Han, D. Liu, J. See, J. Zou, H. Xiong, F. Wu, Partition-aware adaptive switching neural networks for post-processing in HEVC, IEEE Transactions on Multimedia 22 (11) (2019) 2749–2763.

[38] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, S. Ma, Content-aware convolutional neural network for in-loop filtering in High Efficiency Video Coding, IEEE Transactions on Image Processing 28 (7) (2019) 3343–3356.

[39] J. Lin, D. Liu, H. Yang, H. Li, F. Wu, Convolutional neural network-based block up-sampling for HEVC, IEEE Transactions on Circuits and Systems for Video Technology 29 (12) (2018) 3701–3715.

[40] D. Ma, F. Zhang, D. R. Bull, Video compression with low complexity CNN-based spatial resolution adaptation, in: Applications of Digital Image Processing XLIII, Vol. 11510, International Society for Optics and Photonics, 2020, p. 115100D.

[41] F. Zhang, D. Ma, C. Feng, D. R. Bull, Video compression with CNN-based post processing, IEEE MultiMedia 28 (4) (2021) 74–83.

[42] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, Q. Dai, Residual highway convolutional neural networks for in-loop filtering in HEVC, IEEE Transactions on Image Processing 27 (8) (2018) 3827–3841.

[43] F. Zhang, M. Afonso, D. R. Bull, ViSTRA2: Video coding using spatial resolution and effective bit depth adaptation, Signal Processing: Image Communication 97 (2021) 116355.

[44] R. Yang, M. Xu, Z. Wang, T. Li, Multi-frame quality enhancement for compressed video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6664–6673.

[45] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE transactions on pattern analysis and machine intelligence 38 (2) (2015) 295–307.

[46] C. Dong, C. C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2016, pp. 391–407.

[47] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646–1654.

[48] C. Li, L. Song, R. Xie, W. Zhang, CNN based post-processing to improve HEVC, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 4577–4580.

[49] R. Yang, M. Xu, T. Liu, Z. Wang, Z. Guan, Enhancing quality for hevc compressed videos, IEEE Transactions on Circuits and Systems for Video Technology 29 (7) (2018) 2039–2054.

[50] J. Cai, S. Gu, R. Timofte, L. Zhang, NTIRE 2019 challenge on real image super-resolution: methods and results, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–13.

[51] A. Ignatov, R. Timofte, NTIRE 2019 challenge on image enhancement: methods and results, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–9.

[52] X. Meng, X. Deng, S. Zhu, X. Zhang, B. Zeng, A robust quality enhancement method based on joint spatial-temporal priors for video coding, IEEE Transactions on Circuits and Systems for Video Technology 31 (6) (2020) 2401–2414.

[53] C. Ledig, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4681–4690.

[54] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3147–3155.

[55] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 136–144.

[56] F. Li, H. Bai, Y. Zhao, FilterNet: adaptive information filtering network for accurate and fast image super-resolution, IEEE Transactions on Circuits and Systems for Video Technology 30 (6) (2020) 1511–1523.

[57] H. Huang, I. Schiopu, A. Munteanu, Frame-wise CNN-based filtering for intra-frame quality enhancement of HEVC videos, IEEE Transactions on Circuits and Systems for Video Technology 31 (6) (2020) 2100–2113.

[58] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C. Change Loy, ESRGAN: enhanced super-resolution generative adversarial networks, in: Proceedings of the European conference on computer vision (ECCV) workshops, 2018, pp. 1–16.

[59] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2472–2481.

[60] Y. Shi, S. Li, W. Li, A. Liu, Fast and lightweight image super-resolution based on dense residuals two-channel network, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 2826–2830.

[61] N. C. Rakotonirina, A. Rasoanaivo, ESRGAN+: further improving enhanced super-resolution generative adversarial network, in: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 3637–3641.

[62] H. Zhao, B. Zheng, S. Yuan, H. Zhang, C. Yan, L. Li, G. Slabaugh, CBREN: convolutional neural networks for constant bit rate video quality enhancement, IEEE Transactions on Circuits and Systems for Video Technology 32 (7) (2021) 4138–4149.

[63] N. Ahn, B. Kang, K.-A. Sohn, Fast, accurate, and lightweight super-resolution with cascading residual network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 252–268.

[64] N. Ahn, B. Kang, K.-A. Sohn, Photo-realistic image super-resolution with fast and lightweight cascading residual network, arXiv preprint arXiv:1903.02240.

[65] R. Feng, J. Gu, Y. Qiao, C. Dong, Suppressing model overfitting for image super-resolution networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–10.

[66] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 286–301.

[67] T. Dai, H. Zha, Y. Jiang, S.-T. Xia, Image super-resolution via residual block attention networks, in: Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 2019, pp. 3879–3886.

[68] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7794–7803.

[69] D. Liu, B. Wen, Y. Fan, C. C. Loy, T. S. Huang, Non-local recurrent network for image restoration, arXiv preprint arXiv:1806.02919.

[70] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[71] S. Zagoruyko, N. Komodakis, Wide residual networks, arXiv preprint arXiv:1605.07146.

[72] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1492–1500.

[73] D. Ma, M. F. Afonso, F. Zhang, D. R. Bull, Perceptually-inspired super-resolution of compressed videos, in: Applications of Digital Image Processing XLII, Vol. 11137, International Society for Optics and Photonics, 2019, pp. 310–318.

[74] T. Shang, Q. Dai, S. Zhu, T. Yang, Y. Guo, Perceptual extreme super-resolution network with receptive field block, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1–10.

[75] J. Cai, Z. Meng, C. Man Ho, Residual channel attention generative adversarial network for image super-resolution and noise reduction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 454–455.

[76] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1251–1258.

[77] K. Zhang, et al., Aim 2019 challenge on constrained super-resolution: Methods and results, in: Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), IEEE, 2019, pp. 3565–3574.

[78] H. Zhang, et al., ResNeSt: Split-attention networks, arXiv preprint arXiv:2004.08955.

[79] K. Lin, T. H. Li, S. Liu, G. Li, Real photographs denoising with noise domain adaptation and attentive generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–5.

[80] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, F. Huang, Real-world super-resolution via kernel estimation and noise injection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1–10.

[81] W. Wang, R. Guo, Y. Tian, W. Yang, CFSNet: Toward a controllable feature space for image restoration, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 4140–4149.

[82] F. Zhang, F. M. Moss, R. Baddeley, D. R. Bull, BVI-HD: A video quality database for HEVC compressed and texture synthesized content, IEEE Transactions on Multimedia 20 (10) (2018) 2620–2630.

[83] L.-H. Chen, C. G. Bampis, Z. Li, A. Norkin, A. C. Bovik, Perceptually optimizing deep image compression, arXiv preprint arXiv:2007.02711.

[84] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[85] D. Misra, Mish: a self regularized non-monotonic neural activation function, arXiv preprint arXiv:1908.08681.

[86] X. Wang, K. C. Chan, K. Yu, C. Dong, C. Change Loy, EDVR: Video restoration with enhanced deformable convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–10.

[87] R. Muhammad Umer, G. Luca Foresti, C. Micheloni, Deep generative adversarial residual convolutional networks for real-world super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 438–439.

[88] Z. Wang, et al., Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

[89] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol. 2, Ieee, 2003, pp. 1398–1402.

[90] W. Rudin, et al., Principles of mathematical analysis, Vol. 3, McGraw-hill New York, 1964.

[91] D. C. Howell, Statistical methods for psychology, Cengage Learning, 2012.

[92] Netflix Public Dataset, `https://github.com/Netflix/vmaf/blob/master/resource/doc/datasets.md`.

[93] A. V. Katsenou, F. Zhang, M. Afonso, D. R. Bull, A subjective comparison of AV1 and HEVC for adaptive video streaming, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 4145–4149.

[94] J. Y. Lin, R. Song, C.-H. Wu, T. Liu, H. Wang, C.-C. J. Kuo, MCL-V: a streaming video quality assessment database, Journal of Visual Communication and Image Representation 30 (2015) 1–9.

[95] SHVC verification test results, in: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG, ITU-T SG16 WP3, ISO/IEC JTC1/SC29/WG, 2016.

[96] IVP Subjective Quality Video Database, `http://ivp.ee.cuhk.edu.hk/research/database/subjective/`.

[97] VQEG-HD3, `https://www.its.bldrdoc.gov/vqeg/projects/hdtv/hdtv.aspx`.

[98] R. Muhammad Umer, G. Luca Foresti, C. Micheloni, Deep generative adversarial residual convolutional networks for real-world super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1–9.

[99] H. Ren, A. Kheradmand, M. El-Khamy, S. Wang, D. Bai, J. Lee, Real-world super-resolution using generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1–9.

[100] S. W. Park, J. Kwon, SphereGAN: Sphere generative adversarial network based on geometric moment matching and its applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (3) (2020) 1566–1580.

[101] A. Jolicoeur-Martineau, The relativistic discriminator: a key element missing from standard GAN, arXiv preprint arXiv:1807.00734.

[102] D. P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[103] D. Ma, F. Zhang, D. R. Bull, BVI-DVC: A training database for deep video compression, IEEE Transactions on Multimedia 24 (2021) 3847–3858.

[104] S. Liu, et al., Report of AHG11 meeting on neural-network-based video coding, in: the JVET meeting, no. JVET-T0042, Teleconference: ITU-T, ISO/IEC, 2020.

[105] F. Bossen, J. Boyce, X. Li, V. Seregin, K. Suhring, JVET common test conditions and software reference configurations for SDR video, in: the JVET meeting, no. JVET-M1001, ITU-T, ISO/IEC, 2019.

[106] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, in: 13th VCEG Meeting, no. VCEG-M33,Austin, Texas, 2001, pp. USA: ITU–T.

[107] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, M. Manohara, Toward a practical perceptual video quality metric, The Netflix Tech Blog 6.

[108] H. R. Sheikh, A. C. Bovik, G. De Veciana, An information fidelity criterion for image quality assessment using natural scene statistics, IEEE Transactions on Image Processing 14 (12) (2005) 2117–2128.

[109] A. Mercat, M. Viitanen, J. Vanne, UVG dataset: 50/120fps 4k sequences for video codec analysis and development, in: Proceedings of the 11th ACM Multimedia Systems Conference, 2020, pp. 297–302.

[110] Y. Chen, et al., An overview of coding tools in AV1: the first video codec from the alliance for open media, APSIPA Transactions on Signal and Information Processing 9 (2020) e6.

[111] Methodology for the subjective assessment of the quality of television pictures, in: BT.500-11, ITU-R, 2002.

[112] A. Mackin, D. Ma, F. Zhang, D. Bull, A subjective study on videos at various bit depths, in: 2021 Picture Coding Symposium (PCS), IEEE, 2021, pp. 1–5.

[113] C. Villani, Optimal transport: old and new, Vol. 338, Springer Science & Business Media, 2008.

- Presenting a novel block structure, Mul^{2}Res which is the first use of a nested residual learning structure with various kernel sizes.
- Employing enhanced residual non-local blocks and enhanced convolutional block attention modules to improve the representational capability of the network.
- Designing a new GAN training methodology, Relativistic SphereGAN to achieve better training performance.
- Proposing a novel perceptual loss function to further optimise video quality during training.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

David R. Bull reports financial support was provided by Engineering and Physical Sciences Research Council.