

DASR-Net: A Deformable Sampling and Attentive State-Space Restoration Network for In-Loop Filtering in Versatile Video Coding

Yue Wen^{1,a}, Long Xu^{2,b}, Xuande Zhang^{1,c,*}

¹*School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi'an, 710021, China*

²*Faculty of Information Science and Engineering, Ningbo University, Ningbo, 315211, China*

^a3023210855@qq.com, ^bxulong1@nbu.edu.cn, ^c330374142@qq.com

*Corresponding author

Abstract: The rapid development of ultra-high-definition video and streaming media applications has imposed increasingly stringent requirements on reconstruction quality and coding efficiency in H.266/Versatile Video Coding (VVC). Although existing CNN- and Transformer-based learned in-loop filtering methods can improve reconstruction quality, they still face challenges in jointly modeling spatially irregular local artifacts, long-range structural dependencies, and cross-component luma-chroma correlations. To address these issues, this paper proposes a Deformable Sampling and Attentive State-Space Restoration Network (DASR-Net) for learned in-loop filtering in VVC. The proposed network employs a Deformable Sampling Module to adaptively model and suppress spatially irregular local artifacts, including blocking artifacts along block boundaries and ringing artifacts around texture edges. An Attentive State-Space Module is introduced to model long-range structural dependencies and non-local texture correlations, while a Luma-Guided Chroma Fusion module exploits structural cues from the luma component to guide chroma restoration. Experimental results show that, under the All-Intra configuration, DASR-Net achieves average BD-rate savings of 8.96%, 17.64%, and 19.86% for the Y, U, and V components, respectively, relative to the VTM-23.13 anchor, demonstrating the effectiveness of the proposed method.

Keywords: Versatile Video Coding, Learned In-Loop Filtering, Deformable Sampling, Attentive State-Space Modeling, Luma-Guided Chroma Restoration

1. Introduction

With the rapid development of ultra-high-definition video, video conferencing, streaming media transmission, and immersive video applications, the volume of video data continues to grow, placing increasingly stringent demands on coding efficiency^[1]. As a new-generation video coding standard, H.266/Versatile Video Coding (VVC) introduces more flexible tools for block partitioning, intra- and inter-prediction, transform, and quantization than its predecessor, H.265/High Efficiency Video Coding (HEVC), thereby substantially improving rate-distortion performance^[1, 2]. Nevertheless, VVC retains a block-based hybrid coding framework, and the resulting reconstructed frames produced by lossy compression may still exhibit blocking artifacts, ringing artifacts, texture blurring, and other local distortions caused by block partitioning, transform-coefficient quantization, and prediction errors^[1, 3]. To alleviate these distortions, VVC incorporates several filtering tools into the coding loop. The Deblocking Filter (DBF) suppresses blocking artifacts by smoothing pixel discontinuities across block boundaries; Sample Adaptive Offset (SAO) compensates for quantization-induced offsets according to statistical characteristics; the Adaptive Loop Filter (ALF) employs Wiener filtering to minimize reconstruction distortion; and the Cross-Component Adaptive Loop Filter (CC-ALF) exploits correlations between the luma and chroma components to reduce chroma distortion^[3]. Although these conventional in-loop filters can effectively improve reconstruction quality, their designs rely primarily on handcrafted rules and local statistical assumptions and may therefore be less effective in adaptively modeling complex textures, non-stationary content, and coexisting compression artifacts.

In recent years, deep learning techniques have been widely applied to in-loop filtering and compression artifact removal in video coding. Early approaches mainly employed convolutional neural

networks (CNNs) or residual networks to learn nonlinear mappings from reconstructed frames to high-quality frames^[4-8]. Various coding priors, including prediction information, block partitioning information, quantization parameters (QPs), and luma samples, were further incorporated to improve the adaptability of these models to different compression levels and distortion types^[9-12]. In addition, mechanisms such as multi-level feature reuse, multi-level in-loop filtering, luma-guided chroma restoration, and content-aware dynamic networks have been explored to improve the performance and flexibility of learned in-loop filtering^[13-17]. Although these methods improve the quality of VVC reconstructed frames to varying degrees, most of them rely primarily on local convolutions or local-window feature modeling and may therefore be less effective in exploiting cross-region structural correlations and long-range structural dependencies.

YUV 4:2:0 is a commonly used chroma subsampling format in digital video coding, where Y denotes the luma component and U and V denote the two chroma components. The luma component contains twice as many samples as each chroma component along both the horizontal and vertical dimensions. Owing to its higher spatial resolution, the luma component generally preserves richer information about edge contours, texture details, and block boundaries, and its restoration quality has a substantial influence on both the subjective visual quality and objective quality metrics of reconstructed frames. Compression distortions are mainly manifested as blocking artifacts, ringing artifacts, texture blurring, and the loss of high-frequency details. These distortions exhibit distinct local characteristics while also being associated with cross-region structural correlations and long-range structural dependencies. CNNs rely on local convolutional operations, and long-range structural information generally needs to be propagated progressively through multiple stacked local operations. Transformers provide stronger global-context modeling capabilities, but their application to high-resolution reconstructed frames may incur considerable computational overhead, while window-based attention may restrict information interaction across different windows^[18-20]. By contrast, Mamba employs selective state-space modeling to capture long-range structural dependencies and non-local texture correlations with relatively high computational efficiency^[21-23]. In particular, MambaIRv2 facilitates information exchange among spatially distant yet similar regions through attentive state-space modeling and a semantic-guided neighboring mechanism, providing a new approach to long-range structural modeling in reconstructed frames^[23].

Based on the above analysis, this paper proposes a learned in-loop filtering network for VVC, termed the Deformable Sampling and Attentive State-Space Restoration Network (DASR-Net). The network incorporates a Deformable Sampling Module (DSM) and an Attentive State-Space Module (ASSM). DASR-Net takes VVC reconstructed frames generated with the conventional DBF, SAO, ALF, and CC-ALF tools disabled as input and learns residual mappings associated with compression distortions in the feature space to suppress compression artifacts and restore structural details. Specifically, the ASSM captures long-range structural dependencies and non-local texture correlations, whereas the DSM performs content-adaptive sampling and feature aggregation to facilitate the restoration of spatially irregular local artifacts, including blocking artifacts along block boundaries and ringing artifacts around texture edges. Furthermore, because the chroma components have lower spatial resolution and contain relatively limited structural information, the proposed method draws inspiration from CC-ALF in VVC and exploits structural information from the luma component to guide chroma restoration^[3].

The main contributions of this work are summarized as follows:

- (1) A learned in-loop filtering network, termed DASR-Net, is proposed for VVC. The network incorporates attentive state-space modeling into compression artifact restoration to capture cross-region structural correlations and long-range structural dependencies.
- (2) A DSM is designed to enable content-adaptive sampling and feature aggregation for the restoration of spatially irregular local distortions, including blocking artifacts along block boundaries and ringing artifacts around texture edges.
- (3) A Luma-Guided Chroma Fusion (LGCF) module is designed to exploit structural information from the luma component to guide chroma restoration, thereby mitigating chroma-edge blurring, color bleeding, and local color shifts.

2. Related Work

In-loop filtering is a key reconstruction enhancement tool in hybrid video coding frameworks and is primarily used to suppress compression distortions introduced during block-based prediction, transform, and quantization^[1, 3]. VVC employs filtering tools including DBF, SAO, ALF, and CC-ALF to address

block-boundary discontinuities, quantization-induced offsets, reconstruction distortion, and cross-component chroma distortion, respectively^[3]. Although these conventional filters improve coding efficiency and reconstruction quality, their reliance on handcrafted rules and local statistical assumptions may limit their adaptability to complex textures, non-stationary content, and coexisting compression artifacts.

With the development of deep learning for image restoration and compression artifact removal, CNN-based methods have been introduced into video coding for in-loop filtering. Dong et al.^[4] employed a CNN to learn a mapping from compressed images to high-quality images, while Jia et al.^[5] proposed a content-aware multi-model filtering mechanism. Subsequent studies improved CNN-based in-loop filtering through the use of QP information, filter switching, and residual feature reuse^[6-8]. To improve the adaptability of learned models under different coding conditions, subsequent approaches incorporated coding priors such as QP, block partitioning, prediction information, and luma samples. Wang et al.^[9] proposed an attention-based dual-scale CNN using QP and partition priors. Huang et al.^[10, 11] introduced QP-conditioned feature modulation to adapt a single network to different QPs and frame types. Li et al.^[12] further exploited coding side information and luma samples for luma and chroma restoration. In addition, MFRNet, multi-level in-loop filtering, and DILF improved model performance and flexibility through multi-level feature reuse, filtering at multiple coding levels, and dynamic execution paths, respectively^[13-15].

Beyond CNN-based methods, several studies further incorporated frequency-domain information and Transformer architectures. Kathariya et al.^[16, 17] jointly modeled pixel-domain and DCT-domain features and employed Transformers for feature fusion. Zhang et al.^[18] proposed RTNN, which combines the local modeling capability of residual blocks with the long-range dependency modeling capability of Transformers. SwinIR and Restormer reduce the computational cost of high-resolution image restoration through window-based attention and channel-wise attention, respectively^[19, 20].

Despite these advances, jointly modeling spatially irregular local artifacts and long-range structural dependencies remains challenging. Gu and Dao^[21] proposed Mamba for long-sequence modeling with linear-time complexity. MambaIR^[22] subsequently applied Mamba to image restoration and enhanced local feature modeling. MambaIRv2^[23] further introduced an attentive state-space equation and a semantic-guided neighboring mechanism to facilitate information exchange among spatially distant yet similar regions. Nevertheless, existing state-space models have mainly been developed for image restoration tasks such as super-resolution, image denoising, and compressed-image restoration, and their application to learned in-loop filtering for VVC remains relatively limited.

Overall, existing learned in-loop filtering methods have improved compression artifact restoration through coding priors, frequency-domain features, and attention mechanisms, while state-space modeling has demonstrated promising long-range dependency modeling capabilities in image restoration. However, some methods rely on additional coding priors or dedicated frequency-domain branches, while simultaneously modeling spatially irregular local artifacts and long-range structural dependencies remains challenging. In addition, chroma restoration may be constrained by the relatively limited structural information available in the chroma components. To address these limitations, this work proposes a complementary local-global restoration network by combining the DSM with the ASSM, thereby enabling unified modeling of spatially irregular local artifacts and long-range structural dependencies. The LGCF module is further designed to exploit structural information from the luma component for chroma restoration.

3. Proposed Method

3.1. Overall Architecture of DASR-Net

The overall architecture of DASR-Net is illustrated in Figure 1. The proposed network comprises a shallow feature extraction module, a luma restoration branch, a luma-guided chroma restoration pathway consisting of separate U-restoration and V-restoration branches, and reconstruction modules. Under the experimental setting adopted in this work, the network takes a reconstructed frame as input and processes its luma and chroma components through separate restoration branches, thereby enabling coordinated luma-chroma enhancement.

Given an input reconstructed frame I_{rec} consisting of one luma component and two chroma components, it is represented as:

$$I_{\text{rec}} = \{Y_{\text{rec}}, U_{\text{rec}}, V_{\text{rec}}\} \quad (1)$$

where $Y_{\text{rec}} \in \mathbb{R}^{H \times W \times 1}$ denotes the reconstructed luma component, while $U_{\text{rec}}, V_{\text{rec}} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 1}$ denote the two reconstructed chroma components. Here, H and W represent the height and width of the luma component, respectively. The batch dimension is omitted throughout this paper, and all tensors are represented in height–width–channel order.

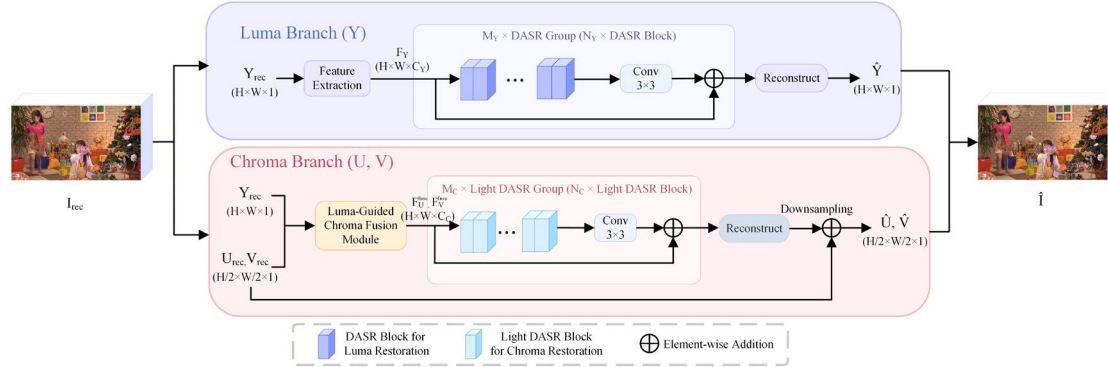


Figure 1: Overall architecture of DASR-Net.

For the luma restoration branch, the input luma component Y_{rec} is first processed by a 3×3 shallow feature extraction convolution with a stride of 1 and a padding size of 1. This layer maps the single-channel input to a C_Y -dimensional feature space, producing the initial luma feature $F_Y^0 \in \mathbb{R}^{H \times W \times C_Y}$, where C_Y denotes the number of feature channels in the luma restoration branch and is set to 174 in this work. The initial luma feature is subsequently fed into a luma restoration backbone comprising M_Y cascaded DASR Groups. Each DASR Group contains N_Y sequentially connected DASR Blocks, a 3×3 convolutional layer at the end of the group, and a group-level residual connection. The DASR Blocks perform spatially irregular local artifact restoration and long-range structural modeling, while the group-level residual connection preserves the input information and facilitates deep feature propagation. In the proposed implementation, the luma restoration backbone contains $M_Y = 6$ DASR Groups, with each group comprising $N_Y = 4$ DASR Blocks, resulting in a total of 24 DASR Blocks. After the six DASR Groups, the backbone output is fused through a 3×3 convolutional layer and added element-wise to the initial luma feature F_Y^0 through a feature-level residual connection. The resulting feature is then directly reconstructed as a single-channel enhanced luma component through a 3×3 output convolutional layer: $\hat{Y} \in \mathbb{R}^{H \times W \times 1}$. It should be noted that the residual connections in the luma branch operate in the feature space. No image-level residual addition between the network output and the input luma component is applied at the output of the luma branch.

For the reconstructed chroma components U_{rec} and V_{rec} , the LGCF module exploits structural information from the luma component Y_{rec} to generate luma-guided chroma fusion features. The detailed upsampling, concatenation, and feature-mapping operations of the LGCF module are described in Section 3.3. The fused U and V features are then fed into two lightweight chroma restoration branches with identical architectures but independent parameters. The number of feature channels in each chroma restoration branch is denoted by C_C and is set to 60. Each chroma restoration branch comprises $M_C = 4$ cascaded lightweight DASR Groups, with each group containing $N_C = 4$ lightweight DASR Blocks, resulting in a total of 16 lightweight DASR Blocks per branch. Compared with the luma restoration branch, the lightweight chroma restoration branches employ simplified configurations in terms of feature channels, window size, state-space parameters, and network depth. The chroma restoration branches predict chroma compensation residuals at the luma resolution. These residuals are subsequently downsampled to the native chroma resolution and added to the corresponding input chroma components through image-level residual connections, producing the enhanced chroma components $\hat{U}$ and $\hat{V}$.

Finally, the enhanced luma and chroma components are recombined according to the YUV 4:2:0 chroma subsampling format to obtain the final enhanced reconstructed frame:

$$\hat{I} = \{\hat{Y}, \hat{U}, \hat{V}\} \quad (2)$$

where $\hat{Y} \in \mathbb{R}^{H \times W \times 1}$, $\hat{U}, \hat{V} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 1}$. The DASR Block and its internal DSM and ASSM are described in detail in Section 3.2, while the LGCF module is presented in Section 3.3.

3.2. DASR Block

This section introduces the core feature-modeling unit of DASR-Net, termed the DASR Block, whose structure is shown in Figure 2. The input feature is denoted as $F_{in} \in \mathbb{R}^{H_f \times W_f \times C}$, where H_f , W_f and C denote the height, width, and number of channels of the current feature map, respectively. For the luma restoration branch, $C = C_Y$, whereas for the chroma restoration branches, $C = C_C$.

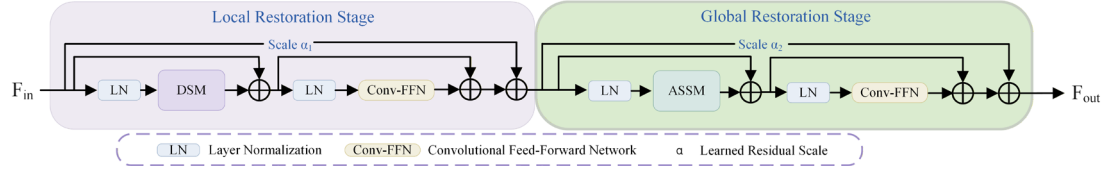


Figure 2: Structure of the DASR Block.

The DASR Block consists of a local restoration stage followed by a global restoration stage. Both stages combine pre-normalization, feature modeling, a Convolutional Feed-Forward Network (Conv-FFN), and residual connections. In the local restoration stage, the input feature is first processed by Layer Normalization and subsequently fed into the DSM for content-adaptive local sampling and feature aggregation. This process improves the modeling of spatially irregular local artifacts, including blocking artifacts along block boundaries and ringing artifacts around texture edges. The DSM output is further processed by another LN layer and the Conv-FFN for feature transformation. Residual connections are applied to both the DSM and Conv-FFN, while a learnable channel-wise scaling factor modulates the output of the local restoration stage. In the global restoration stage, the locally enhanced feature is first processed by LN and then fed into the ASSM for long-range structural dependency modeling. The ASSM facilitates information exchange among spatially distant yet structurally similar regions and captures cross-region structural correlations. The resulting feature is subsequently processed by another LN layer and Conv-FFN for further feature transformation. Residual connections are likewise applied to both the ASSM and Conv-FFN, while another learnable channel-wise scaling factor modulates the output of the global restoration stage.

Through this sequential design, the DASR Block performs spatially irregular local artifact modeling followed by long-range structural dependency modeling within a single basic unit, thereby establishing a complementary local-global feature restoration mechanism. In Figure 2, α_1 and α_2 denote the learnable channel-wise scaling factors of the local and global restoration stages, respectively. The detailed implementations of the DSM and ASSM are presented below.

3.2.1. Deformable Sampling Module

Compression artifacts around block boundaries and texture edges often vary in spatial location, shape, and strength. Fixed convolutional kernels or regular sampling patterns may therefore be less effective in accommodating such content-dependent distortions. To address this issue, the proposed DSM draws on the content-adaptive sampling principles of deformable convolution and deformable attention [24, 25] and adapts them to compression artifact restoration. Its structure is illustrated in Figure 3.

Rather than retaining the full query-key correlation computation used in attention-based formulations, the DSM confines content-adaptive sampling to local windows and omits explicit key construction. It mainly comprises query and value feature projections, offset and weight prediction, deformable sampling, weighted aggregation, and output projection. This design improves the flexibility of local feature modeling while controlling computational overhead.

Given an input feature $F \in \mathbb{R}^{H_f \times W_f \times C}$, the DSM first generates the query feature Q and value feature V through linear projections:

$$Q = \phi_Q(F), \quad V = \phi_V(F) \quad (3)$$

where $\phi_Q(\cdot)$ and $\phi_V(\cdot)$ denote the query and value projection functions, respectively. The query feature Q is used to predict local sampling offsets and aggregation weights, while the value feature V provides the source features for subsequent deformable sampling. Unlike standard self-attention, the DSM does not explicitly construct key features or compute query-key correlations. For a query position p_i in the feature map, the DSM defines K regular reference sampling points within a local window. The k -th deformable sampling position is formulated as:

$$\tilde{p}_{i,k} = p_i + r_k + \Delta p_{i,k}, k = 1, 2, \dots, K \quad (4)$$

where r_k denotes the base offset of the k -th regular reference point relative to the query position p_i , and $\Delta p_{i,k}$ denotes the content-adaptive offset predicted from the query feature.

Because the deformable sampling positions generally lie at non-integer coordinates, the DSM extracts the corresponding feature responses from the value feature through bilinear interpolation and aggregates them according to the predicted weights:

$$F_{\text{DSM}}(p_i) = \phi_o \left(\sum_{k=1}^K a_{i,k} \mathcal{B}(V, \tilde{p}_{i,k}) \right) \quad (5)$$

where $a_{i,k}$ denotes the normalized aggregation weight associated with the k -th sampling point, $\mathcal{B}(\cdot)$ denotes bilinear interpolation sampling, and $\phi_o(\cdot)$ denotes the output projection. The aggregation weights are normalized across the K sampling points corresponding to the same query position.

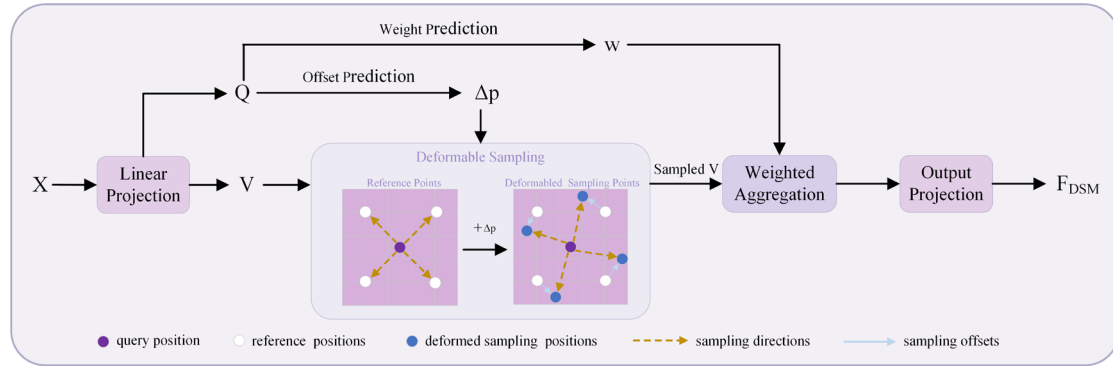


Figure 3: Structure of the Deformable Sampling Module.

Through this content-adaptive sampling process, the DSM dynamically adjusts the sampling positions according to local spatial structures, thereby enhancing the restoration of spatially irregular local artifacts around block boundaries and texture edges. The effectiveness of the DSM is further evaluated through ablation experiments in Section 4.4.

3.2.2. Attentive State-Space Module

The DSM primarily performs adaptive sampling and feature aggregation within local windows. To further expand the range of feature interaction, the ASSM from MambaIRv2^[23] is incorporated into the DASR Block to model long-range structural dependencies in features extracted from reconstructed frames. Given an input feature $F_d \in \mathbb{R}^{H_f \times W_f \times C}$, the ASSM first applies positional encoding to preserve the original spatial information. The Semantic Guided Neighboring (SGN) mechanism then rearranges and serializes the two-dimensional feature map such that spatially distant yet semantically similar positions become adjacent in the resulting one-dimensional sequence. The reordered features are subsequently processed by the Attentive State-Space Equation (ASE).

Unlike standard Mamba^[21], which mainly relies on causal modeling of preceding states along the scanning sequence, the ASE employs instance-specific prompts to supplement semantic information beyond the preceding scanned context. Its basic formulation is given by

$$h_i = Ah_{i-1} + Bx_i, \quad y_i = (C_i + P_i)h_i + Dx_i \quad (6)$$

where x_i , h_i , and y_i denote the input feature, state variable, and output feature at the i -th sequence position, respectively. A , B , C_i , and D denote the state-transition, input, output, and direct-feedthrough matrices, respectively. P_i denotes an instance-specific prompt dynamically retrieved from a learnable prompt pool according to the feature at the current position. By incorporating the selected prompt into the output mapping matrix, the ASE supplements the state-space output with semantic information associated with relevant positions beyond the preceding scanned context. This mechanism facilitates information exchange among spatially distant yet semantically similar regions. After state-space modeling, the SGN mechanism restores the features to their original spatial arrangement by applying the inverse of the preceding reordering operation. A subsequent linear projection produces the ASSM output feature. This work adopts the basic ASSM architecture of MambaIRv2 and applies it to the restoration of features extracted from reconstructed frames, thereby enhancing the modeling of cross-region structural correlations and non-local texture correlations. Further implementation details of prompt learning and semantic-guided rearrangement are provided in the original MambaIRv2 paper^[23]. The

effectiveness of the ASSM in long-range structural dependency modeling is further evaluated through ablation experiments in Section 4.4.

3.3. Luma-Guided Chroma Fusion Module

The LGCF module is designed to exploit structural information from the luma component to guide chroma restoration, as illustrated in Figure 4. The module takes the reconstructed luma component Y_{rec} and the reconstructed chroma components U_{rec} and V_{rec} as input.

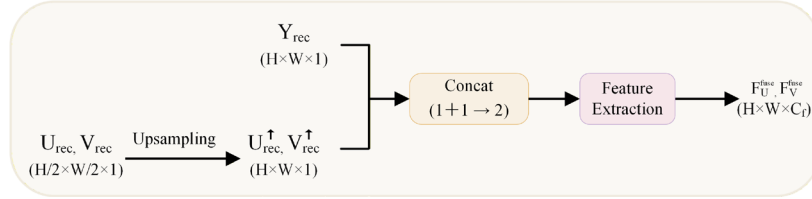


Figure 4: Structure of the Luma-Guided Chroma Fusion Module.

First, U_{rec} and V_{rec} are upsampled to the luma resolution through bilinear interpolation, producing $U_{\text{rec}}^{\uparrow}$ and $V_{\text{rec}}^{\uparrow}$, respectively. The reconstructed luma component Y_{rec} is then separately concatenated with $U_{\text{rec}}^{\uparrow}$ and $V_{\text{rec}}^{\uparrow}$ along the channel dimension, producing two tensors of size $H \times W \times 2$. The concatenated inputs are processed by the shallow feature extraction layers of the U-restoration and V-restoration branches, which map the two input channels to a C_c -dimensional feature space. This operation produces the luma-guided chroma fusion features $F_U^{\text{fuse}}, F_V^{\text{fuse}} \in \mathbb{R}^{H \times W \times C_c}$, where C_c denotes the number of feature channels in the chroma restoration branches and is set to 60. These fused features contain both the intrinsic information of the corresponding chroma component and the structural guidance provided by the luma component. They are subsequently used as inputs to the lightweight chroma restoration backbones. The two chroma restoration branches separately predict chroma compensation residuals at the luma resolution. Because interpolation errors may be introduced during chroma upsampling and downsampling, the predicted residuals are first downsampled to the native chroma resolution and then added to the corresponding input chroma components U_{rec} and V_{rec} through image-level residual connections. This design preserves the original chroma information while allowing the network to focus on learning residual corrections for compression artifacts.

The ablation results show that removing the LGCF module reduces the BD-rate savings for the U and V components from 18.94% and 21.05% to 13.57% and 15.76%, respectively, indicating that structural information from the luma component effectively guides chroma restoration.

4. Experimental Results and Analysis

4.1. Experimental Setup

The DIV2K dataset^[26] was divided into 950 training images and 50 validation images. Under the All-Intra (AI) configuration, all images were independently encoded and decoded as individual frames without relying on reference pictures. The original PNG images were first converted to the YUV 4:2:0 format and then encoded and decoded using VTM-23.13 under the AI configuration. The experiments were conducted at QP values of 22, 27, 32, 37, and 42. During training-data generation, the conventional DBF, SAO, ALF, and CC-ALF tools were disabled. The resulting unfiltered reconstructed frames were used as low-quality (LQ) inputs, while the corresponding original images served as ground-truth (GT) targets, thereby forming LQ/GT training pairs.

A separate QP-specific model was trained for each QP value, resulting in five models with identical network architectures but independently learned parameters. During testing, the model corresponding to the coding QP of each reconstructed frame was selected for filtering. During training, 128×128 patches were randomly cropped from the Y component, while spatially aligned 64×64 patches were synchronously cropped from the corresponding U and V components. The chroma patches remained at their native YUV 4:2:0 resolution during data loading and were upsampled to the luma resolution within the LGCF module. Random horizontal flipping and rotation were employed for data augmentation to improve the model's adaptability to different texture orientations and local structural variations.

The network was implemented using PyTorch 2.0.1 and trained for 10,000 iterations on a single NVIDIA GeForce RTX 3090 GPU. The Adam optimizer was employed with $\beta_1 = 0.9$ and $\beta_2 = 0.99$,

without weight decay. The initial learning rate was set to 2×10^{-4} and adjusted using the CosineAnnealingRestartLR scheduler, with a minimum learning rate of 1×10^{-6} . Gradient accumulation was employed to obtain an effective batch size of 8. The L1 loss was adopted as the reconstruction objective, and the loss-weight ratio for the Y, U, and V components was set to 4:1:1.

During testing, 22 standard sequences from Classes A1 to E were selected according to the Joint Video Experts Team (JVET) Common Test Conditions (CTC). In addition to the standard QP values of 22, 27, 32, and 37, QP 42 was included to cover higher compression levels. Peak Signal-to-Noise Ratio (PSNR) was used to evaluate reconstruction quality, while BD-rate was employed to assess rate-distortion performance. DASR-Net was evaluated using an offline inference procedure. For the VTM-23.13 anchor, DBF, SAO, ALF, and CC-ALF were enabled under the default AI configuration, while neural-network-based filtering was disabled. For the proposed method, these conventional in-loop filtering tools were disabled to obtain unfiltered reconstructed frames, which were subsequently processed by the corresponding QP-specific DASR-Net model. The enhanced outputs were not fed back into the encoder. Under the AI configuration, each picture is independently intra-coded; therefore, the offline evaluation does not affect the coding of other frames. The bitstream rates generated under the corresponding VTM configurations and the PSNR values of the enhanced frames were used for rate-distortion evaluation. This evaluation protocol provides a consistent basis for comparing the rate-distortion performance of the conventional VTM anchor and the proposed neural filtering scheme.

4.2. Objective Performance Comparison

Table 1 presents the per-sequence BD-rate results of the proposed method relative to the VTM-23.13 anchor. Averaged over all test sequences, DASR-Net achieves BD-rate savings of 8.96%, 17.64%, and 19.86% for the Y, U, and V components, respectively, demonstrating improved rate-distortion performance for both the luma and chroma components.

Table 1: Per-sequence BD-rate results of DASR-Net relative to the VTM-23.13 anchor.

Class	Sequence	Y	U	V
Class A1	Campfire	-4.60%	-12.13%	-8.38%
	FoodMarket4	-9.32%	-17.42%	-18.15%
	Tango2	-12.84%	-23.72%	-28.12%
Class A2	CatRobot	-6.00%	-10.07%	-14.41%
	DaylightRoad2	-10.57%	-25.03%	-19.25%
	ParkRunning3	-11.50%	-24.00%	-24.11%
Class B	BQTerrace	-6.33%	-10.77%	-11.10%
	Cactus	-6.98%	-13.55%	-16.35%
	MarketPlace	-9.86%	-28.15%	-27.27%
	RitualDance	-14.52%	-20.41%	-28.78%
	BasketballDrive	-8.83%	-21.82%	-21.74%
Class C	BQMall	-7.79%	-14.19%	-15.85%
	BasketballDrill	-11.78%	-22.16%	-19.76%
	PartyScene	-6.50%	-11.39%	-17.42%
	RaceHorses	-5.26%	-13.37%	-20.37%
Class D	BQSquare	-8.42%	-10.40%	-19.84%
	BasketballPass	-9.71%	-25.46%	-26.80%
	BlowingBubbles	-7.78%	-12.55%	-18.22%
	RaceHorses	-6.99%	-16.51%	-13.97%
Class E	FourPeople	-10.52%	-16.34%	-20.49%
	Johnny	-12.10%	-21.17%	-27.99%
	KristenAndSara	-8.88%	-17.54%	-18.61%
Average		-8.96%	-17.64%	-19.86%

Relatively large gains are observed on Tango2 and RitualDance. For Tango2, the proposed method achieves BD-rate savings of 12.84%, 23.72%, and 28.12% for the Y, U, and V components, respectively. The corresponding savings for RitualDance are 14.52%, 20.41%, and 28.78%. These sequences contain relatively clear human contours, clothing textures, and local edges. Such content provides distinct structural cues and may therefore be more favorable for the long-range structural modeling and content-adaptive local restoration mechanisms of the proposed network. By contrast, relatively smaller gains are observed on Campfire and BQTerrace, for which the Y-component BD-rate savings are 4.60% and 6.33%,

respectively. The relatively limited gains may be related to rapidly varying flames, fine-grained background textures, and complex repetitive structures, which increase the difficulty of compression artifact restoration.

To further examine the performance levels of different learned in-loop filtering methods, the Content-Aware Dynamic In-Loop Filter With Adjustable Complexity (DILF)^[15] was selected for comparison. The DILF results were taken from DILF with the balancing factor set to $\lambda = 1 \times 10^{-5}$. DILF was evaluated using VTM-11.0 under the AI configuration at QP values of 22, 27, 32, 37, and 42, with all conventional in-loop filtering tools enabled for the VTM-11.0 anchor. The proposed method was evaluated using VTM-23.13. Therefore, the following results provide only an indicative comparison of the reported performance levels and do not constitute a strict comparison under identical experimental conditions. Table 2 compares the class-average BD-rate results reported for DASR-Net and DILF. Numerically, DASR-Net achieves an average Y-component BD-rate saving of 8.96%, compared with the 8.08% saving reported for DILF. For the U and V components, DASR-Net obtains average savings of 17.64% and 19.86%, respectively, which are close to the corresponding DILF results of 17.99% and 20.96%. Within the reported results, DASR-Net yields larger Y-component BD-rate savings in Classes A2 and B and larger chroma-component savings in several classes. However, because the two methods were evaluated using different VTM versions and implementation settings, these numerical results should be interpreted only as an indicative comparison. Despite the differences in evaluation settings, the comparison suggests that DASR-Net maintains a comparable overall performance level while achieving relatively larger luma coding gains in several test classes.

Figure 5 shows the Y-component rate–distortion curves of DASR-Net and the VTM-23.13 anchor on representative test sequences. At comparable bitrates, DASR-Net generally achieves higher Y-PSNR, which is consistent with the BD-rate results reported in Table 1.

Table 2: Class-average BD-rate comparison between DILF and DASR-Net.

Class	DILF			DASR-Net		
	Y	U	V	Y	U	V
Class A1	-8.47%	-18.09%	-21.31%	-8.92%	-17.76%	-18.22%
Class A2	-7.03%	-18.94%	-16.11%	-9.36%	-19.70%	-19.26%
Class B	-7.81%	-16.58%	-21.71%	-9.30%	-18.94%	-21.05%
Class C	-7.28%	-18.25%	-21.53%	-7.83%	-15.28%	-18.35%
Class D	-8.02%	-17.63%	-23.08%	-8.23%	-16.23%	-19.71%
Class E	-10.18%	-19.41%	-20.61%	-10.50%	-18.35%	-22.36%
Average	-8.08%	-17.99%	-20.96%	-8.96%	-17.64%	-19.86%

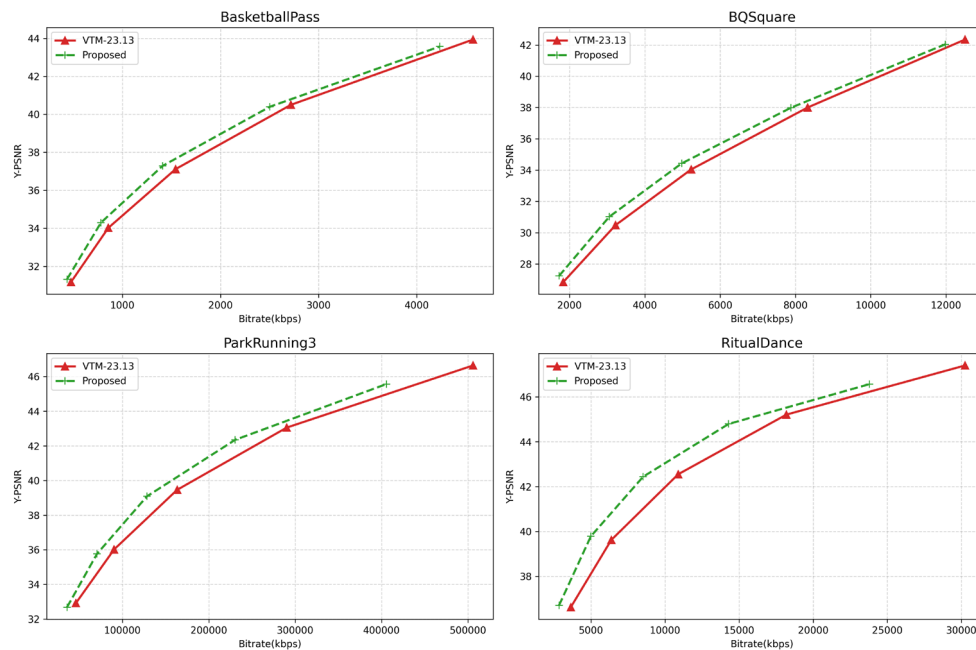


Figure 5: Comparison of Y-component rate–distortion curves between DASR-Net and the VTM-23.13 anchor on representative test sequences at QP values of 22, 27, 32, 37, and 42.

4.3. Visual Quality Comparison

To further evaluate the visual enhancement achieved by the proposed method, representative local regions from the test sequences are enlarged for comparison, as shown in Figure 6. The figure shows the original frames with the selected regions, the corresponding VTM-23.13 reconstructions, the DASR-Net outputs, and the ground-truth regions. The enlarged views provide a more direct comparison of restoration quality around edges, textures, and color-transition regions.

As shown in Figure 6, the anchor reconstructions still exhibit visible compression artifacts in detailed regions, including blurred edges, texture-detail loss, and local blocking artifacts. By comparison, the proposed method more effectively restores local structures such as clothing edges, human contours, and the thin lines of the basketball net, producing enhanced results that are visually closer to the corresponding ground-truth regions. These observations suggest that the proposed network not only suppresses compression artifacts but also preserves and restores structural details during the restoration process.

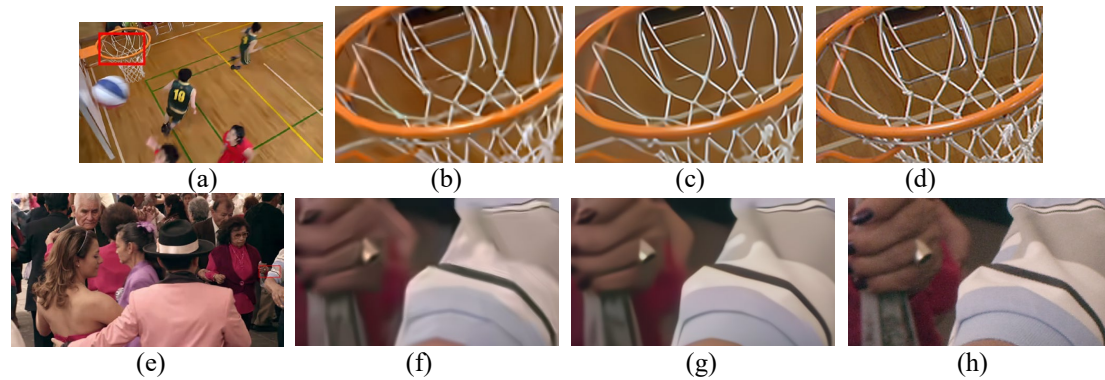


Figure 6: Local visual-quality comparison between DASR-Net and VTM-23.13 on representative test sequences. Panels (a) and (e) show the original frames and the selected enlarged regions; panels (b) and (f) show the VTM-23.13 reconstruction results; panels (c) and (g) show the DASR-Net enhancement results; and panels (d) and (h) show the corresponding ground-truth local regions.

The visual results are consistent with the objective results presented above. The luma restoration branch improves edge and texture details, while the LGCF module alleviates chroma-edge blurring and color bleeding. Overall, the proposed method provides better visual quality than the VTM-23.13 anchor, further demonstrating its effectiveness in compression artifact restoration.

4.4. Ablation Study

To evaluate the effectiveness of the key modules and training strategies, ablation experiments were conducted on the Class B test sequences, as summarized in Table 3.

Table 3: Ablation results for the key modules and training strategies on the Class B test sequences.

	Y	U	V
Proposed	-9.30%	-18.94%	-21.05%
w/o Augmentation	-9.11%	—	—
Patch size = 64×64	-8.29%	—	—
w/o DSM	-6.79%	—	—
w/o ASSM	-4.72%	—	—
w/o LGCF	—	-13.57%	-15.76%

Removing the DSM and ASSM reduces the Y-component BD-rate savings from 9.30% to 6.79% and 4.72%, respectively. These results indicate that both local deformable sampling and long-range structural dependency modeling contribute to luma restoration, with a larger performance degradation observed when the ASSM is removed. Removing the LGCF module reduces the U- and V-component BD-rate savings from 18.94% and 21.05% to 13.57% and 15.76%, respectively, demonstrating the benefit of exploiting luma-derived structural information for chroma restoration. Reducing the luma training patch size from 128×128 to 64×64 decreases the Y-component BD-rate saving from 9.30% to 8.29%, suggesting that a larger training context provides more spatial information for the restoration of reconstructed frames. Removing data augmentation results in only a slight reduction in the Y-component BD-rate saving, from 9.30% to 9.11%, suggesting that data augmentation provides a modest

improvement in model generalization.

Overall, the ablation results verify the complementary contributions of local deformable sampling, long-range structural dependency modeling, and luma-guided chroma fusion to the restoration of reconstructed frames.

5. Conclusions

To address blocking artifacts, ringing artifacts, texture blurring, and chroma distortion in VVC reconstructed frames, this paper proposes a neural network-based in-loop filtering method termed DASR-Net. The proposed method incorporates deformable sampling, attentive state-space modeling, and luma-guided chroma fusion. Specifically, the DSM performs content-adaptive sampling and feature aggregation to model and suppress spatially irregular local artifacts; the ASSM models long-range structural dependencies and non-local texture correlations; and the LGCF module exploits structural information from the luma component to guide chroma restoration. Experimental results show that, under the AI configuration, DASR-Net achieves average BD-rate savings of 8.96%, 17.64%, and 19.86% for the Y, U, and V components, respectively, relative to the VTM-23.13 anchor, while also improving the visual quality of reconstructed frames. The ablation results further demonstrate the effectiveness of the key modules and training strategies. Future work will focus on further reducing model complexity and improving chroma restoration performance, while exploring the integration of DASR-Net into VVC encoders and its deployment in real-time applications.

References

- [1] Bross B, Wang Y K, Ye Y, et al. Overview of the Versatile Video Coding (VVC) Standard and its Applications[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(10): 3736–3764.
- [2] Sullivan G J, Ohm J R, Han W J, et al. Overview of the High Efficiency Video Coding (HEVC) Standard[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12): 1649–1668.
- [3] Karczewicz M, Hu N, Taquet J, et al. VVC In-Loop Filters[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(10): 3907–3925.
- [4] Dong C, Deng Y, Loy C C, et al. Compression Artifacts Reduction by a Deep Convolutional Network[C]. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015: 576–584.
- [5] Jia C, Wang S, Zhang X, et al. Content-Aware Convolutional Neural Network for In-Loop Filtering in High Efficiency Video Coding[J]. *IEEE Transactions on Image Processing*, 2019, 28(7): 3343–3356.
- [6] Song X, Yao J, Zhou L, et al. A Practical Convolutional Neural Network as Loop Filter for Intra Frame[C]. *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018: 1133–1137.
- [7] Ding D, Kong L, Chen G, et al. A Switchable Deep Learning Approach for In-Loop Filtering in Video Coding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020, 30(7): 1871–1887.
- [8] Chen S, Chen Z, Wang Y, et al. In-Loop Filter with Dense Residual Convolutional Neural Network for VVC[C]. *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2020: 149–152.
- [9] Wang M Z, Wan S, Gong H, et al. Attention-Based Dual-Scale CNN In-Loop Filter for Versatile Video Coding[J]. *IEEE Access*, 2019, 7: 145214–145226.
- [10] Huang Z, Guo X, Shang M, et al. An Efficient QP Variable Convolutional Neural Network Based In-loop Filter for Intra Coding[C]. *2021 Data Compression Conference (DCC)*, 2021: 33–42.
- [11] Huang Z, Sun J, Guo X, et al. One-for-All: An Efficient Variable Convolution Neural Network for In-Loop Filter of VVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022, 32(4): 2342–2355.
- [12] Li Y, Zhang L, Zhang K. Convolutional Neural Network Based In-Loop Filter For VVC Intra Coding[C]. *2021 IEEE International Conference on Image Processing (ICIP)*, 2021: 2104–2108.
- [13] Ma D, Zhang F, Bull D R. MFRNet: A New CNN Architecture for Post-Processing and In-loop Filtering[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2021, 15(2): 378–387.
- [14] Zhu L, Zhang Y, Li N, et al. Neural Network Based Multi-Level In-Loop Filtering for Versatile Video Coding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024, 34(11): 12092–12096.
- [15] Man H, Wang H, Lu R, et al. Content-Aware Dynamic In-Loop Filter With Adjustable Complexity for VVC Intra Coding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025, 35(6):

6114–6128.

- [16] Kathariya B, Li Z, Wang H, et al. Multi-Stage Spatial and Frequency Feature Fusion using Transformer in CNN-Based In-Loop Filter for VVC[C]. 2022 Picture Coding Symposium (PCS), 2022: 373–377.
- [17] Kathariya B, Li Z, Geert V der A. Joint Pixel and Frequency Feature Learning and Fusion via Channel-Wise Transformer for High-Efficiency Learned In-Loop Filter in VVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024, 34(5): 4070–4083.
- [18] Zhang H, Liu Y, Jung C, et al. RTNN: A Neural Network-Based In-Loop Filter in VVC Using Resblock and Transformer[J]. *IEEE Access*, 2024, 12: 104599–104610.
- [19] Liang J, Cao J, Sun G, et al. SwinIR: Image Restoration Using Swin Transformer[C]. 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021: 1833–1844.
- [20] Zamir S W, Arora A, Khan S, et al. Restormer: Efficient Transformer for High-Resolution Image Restoration[C]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022: 5718–5729.
- [21] Gu A, Dao T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces[C]. *First Conference on Language Modeling (COLM)*, 2023.
- [22] Guo H, Li J, Dai T, et al. MambaIR: A Simple Baseline for Image Restoration with State-Space Model[C]. *Computer Vision – ECCV 2024*, 2025: 222–241.
- [23] Guo H, Guo Y, Zha Y, et al. MambaIRv2: Attentive State Space Restoration[C]. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025: 28124–28133.
- [24] Dai J, Qi H, Xiong Y, et al. Deformable Convolutional Networks[C]. 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 764–773.
- [25] Xia Z, Pan X, Song S, et al. Vision Transformer with Deformable Attention[C]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022: 4784–4793.
- [26] Agustsson E, Timofte R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017: 1122–1131.