Plug-and-Play Versatile Compressed Video Enhancement

Huimin Zeng Jiacheng Li Zhiwei Xiong*
University of Science and Technology of China
{zenghuimin, jclee}@mail.ustc.edu.cn zwxiong@ustc.edu.cn

Abstract

As a widely adopted technique in data transmission, video compression effectively reduces the size of files, making it possible for real-time cloud computing. However, it comes at the cost of visual quality, posing challenges to the robustness of downstream vision models. In this work, we present a versatile codec-aware enhancement framework that reuses codec information to adaptively enhance videos under different compression settings, assisting various downstream vision tasks without introducing computation bottleneck. Specifically, the proposed codec-aware framework consists of a compression-aware adaptation (CAA) network that employs a hierarchical adaptation mechanism to estimate parameters of the frame-wise enhancement network, namely the bitstream-aware enhancement (BAE) network. The BAE network further leverages temporal and spatial priors embedded in the bitstream to effectively improve the quality of compressed input frames. Extensive experimental results demonstrate the superior quality enhancement performance of our framework over existing enhancement methods, as well as its versatility in assisting multiple downstream tasks on compressed videos as a plug-and-play module. Code and models are available at https://huimin-zeng.github.io/PnP-VCVE/.

1. Introduction

With the flower booming of short video platforms, video has become one of the most popular multimedia formats. In addition to distributing visual content, in practical scenarios (*e.g.*, autonomous driving [43, 48]), it is common to upload the captured videos to the cloud end for further visual analysis and downstream applications (*e.g.*, object detection [5, 51] and segmentation [20, 44]). However, due to the bandwidth constraint during transmission, these videos are typically compressed with varying levels, resulting in poor visual quality and suboptimal performance in downstream tasks [22, 59] (*e.g.*, inaccurate segmentation boundaries in Fig. 1). Given the crucial role of videos in data transmission, there is a critical need for a versatile solution to enhance videos of diverse compression levels and effectively support various downstream tasks.

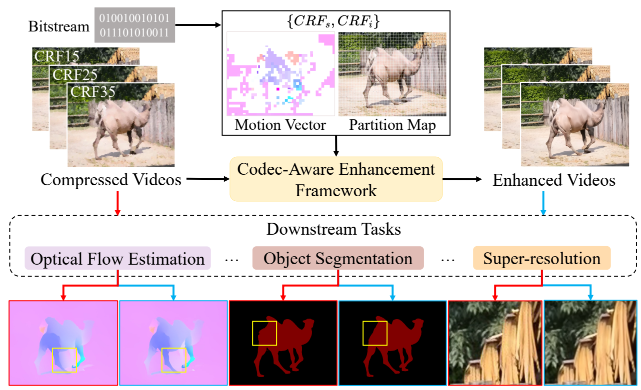


Figure 1. The proposed codec-aware enhancement framework reuses codec information to adaptively enhance videos across different compression settings, while assisting in various downstream tasks in a plug-and-play manner.

Existing video enhancement methods are hard to respond to these demands. Specifically, to effectively enhance videos of different compression levels, previous methods [11, 12, 16, 22, 32, 55] employ separate enhancement models for each compression level, which is inflexible occurring unseen compression levels. Recent approaches [8, 18, 47] consider this issue as the generalization ability across diverse compression levels, therefore randomly selecting inputs of different compression levels during training. However, such a training strategy is compression-agnostic and offers limited improvement. Most importantly, the aforementioned methods focus primarily on improving perceptual quality, neglecting the need to assist in downstream tasks in real-world scenarios.

Based on the mismatch between the versatility demand and existing solutions, here we summarize the following criteria of a favorable solution: 1) adaptively enhance videos of varying compression levels with a single model; 2) effectively assist various downstream tasks on compressed videos in a plug-and-play manner; and 3) given the

*Corresponding author.

practical scenarios where real-time processing is required, it should meet the above objectives without causing a computation bottleneck. To achieve this, we introduce a codec-aware enhancement framework (as shown in Fig. 1) that reuses codec information embedded in the bitstream. By incorporating compression factors, the framework dynamically adjusts its parameters to flexibly enhance inputs of different compression levels. By reusing motion vectors and partition maps, it efficiently aggregates temporal and spatial clues without introducing redundant computations.

Specifically, our codec-aware enhancement framework comprises a compression-aware adaptation (CAA) network and a bitstream-aware enhancement (BAE) network. The CAA network serves as a "meta" network that dynamically adjusts the parameters of the subsequent BAE network. A hierarchical adaptation mechanism is proposed to first estimate sequence-adaptive parameters based on the sequence compression level, and then re-weight these parameters according to the frame compression level, thereby achieving a BAE network tailored for each input frame. The frame-adaptive BAE network conducts motion vector alignment to aggregate intra-frame information and provide useful clues for the current frame. Subsequently, based on the region complexity indicated by partition maps, the region-aware refinement assigns independent filters for different regions, achieving flexible enhancement for different regions. Comprehensive experiments are conducted to demonstrate the superiority of our method in improving the quality of compressed videos, and the effectiveness of assisting in various downstream tasks (*i.e.*, video super-resolution, optical flow estimation, and video object segmentation). Our contributions are summarized as follows:

- We present a codec-aware framework for versatile compressed video enhancement, which adaptively enhances input videos of different compression levels and supports a wide range of downstream vision tasks.
- We develop a compression-aware adaptation (CAA) network and a bitstream-aware enhancement (BAE) network that utilize the off-the-shelf codec information, contributing to generalizing across different compression settings and boosting the enhancement performance with a unified framework.
- Experimental results show the superiority of our method over existing enhancement methods, and its effectiveness in serving as a plug-and-play enhancement module to assist in downstream tasks.

2. Related Work

2.1. Compressed Video Enhancement

Existing compressed video enhancement methods can be categorized into in-loop and post-processing methods. Although in-loop methods [15, 24, 26, 40] effectively improve the quality of reconstructed frames, they embed filters in the encoding and decoding loops, therefore not suitable for enhancing already compressed videos. While the post-processing methods [17, 25, 27, 36, 42, 47, 50, 56, 58, 61, 63, 64] provide more practical solutions to enhance compressed videos by placing filters at the decoder side. Observing the quality fluctuation across frames, MFQE [55] locates the peak quality frame (PQF) with an SVM-based detector and proposes a multi-frame quality enhancement mechanism to enhance non-PQFs. MFQE 2.0 [22] further designs a BiLSTM-based detector and performs multi-frame quality enhancement for both non-PQF and PQF. To address the inaccuracies in optical flow estimation from compressed videos, STDF [16] proposes estimating the offset field using spatio-temporal deformable convolution. S2SVR [32] introduces a sequence-to-sequence network to model long-range dependencies within frames. The aforementioned methods inflexibly equip a separate model for each compression level, while we propose to adaptively handle diverse inputs with a single unified model. Recent methods [18, 64] utilize spatial priors from bitstream to address multiple compression levels, however, they only consider I/P-frames, whereas we design a hierarchical adaptation mechanism to address all types of frames.

2.2. Codec-Aware Video Super-Resolution

Some works in video super-resolution (VSR) [11, 12, 28, 29, 52, 60, 62] explore ways of using codec information such as motion estimation and spatial prior for reconstruction. COMISR [29] focuses on reducing accumulated warping errors caused by the random locations of the intra-frame from compressed video frames. Chen *et al.* [11] employ motion vectors to build the temporal relationship and suppress coding artifacts. CVCP [12] utilizes motion vectors and spatial priors with a guided soft alignment scheme and guided SFT layer, respectively. CIAF [60] leverages motion vectors and residuals to model temporal relationships and skip redundant computations, respectively. Despite leveraging codec information, these methods focus on a single task (*i.e.*, VSR). In contrast, our method not only shows competitive performance in VSR (see supplementary materials), but also effectively supports a range of downstream tasks, which is not explored by the aforementioned works.

2.3. Dynamic Neural Networks

Instead of setting separate models for different inputs, the early mixture of expert (MoE) structure [2, 19, 37] constructs parallel network branches and selectively executes branches to obtain the weighted outputs. Instead of increasing the number of parallel branches, the dynamic parameters ensemble strategy [2, 13, 19, 23, 37, 54] presets parallel expert layers and selectively fusing their parameters to promote the network capability and generalization ability, therefore serving as an efficient alternative to MoE. To promote the generalization ability of pre-trained models, Gain-

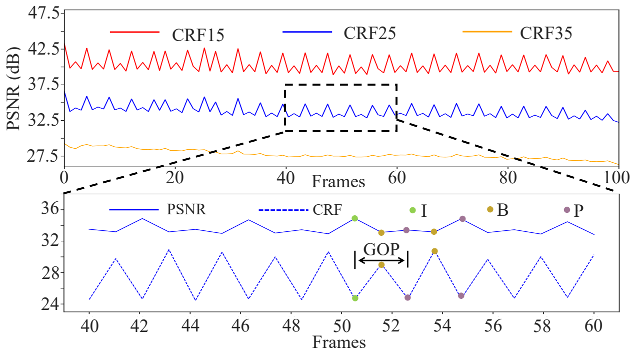


Figure 2. Hierarchical structure of quality adjustment, where frames are divided into multiple groups of pictures (GOP). The Constant Rate Factor (CRF) affects video quality at both sequence and frame levels. An increase in the CRF value indicates a reduction in video quality (*e.g.*, lower PSNR values).

tune [38] proposes to predict a single multiplicative scaling parameter for each channel according to test samples, thus modifying static models to test-adaptive ones. Li *et al.* [30] handle the conflicts between the domain-agnostic model and multiple target domains with dynamic transfer, which is simply modeled by combining residual matrices and a static convolution matrix. DRConv [10] divides the input image into different regions with a learnable mask and assigns multiple filters for these regions, which enhances the feature representation ability without introducing a noticeable computation burden. Instead of searching and learning conditions for dynamic parameters, we leverage priors embedded in the bitstream as the condition.

3. Preliminaries

We take H.264 [53] as a representative standard to analyze available codec information. Note recent codecs [4, 46] also provide similar priors (*i.e.*, CRF, motion vector, partition map), therefore assuring the applicability of our method. To reduce transmission bandwidth, codecs compress videos by adjusting quality and reducing redundancy.

3.1. Hierarchical Quality Adjustment

Video quality is commonly influenced by the constant rate factor (CRF), which involves hierarchical adjustment for both sequence-wise and frame-wise compression.

Sequence-wise CRF. The CRF ranges from 0 to 51 to balance compression efficiency and visual quality. A higher CRF results in more compact output but increased pixel loss (*e.g.*, the average PSNR of CRF35 is much lower than CRF15 in Fig. 2). By considering the sequence-wise CRF (denoted as CRF_s), the enhancement network can be tailored to handle videos of different compression levels.

Frame-wise CRF As shown in Fig. 2, a video sequence is divided into multiple groups of pictures (GOP) and further categorized as I-frames, P-frames, or B-frames. The CRF value of each frame (denoted as CRF_i) is dynamically adjusted based on CRF_s so that lower CRF_i is assigned for I/P frames to maintain quality and higher CRF_i for B frames for compact representations.

Inspired by the hierarchical quality adjustment paradigm, we design a hierarchical adaptation paradigm that first performs sequence adaptation to predict network parameters based on CRF_s, and then re-weights these parameters according to CRF_i for frame adaptation. In practical scenarios where CRF_i is unavailable (*e.g.*, limited access to full bitstream), the proposed method can instead use slice type (I/P/B) for frame adaptation, which is demonstrated to yield similar performance in Sec. 5.2.

3.2. Redundancy Reduction

To reduce redundancy and improve the entropy of the bitstream, codecs block-wisely perform motion estimation to model the intra-frame correlations and embed the correlations in the bitstream for decoding.

Partition map. As shown in Fig. 1, different regions of each frame are partitioned into blocks of varying sizes (*e.g.*, H.264 provides macroblocks of 16×16, 16×8, 8×16, and 8×8) according to the texture complexity. Flat regions (*e.g.*, the ground) can tolerate higher quantization errors and are therefore divided into blocks of large size, while complex regions (*e.g.*, leaves and fence) take smaller blocks to maintain details. To effectively enhance regions of different complexity, we propose dynamically assigning filters based on the partition map that indicates region complexity.

Motion vector. Motion vectors are utilized in decoding to aggregate information from reference frames and propagate information of current frame. As illustrated in Fig. 1, they describe the relationship between current frame and its reference frames in a block-wise manner. Although motion vectors can be noisy and are less precise than optical flow, they effectively align reference frames with current frame, serving as a cost-effective alternative for optical flow.

4. Codec-Aware Enhancement Framework

4.1. Overview

As shown in Fig. 3(a), the proposed method comprises a compression-aware adaption (CAA) network $\mathcal{G}_\phi$ and a bitstream-aware enhancement (BAE) network $\mathcal{F}_{\theta_i}$. The CAA network employs a hierarchical compression adaptation mechanism to estimate parameters for the frame-adaptive BAE network, which then aggregates intra-frame information and performs region-aware refinement to enhance the input compressed frame.

4.2. Compression-Aware Adaptation Network

To handle sequences of varying compression levels and quality fluctuations across frames, as illustrated in Fig. 3(a), the CAA network $\mathcal{G}_\phi$ utilizes CRF_s for sequence adaptation to estimate sequence-adaptive parameters, and perform frame adaptation to refine these parameters based on CRF_i.

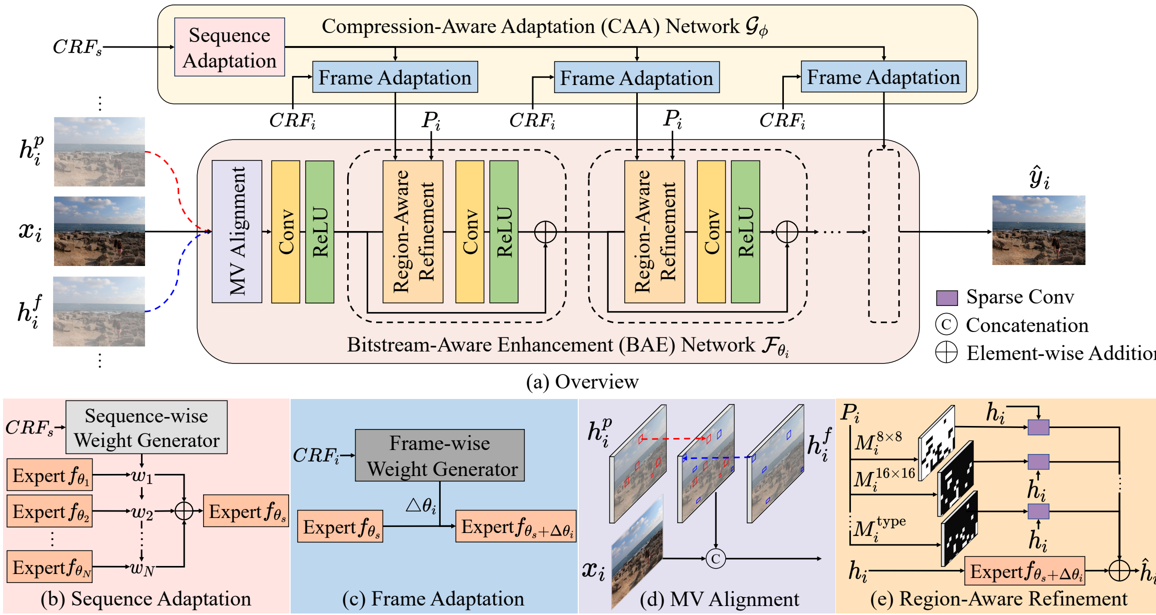


Figure 3. The proposed Codec-Aware Enhancement Framework consists of two sub-networks: 1) the Compression-Aware Adaptation (CAA) Network, which hierarchically applies sequence adaptation and frame adaptation to dynamically adjust parameters of the enhancement network; and 2) Bitstream-Aware Enhancement (BAE) Network, which leverages motion vectors to align frames and conducts region-aware refinement to flexibly enhance regions of different complexity.

Sequence adaptation. To ensure robust performance across multiple compression settings without increasing complexity, we propose estimating sequence-adaptive parameters for the enhancement network instead of fusing features from separate submodels. As shown in Fig. 3(b), parallel expert layers $\{f_{\theta_1}, f_{\theta_2}, ..., f_{\theta_N}\}$, which share the same architecture but have independent parameters, serve as the basis for parameter combination. The sequence-wise CRF_s is adopted as the condition to re-weight parameters of these expert layers, which is expressed as follows,

$$f_{\theta_s} = \mathcal{G}_{\phi_s}(CRF_s, \{f_{\theta_1}, f_{\theta_2}, ..., f_{\theta_N}\}) = \sum_{n=1}^{N} w_n f_{\theta_n}, \quad (1)$$

where f_{θ_s} and $\mathcal{G}_{\phi_s}$ denote the sequence-adaptive expert layer and the sequence-wise weight generator, respectively. w_n denotes the weight for each expert layer. We set $N = 6$ (see ablation studies in supplementary materials) and visualize w_n against different CRF_s in Fig. 4, which shows that each expert layer has a distinct preference for specific CRF_s. Compared to MoE that re-weights output features, re-weighting expert layer parameters (as shown in Eq. 1) is computationally efficient and comparable to the network constructed with a single expert layer. Note that CRF_s is constant for frames within the same sequence, f_{θ_s} is predicted only once and reused by subsequent frames.

Frame adaptation. To flexibly enhance frames with different visual quality, we propose to re-weight the sequence-based f_{θ_s} using frame-wise CRF_i. We attribute the quality fluctuation between the sequence and current frame to the disparity between CRF_s and CRF_i, which can be addressed by introducing a set of frame-wise auxiliary parameters $\triangle\theta_i$. As shown in Fig. 3(c), the auxiliary parameters $\triangle\theta_i$ that conditioned on CRF_i re-weights the sequence-adaptive f_{θ_s} to obtain the frame-adaptive expert layer f_{θ_i}, which is expressed as follows,

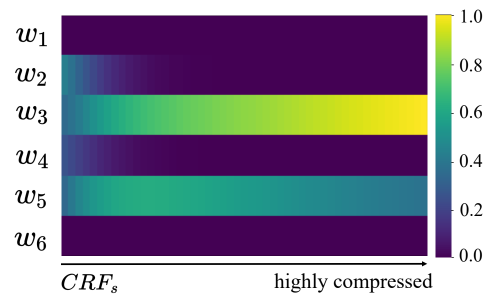


Figure 4. Visualization of w_n against different CRF_s, where each expert shows a distinct preference for specific CRF_s values.

$$f_{\theta_i} = \mathcal{G}_{\phi_i}(CRF_i, f_{\theta_s}) = f_{\theta_s + \triangle\theta_i}, \quad (2)$$

where f_{θ_i} and $\mathcal{G}_{\phi_i}$ denote the estimated frame-adaptive expert layer and the frame-wise parameters generator, respectively. As shown by the black dashed lines in Fig. 3(a), the obtained f_{θ_i} is used to construct the enhancement blocks, resulting in the frame-adaptive BAE network $\mathcal{F}_{\theta_i}$ (introduced in the following Sec. 4.3).

4.3. Bitstream-Aware Enhancement Network

To leverage high-quality frames and propagate information, the BAE network $\mathcal{F}_{\theta_i}$ utilizes motion vectors to align reference frames with the current frame. Meanwhile, the par-

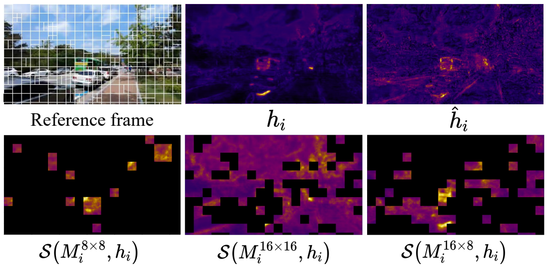


Figure 5. Visualization of features in region-aware refinement, where h_i and $\hat{h}_i$ indicate the input and output features, respectively. The refined features are denoted in the format of $\mathcal{S}(M_i^{type}, h_i)$.

tition map serves as spatial complexity guidance to enable flexible enhancement of different regions.

Motion vector alignment. Since the motion vectors roughly model the temporal relationship in a block-wise manner, for each block of the current frame shown in Fig. 3(d), motion vectors locate blocks with similar content in the previous and future reference features (highlighted with red and blue boxes). The warped reference features are concatenated with current frame along the channel dimension as input of the BAE network, expressed as follows,

$$\hat{x}_i = [MV(h_i^p), MV(h_i^f), x_i], \tag{3}$$

where $\hat{x}_i$, h_i^p and h_i^f denote the current input frame, enhanced features of previous and future reference frames, respectively. MV denotes warping reference features based on motion vectors. $[,]$ denotes concatenation along channel dimension. Bilinear interpolation is adopted for the case that the offsets of motion vectors are not integers.

Region-aware refinement. To effectively enhance regions of different complexity, we propose to dynamically assign different filters for regions based on the partition map. As shown in Fig. 3(e), the block-based partition map P_i is decoupled into multiple binary masks $\{M_i^{8\times8}, M_i^{16\times16}, ..., M_i^{type}\}$ according to the size of macroblocks, allowing separate refinement of regions using sparse convolution [33]. The output is defined as the sum of frame-adaptive extracted features and separately region-aware refined features, depicted as follows,

$$\hat{h_i} = \mathcal{F}_{\theta_i}(\hat{x}_i, P_i) = f_{\theta_i} * h_i + \sum_{type=1}^{M} \mathcal{S}(M_i^{type}, h_i), \tag{4}$$

where $\hat{h_i}$ indicates the output features. $\mathcal{S}$ denotes the operations applying sparse convolution guided by mask M_i^{type} to refine input features. In H.264 standard, three types of macroblocks are used (16×16, $8\times16/16\times8$, and 8×8), thus M is set to 3. Features in region-aware refinement are visualized in Fig 5, where the refined features are denoted with operations like $\mathcal{S}(M_i^{8\times8}, h_i)$. As can be seen, the output features $\hat{h}_i$ contain more fine-grained and high-frequency details than h_i. Meanwhile, refined features provide distinct activations for different regions. For instance, $\mathcal{S}(M_i^{8\times8}, h_i)$ focuses on static objects (*e.g.*, trees) while $\mathcal{S}(M_i^{8\times16}, h_i)$ focuses on moving objects (*e.g.*, the bus).

4.4. Loss function

We adopt Charbonnier penalty loss [9] as the loss function and train the proposed codec-aware enhancement framework in an end-to-end manner. The specific loss function is expressed as follows,

$$\mathcal{L} = \frac{1}{T}\sum_{i=1}^{T}\sqrt{\|y_i - \hat{y}_i\|^2 + \epsilon^2}, \tag{5}$$

where y_i, $\hat{y}_i$ and T indicate the uncompressed ground truth, the predicted output, and the length of the input sequence. ϵ is set to 1×10^{-12}.

5. Experiments

5.1. Experimental Settings

Compression settings. H.264 is a popular video compression standard that compresses nearly 85% of internet videos [1], and tends to introduce more severe degradations than H.265 and H.266. We adopt H.264 [45] and compress videos with the CRF_s values of 15, 25, and 35.

Tasks and training dataset. Our tasks involve quality enhancement and assisting downstream tasks on compressed inputs. The primary downstream tasks include video super-resolution, optical flow estimation and video object segmentation, with video inpainting reported as an extension to fully evaluate the versatility. Training splits of REDS [39] and DAVIS [41] datasets are combined for training.

Compared methods. We compare with representative methods in compressed video enhancement, including MFQE 2.0 [22], STDF [16], S2SVR [32] and Metabit [18]. For a fair comparison, we fully retrain these methods with the same training dataset and configurations. For downstream tasks, the compressed video is first enhanced by quality enhancement methods, and then fed to downstream models for corresponding tasks and further assessment.

5.2. Results

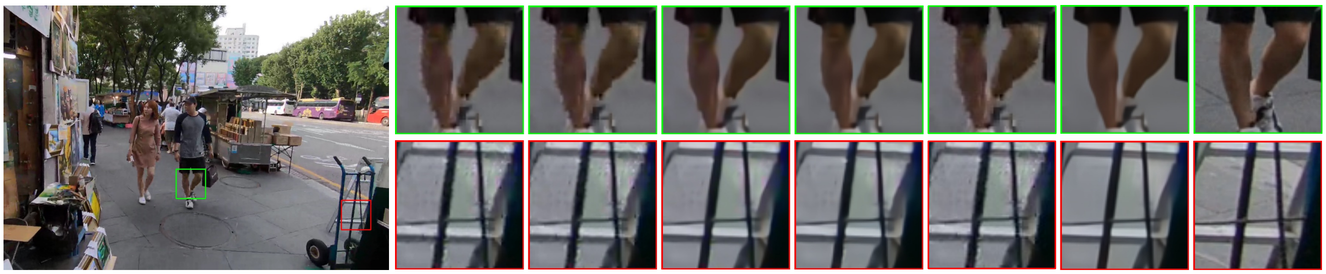
Our evaluations are two-fold: 1) verifying the quality enhancement performance on seen, unseen, and highly compressed scenarios; 2) evaluating the versatility to assist different downstream tasks on multiple compression settings.

5.2.1. Quality Enhancement Performance

Quantitative results. The results of compressed video quality enhancement are reported in Tab.1, which is evaluated on the REDS4 datasets [39] using PSNR and SSIM (the higher the better). Note the CRF values of 18, 28 and 38 are not included during training. For each method, we include the model complexity and inference speed. For our method, we report results of both applying CRF_i and its substitution with slice type (highlighted with grey). As shown in Tab. 1, leveraging slice type yields comparable performance

Method	Param/M	FLOPs/G	Speed/ms	FPS	CRF15	CRF25	CRF35	CRF18	CRF28	CRF38
					PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑
Input	-	-	-	-	41.04 / 0.9785	34.92 / 0.9363	29.25 / 0.8238	39.12 / 0.9698	33.18 / 0.9123	27.69 / 0.7725
MFQE 2.0 [22]	1.64	51	53	19	40.95 / 0.9806	34.83 / 0.9378	29.22 / 0.8256	38.97 / 0.9712	33.13 / 0.9140	27.67 / 0.7742
STDF [16]	1.27	45	38	26	41.15 / 0.9793	35.23 / 0.9398	29.74 / 0.8359	39.28 / 0.9712	33.58 / 0.9178	28.11 / 0.7853
S2SVR [32]	7.43	294	-	-	41.96 / 0.9834	35.61 / 0.9445	29.87 / 0.8391	39.88 / 0.9755	33.87 / 0.9223	28.19 / 0.7881
Metabit [18]	1.60	92	24	42	41.04 / 0.9785	34.92 / 0.9363	29.25 / 0.8238	39.11 / 0.9698	33.18 / 0.9123	27.69 / 0.7725
Ours	4.56	47	36	28	42.22 / 0.9842	35.90 / 0.9468	30.17 / 0.8471	40.17 / 0.9767	34.16 / 0.9258	28.49 / 0.7985
					42.24 / 0.9842	**35.91 / 0.9468**	**30.19 / 0.8472**	**40.18 / 0.9767**	**34.17 / 0.9258**	**28.52 / 0.7985**

Table 1. Quantitative results on quality enhancement, where PSNR and SSIM (higher is better) are adopted for evaluation. The best and second best results are marked with **bold** and underline. Results obtained by replacing CRF_i with slice type are highlighted with grey.



Frame 030, Clip 020, CRF35 | Input | MFQE 2.0 | STDF | S2SVR | Metabit | Ours | GT

Figure 6. Qualitative results on quality enhancement, where our method effectively reduces the compression artifacts, achieving visually pleasant results. In contrast, the results of the compared methods still contain severe distortions (*e.g.*, the calf in the 1st row).

Method	CRF15	CRF25	CRF35
	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑
BasicVSR	29.24 / 0.8212	26.19 / 0.7131	23.40 / 0.6005
+ MFQE 2.0	29.29 / 0.8233	26.28 / 0.7182	23.46 / 0.6056
+ STDF	29.31 / 0.8247	26.51 / 0.7293	23.80 / 0.6249
+ S2SVR	29.45 / 0.8288	26.70 / 0.7346	23.86 / 0.6270
+ Metabit	29.24 / 0.8211	26.19 / 0.7131	23.39 / 0.6005
+ Ours	**29.54 / 0.8328**	**26.85 / 0.7419**	**24.02 / 0.6361**
IconVSR	29.29 / 0.8230	26.19 / 0.7130	23.39 / 0.6003
+ MFQE 2.0	29.37 / 0.8254	26.28 / 0.7182	23.45 / 0.6055
+ STDF	29.36 / 0.8263	26.52 / 0.7292	23.79 / 0.6248
+ S2SVR	29.54 / 0.8306	26.71 / 0.7345	23.85 / 0.6269
+ Metabit	29.29 / 0.8230	26.19 / 0.7130	23.39 / 0.6003
+ Ours	**29.63 / 0.8344**	**26.86 / 0.7418**	**24.01 / 0.6360**
BasicVSR++	29.61 / 0.8303	26.19 / 0.7118	23.38 / 0.5998
+ MFQE 2.0	29.66 / 0.8322	26.27 / 0.7169	23.44 / 0.6051
+ STDF	29.68 / 0.8338	26.53 / 0.7289	23.79 / 0.6247
+ S2SVR	29.82 / 0.8371	26.72 / 0.7346	23.85 / 0.6269
+ Metabit	29.61 / 0.8303	26.19 / 0.7118	23.38 / 0.5997
+ Ours	**29.92 / 0.8407**	**26.87 / 0.7419**	**24.00 / 0.6358**

Table 2. Quantitative results of $\times 4$ VSR, where the best and second best results are highlighted with **bold** and underline.

to CRF_i, with a negligible decrease of PSNR (<0.03 dB), demonstrating the feasibility of replacing CRF_i with slice type in practical. The proposed method notably improves the quality of compressed input, achieving a PSNR gain of 1.2 dB on CRF15, while MFQE 2.0 and Metabit lead to no improvement. With similar computation cost and inference speed, our method significantly outperforms STDF, obtaining a PSNR gain of 1.09 dB on CRF15. Compared to S2SVR, our approach takes only 61% of the parameters and 16% of the FLOPs, and achieves a throughput of 28 FPS, which underlines its practicality. In addition, our method shows robustness and generalization ability on unseen scenarios (*i.e.*, CRF18, CRF28 and CRF38), achieving up to 1.06 dB PSNR gain on CRF18. In contrast, other methods trained with mixed compression settings show sub-optimal performance. For instance, STDF and S2SVR only achieve PSNR gains of 0.16 dB and 0.76 dB at CRF18, while MFQE 2.0 shows no improvement. Quantitative results on highly compressed scenarios (*i.e.*, CRF40, CRF45, CRF48) are included in the supplementary materials.

Qualitative results. Qualitative comparisons are provided in Fig. 6. As can be seen, MFQE 2.0 and Metabit struggle to improve the quality of compressed inputs. Both STDF and S2SVR cannot adequately remove compression artifacts (*e.g.*, boundary of the calf), while the proposed method effectively eliminates the compression artifacts, preserving accurate edges and textures. We provide more qualitative comparisons in the supplementary materials.

5.2.2. Versatility Evaluation

To evaluate the versatility in assisting practical downstream tasks, we employ the implementation that utilizes slice type for frame adaptation (as described in Sec. 3.1) to enhance compressed inputs for downstream tasks. More qualitative comparisons are included in the supplementary materials.

Video super-resolution. We adopt BasicVSR [6], IconVSR [6], and BasicVSR++ [7] as the representative baseline methods for $\times 4$ video super-resolution (VSR), which are trained on "clean" data without considering compresstion. The evaluation is conducted on the REDS4 dataset [39] and summarized with PSNR and SSIM (the higher the better). As depicted in Tab. 2, pre-enhancing compressed inputs with Metabit fails to improve the performance of downstream VSR models. In contrast, pre-enhancing with our

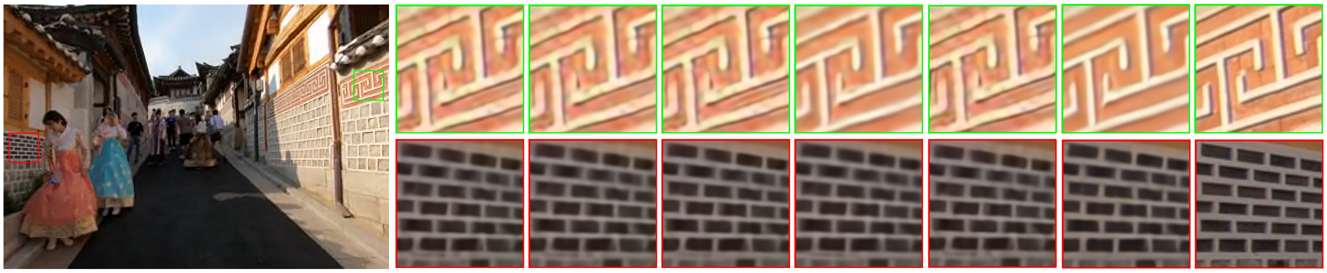


Figure 7. Qualitative results of ×4 VSR. Pre-enhancing with our method before VSR effectively avoids amplifying compression artifacts. While other methods cannot fully eliminate the artifacts and even severe the distortions (*e.g.*, results of MFQE 2.0).

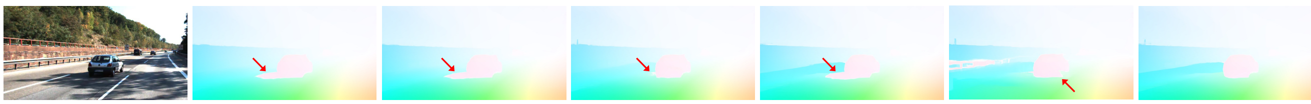


Figure 8. Qualitative results of optical flow estimation. As can be seen, pre-processing with our method effectively corrects the mispredicted optical flow, especially the boundaries of moving objects (*e.g.*, edge of the moving car).

Method	CRF15	CRF25	CRF35
	EPE↓ / F1-all↓	EPE↓ / F1-all↓	EPE↓ / F1-all↓
RAFT	5.26 / 17.81	7.37 / 22.13	16.73 / 44.70
+ MFQE 2.0	5.32 / 17.83	7.27 / 21.92	16.68 / 44.74
+ STDF	5.34 / 17.93	7.13 / 22.16	15.92 / 44.04
+ S2SVR	5.22 / 17.71	6.90 / 21.57	15.73 / 44.62
+ Metabit	5.32 / 17.86	7.33 / 22.07	16.69 / 44.28
+ Ours	**5.20 / 17.69**	**6.52 / 20.99**	**14.84 / 42.56**
DEQ	3.99 / 13.71	5.40 / 17.33	13.94 / 41.63
+ MFQE 2.0	3.97 / 13.73	5.14 / 17.06	14.06 / 41.72
+ STDF	4.08 / 13.84	5.27 / 17.38	13.52 / 40.75
+ S2SVR	4.01 / 13.69	5.22 / 16.99	13.74 / 41.65
+ Metabit	**3.92** / 13.78	5.30 / 17.33	13.84 / 41.21
+ Ours	3.97 / **13.68**	**4.96 / 16.54**	**13.09 / 39.43**
KPAFlow	4.46 / 16.07	6.71 / 20.96	16.50 / 45.13
+ MFQE 2.0	4.42 / 15.96	6.71 / 20.92	16.70 / 45.29
+ STDF	4.52 / 16.19	6.96 / 21.53	15.62 / 44.17
+ S2SVR	**4.37 / 15.96**	6.10 / 19.76	15.62 / 44.23
+ Metabit	4.47 / 16.17	6.75 / 21.08	16.68 / 44.90
+ Ours	4.43 / 16.10	**5.59 / 18.73**	**14.83 / 41.24**

Table 3. Quantitative results of optical flow estimation, where we highlight the best and second best results with **bold** and underline.

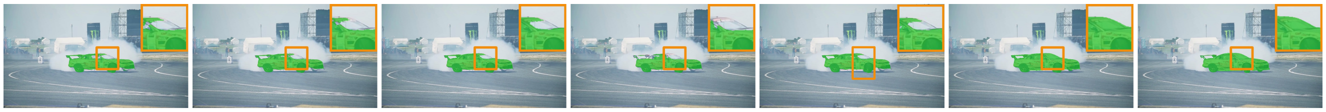
Method	CRF15	CRF25	CRF35
	Avg↑ / $\mathcal{J}$↑ / $\mathcal{F}$↑	Avg↑ / $\mathcal{J}$↑ / $\mathcal{F}$↑	Avg↑ / $\mathcal{J}$↑ / $\mathcal{F}$↑
STCN	85.07 / 81.83 / 88.32	84.35 / 80.96 / 87.74	79.20 / 76.04 / 82.37
+ MFQE 2.0	84.96 / 81.71 / 88.21	84.30 / 80.92 / 87.69	79.28 / 76.11 / 82.44
+ STDF	85.01 / 81.73 / 88.28	84.23 / 80.97 / 87.50	79.77 / 76.52 / 83.02
+ S2SVR	85.17 / 81.93 / 88.41	84.46 / 81.20 / 87.72	80.04 / 76.88 / 83.20
+ Metabit	84.56 / 80.97 / 88.14	83.86 / 80.18 / 87.55	79.03 / 75.65 / 82.40
+ Ours	**85.21 / 81.99 / 88.44**	**84.63 / 81.42 / 87.85**	**81.57 / 78.46 / 84.69**
DeAoT	85.90 / 82.89 / 88.91	85.18 / 82.37 / 88.00	82.87 / 79.86 / 85.88
+ MFQE 2.0	85.86 / 82.84 / 88.88	85.20 / 82.38 / 88.03	82.86 / 79.86 / 85.85
+ STDF	85.83 / 82.80 / 88.87	85.18 / 82.27 / 88.09	**82.90 / 79.92** / 85.89
+ S2SVR	86.05 / 83.09 / 89.01	85.05 / 82.07 / 88.04	82.64 / 79.63 / 85.65
+ Metabit	85.47 / 82.04 / 88.90	84.95 / 81.57 / **88.33**	82.32 / 79.04 / 85.59
+ Ours	**86.08 / 83.13 / 89.03**	**85.31 / 82.38** / 88.25	82.88 / 79.83 / **85.92**
QDMN	85.16 / 82.20 / 88.11	84.16 / 81.20 / 87.12	79.39 / 76.61 / 82.18
+ MFQE 2.0	85.13 / 82.20 / 88.06	84.15 / 81.18 / 87.13	79.51 / 76.75 / 82.27
+ STDF	85.32 / 82.38 / 88.27	83.36 / 80.44 / 86.27	79.64 / 76.69 / 82.59
+ S2SVR	85.28 / 82.32 / 88.23	83.64 / 80.65 / 86.63	79.02 / 76.15 / 81.89
+ Metabit	84.50 / 81.14 / 87.87	83.68 / 80.30 / 87.06	79.47 / 76.41 / 82.52
+ Ours	**85.34 / 82.41 / 88.27**	**84.37 / 81.42 / 87.32**	**79.78 / 76.92 / 82.65**

Table 4. Quantitative results of VOS, where the best and second best results are highlighted with **bold** and underline.

framework yields consistent improvement especially in scenarios of high compression (*e.g.*, up to 0.62 dB PSNR gain with BaiscVSR++ on CRF35). Meanwhile, our method significantly outperforms MFQE 2.0 and STDF, achieving PSNR gains of 0.25 dB and 0.23 dB over MFQE 2.0 and STDF on BasicVSR/CRF15, respectively. Compared with S2SVR, the proposed method offers more effective support to VSR models with lower complexity. As shown in Fig. 7, performing VSR on compressed data inevitably amplifies compression artifacts (*e.g.*, the 1st column), while results pre-enhanced with our method maintain accurate edges and textures, avoiding distortions seen in other methods.

Optical flow estimation. We adopt RAFT [49], DEQ [3], and KPAFlow [35] as baseline models for optical flow estimation. Evaluation on the KITTI-2015 dataset [21] is summarized with EPE (end-point-error) and F1-all loss, where lower values indicate better accuracy. As shown in Tab. 3, our method consistently reduces the EPE and F1-all loss across all baseline models, demonstrating its effectiveness in improving optical flow estimation. In contrast, methods such as MFQE 2.0, STDF and Metabit fail to deliver consistent improvements. For instance, MFQE 2.0 fails to improve the performance of RAFT on CRF15, STDF and Metabit detrimentally affects the performance of DEQ and KPAFlow on CRF15. Visualizations of predicted optical flow are shown in Fig. 8, where inaccurate boundaries are highlighted with red arrows. As can be seen, optical flow estimated from compressed inputs contains inaccurate boundaries, especially near-motion ones. The proposed method helps to deliver more accurate results in these regions compared to others. For instance, it effectively corrects the optical flow near the car that was mispredicted by DEQ, while MFQE 2.0 and S2SVR provide limited improvement.

Video object segmentation. For video object segmentation



STCN + MFQE 2.0 + STDF + S2SVR + Metabit + Ours GT

Figure 9. Qualitative results of VOS. As can be seen, directly performing VOS on compressed inputs leads to inaccurate masks, whereas pre-enhancing with our method effectively improves the accuracy, especially for the regions of irregular shapes (*e.g.*, the windshield).

Masked frame E^2FGVI + MFQE 2.0 + STDF + S2SVR + Metabit + Ours

Figure 10. Qualitative results of video inpainting. As can be seen, pre-enhancing the compressed inputs with the proposed method helps to reduce the artifacts and color distortion in the removed region, providing more visually pleasant results.

(VOS), we adopt STCN [14], DeAoT [57] and QDMN [34] as representative baselines. Evaluations on DAVIS-17 val dataset [41] are summarized with the following metrics: the $\mathcal{J}$ (average IoU), the $\mathcal{F}$ score (boundary similarity), and the average of the above metrics (denoted as Avg). Higher values indicate better segmentation accuracy. As shown in Tab. 4, the proposed method shows the best performance in improving accuracy across VOS models. For instance, it elevates the average accuracy for up to 2.37% (79.20% to 81.57%) on STCN at CRF35, while MFQE 2.0, STDF and S2SVR yield limited improvement of 0.08%, 0.57% and 0.84%, respectively. And Metabit provides no improvement on STCN at CRF35. The results of VOS are included in Fig. 9, where accurately segmenting objects in compressed videos is challenging for VOS baselines (*e.g.*, inaccurate mask of the windshield predicted by STCN). Pre-enhancing the compressed videos with MFQE 2.0 and S2SVR struggles to address this issue, whereas the proposed method significantly refines the segmentation results, demonstrating its effectiveness in assisting the VOS task.

Video inpainting. We take E^{2}FGVI [31] as the video inpainting model and perform video object removal on DAVIS-17 val dataset [41]. The qualitative results are shown in Fig. 10. As can be seen, compression-included misalignment between objects and masks hinders the ability to remove specified objects, causing color distortions (*e.g.*, the horse region). Pre-enhancing the compressed inputs with the proposed method notably refines the artifacts and distortions, yielding more visually pleasing results.

5.3. Ablation Studies

We start with a baseline that concatenates reference frames and current frames as input, without using codec information. We then progressively equip the baseline with MV alignment, region-aware refinement, sequence adaptation, and frame adaptation to assess their contributions.

MV alignment. As shown in the 2nd row of Tab. 5, incorporating MV alignment yields a PSNR gain of up to 0.65 dB on CRF15, demonstrating the effectiveness of MV in aligning reference frames and current frame.

Model	CRF15	CRF25	CRF35
	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑
Baseline	41.04 / 0.9785	34.92 / 0.9363	29.25 / 0.8238
+ MV Align.	41.69 / 0.9821	35.59 / 0.9437	29.95 / 0.8403
+ RA Refine.	42.04 / 0.9837	35.70 / 0.9449	30.00 / 0.8427
+ Seq. Adapt.	42.08 / 0.9838	35.76 / 0.9458	30.04 / 0.8444
+ Frame Adapt.	42.14 / 0.9839	35.81 / 0.9460	30.09 / 0.8446

Table 5. Ablation studies on MV alignment, region-aware refinement, sequence adaptation, and frame adaptation.

Region-aware refinement. We further incorporate the region-aware refinement module to refine the features of different regions. As shown in the 3rd row of Tab. 5, it leads to notable PSNR gains of 0.35dB, 0.11dB and 0.05dB on CRF15, CRF25 and CRF35, respectively.

Sequence adaptation. As shown in the 4th row of Tab. 5, sequence adaptation brings PSNR gains of 0.04 dB, 0.06 dB and 0.04 dB on CRF15, CRF25 and CRF35, respectively.

Frame adaptation. As shown in the 5th row of Tab. 5, frame adaptation improves PSNR by 0.06 dB 0.05 dB and 0.05 dB on CRF15, CRF25 and CRF35, respectively. We further analyze its effectivness on improving the temporal consistency in the supplementary materials.

6. Conclusion

In this paper, we introduce a versatile codec-aware enhancement framework that adaptively handles diverse compression settings and serves as a plug-and-play enhancement module to consistently boost various downstream tasks. By reusing the off-the-shelf codec information, our method minimizes additional computational costs. Compared with existing compressed video enhancement solutions, it shows superority in both enhancement performance and robustness, making it possible to deploy pre-trained models on compressed videos without a significant performance drop.

Acknowledgments. We acknowledge funding from the National Natural Science Foundation of China under Grants 62131003 and 62021001.

References

[1] Video developer report 2022-2023. https://bitmovin.com/downloads/assets/bitmovin-7th-video-developer-report-2023-2024.pdf. 5

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pages 3366–3375, 2017. 2

[3] Shaojie Bai, Zhengyang Geng, Yash Savani, and J. Zico Kolter. Deep equilibrium optical flow estimation. In *CVPR*, 2022. 7

[4] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 3

[5] Yuxuan Cai, Hongjia Li, Geng Yuan, Wei Niu, Yanyu Li, Xulong Tang, Bin Ren, and Yanzhi Wang. Yolobile: Real-time object detection on mobile devices via compression-compilation co-design. In *AAAI*, pages 955–963, 2021. 1

[6] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *CVPR*, pages 4947–4956, 2021. 6

[7] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, pages 5972–5981, 2022. 6

[8] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *CVPR*, pages 5962–5971, 2022. 1

[9] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, pages 168–172, 1994. 5

[10] Jin Chen, Xijun Wang, Zichao Guo, Xiangyu Zhang, and Jian Sun. Dynamic region-aware convolution. In *CVPR*, pages 8064–8073, 2021. 3

[11] Peilin Chen, Wenhan Yang, Long Sun, and Shiqi Wang. When bitstream prior meets deep prior: Compressed video super-resolution with learning from decoding. In *ACM MM*, pages 1000–1008, 2020. 1, 2

[12] Peilin Chen, Wenhan Yang, Meng Wang, Long Sun, Kangkang Hu, and Shiqi Wang. Compressed domain deep video super-resolution. *IEEE Transactions on Image Processing*, 30:7156–7169, 2021. 1, 2

[13] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *CVPR*, pages 11030–11039, 2020. 2

[14] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021. 8

[15] Yuanying Dai, Dong Liu, and Feng Wu. A convolutional neural network approach for post-processing in hevc intra coding. In *MultiMedia Modeling: 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I 23*, pages 28–39. Springer, 2017. 2

[16] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. Spatio-temporal deformable convolution for compressed video quality enhancement. In *AAAI*, pages 10696–10703, 2020. 1, 2, 5, 6

[17] Cunhui Dong, Haichuan Ma, Zhuoyuan Li, Li Li, and Dong Liu. Temporal wavelet transform-based low-complexity perceptual quality enhancement of compressed video. *IEEE TCSVT*, pages 1–1, 2023. 2

[18] Max Ehrlich, Jon Barker, Namitha Padmanabhan, Larry Davis, Andrew Tao, Bryan Catanzaro, and Abhinav Shrivastava. Leveraging bitstream metadata for fast, accurate, generalized compressed video quality enhancement. In *WACV*, pages 1517–1527, 2024. 1, 2, 5, 6

[19] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013. 2

[20] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 7

[22] Zhenyu Guan, Qunliang Xing, Mai Xu, Ren Yang, Tie Liu, and Zulin Wang. Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3): 949–963, 2019. 1, 2, 5, 6

[23] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456, 2021. 2

[24] Hongyue Huang, Ionut Schiopu, and Adrian 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):2100–2113, 2020. 2

[25] Zhijie Huang, Jun Sun, and Xiaopeng Guo. Fastcnn: Towards fast and accurate spatiotemporal network for hevc compressed video enhancement. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19 (3):1–22, 2023. 2

[26] Chuanmin Jia, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Jiaying Liu, Shiliang Pu, and Siwei Ma. Content-aware convolutional neural network for in-loop filtering in high efficiency video coding. *IEEE Transactions on Image Processing*, 28(7):3343–3356, 2019. 2

[27] Nanfeng Jiang, Weiling Chen, Jielian Lin, Tiesong Zhao, and Chia-Wen Lin. Video compression artifacts removal with spatial-temporal attention-guided enhancement. *IEEE Transactions on Multimedia*, 2023. 2

[28] Feng Li, Yixuan Wu, Anqi Li, Huihui Bai, Runmin Cong, and Yao Zhao. Enhanced video super-resolution network towards compressed data. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(7):1–21, 2024. 2

[29] Yinxiao Li, Pengchong Jin, Feng Yang, Ce Liu, Ming-Hsuan Yang, and Peyman Milanfar. Comisr: Compression-informed video super-resolution. In *ICCV*, pages 2543–2552, 2021. 2

[30] Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. Dynamic transfer for multi-source domain adaptation. In *CVPR*, pages 10998–11007, 2021. 3

[31] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, 2022. 8

[32] Jing Lin, Xiaowan Hu, Yuanhao Cai, Haoqian Wang, Youliang Yan, Xueyi Zou, Yulun Zhang, and Luc Van Gool. Unsupervised flow-aligned sequence-to-sequence learning for video restoration. In *ICML*, pages 13394–13404. PMLR, 2022. 1, 2, 5, 6

[33] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *CVPR*, pages 806–814, 2015. 5

[34] Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, and Yujiu Yang. Learning quality-aware dynamic memory for video object segmentation. In *ECCV*, pages 468–486, 2022. 8

[35] Ao Luo, Fan Yang, Xin Li, and Shuaicheng Liu. Learning optical flow with kernel patch attention. In *CVPR*, pages 8906–8915, 2022. 7

[36] Di Ma, Fan Zhang, and David R Bull. Cvegan: a perceptually-inspired gan for compressed video enhancement. *Signal Processing: Image Communication*, page 117127, 2024. 2

[37] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018. 2

[38] Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter Crozier, Carlos Fernandez-Granda, and Eero Simoncelli. Adaptive denoising via gaintuning. *NeurIPS*, 34:23727–23740, 2021. 3

[39] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, pages 0–0, 2019. 5, 6

[40] Zhaoqing Pan, Xiaokai Yi, Yun Zhang, Byeungwoo Jeon, and Sam Kwong. Efficient in-loop filtering based on enhanced deep convolutional neural networks for hevc. *IEEE Transactions on Image Processing*, 29:5352–5366, 2020. 2

[41] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 5, 8

[42] Darren Ramsook and Anil Kokaram. Learnt deep hyperparameter selection in adversarial training for compressed video enhancement with a perceptual critic. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2420–2424. IEEE, 2023. 2

[43] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *CVPRW*, pages 587–597, 2018. 1

[44] Mennatullah Siam, Alex Kendall, and Martin Jagersand. Video class agnostic segmentation benchmark for autonomous driving. In *CVPR*, pages 2825–2834, 2021. 1

[45] Gary J Sullivan, Pankaj N Topiwala, and Ajay Luthra. The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions. *Applications of Digital Image Processing XXVII*, 5558:454–474, 2004. 5

[46] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. 3

[47] Weiheng Sun, Xiaohai He, Chao Ren, Shuhua Xiong, and Honggang Chen. A quality enhancement network with coding priors for constant bit rate video coding. *Knowledge-Based Systems*, 258:110010, 2022. 1, 2

[48] Xuebin Sun, Sukai Wang, Miaohui Wang, Shing Shin Cheng, and Ming Liu. An advanced lidar point cloud sequence coding scheme for autonomous driving. In *MM*, pages 2793–2801, 2020. 1

[49] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 7

[50] Jianyi Wang, Mai Xu, Xin Deng, Liquan Shen, and Yuhang Song. Mw-gan+ for perceptual quality enhancement on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4224–4237, 2021. 2

[51] Shiyao Wang, Hongchao Lu, and Zhidong Deng. Fast object detection in compressed video. In *ICCV*, pages 7104–7113, 2019. 1

[52] Yingwei Wang, Takashi Isobe, Xu Jia, Xin Tao, Huchuan Lu, and Yu-Wing Tai. Compression-aware video super-resolution. In *CVPR*, pages 2012–2021, 2023. 2

[53] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003. 3

[54] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *NeurIPS*, 32, 2019. 2

[55] Ren Yang, Mai Xu, Zulin Wang, and Tianyi Li. Multi-frame quality enhancement for compressed video. In *CVPR*, pages 6664–6673, 2018. 1, 2

[56] Ren Yang, Mai Xu, Tie Liu, Zulin Wang, and Zhenyu Guan. Enhancing quality for hevc compressed videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(7): 2039–2054, 2019. 2

[57] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022. 8

[58] Li Yu, Wenshuai Chang, Shiyu Wu, and Moncef Gabbouj. End-to-end transformer for compressed video quality enhancement. *IEEE Transactions on Broadcasting*, 70(1):197–207, 2024. 2

[59] Haichao Zhang, Jianchao Yang, Yanning Zhang, Nasser M Nasrabadi, and Thomas S Huang. Close the loop: Joint blind image restoration and recognition with sparse representation prior. In *ICCV*, pages 770–777, 2011. 1

[60] Hengsheng Zhang, Xueyi Zou, Jiaming Guo, Youliang Yan, Rong Xie, and Li Song. A codec information assisted framework for efficient compressed video super-resolution. In *ECCV*, pages 220–235. Springer, 2022. 2

[61] Saiping Zhang, Luis Herranz, Marta Mrak, Marc Górriz Blanch, Shuai Wan, and Fuzheng Yang. Dcngan: A deformable convolution-based gan with qp adaptation for perceptual quality enhancement of compressed video. In *ICASSP*, pages 2035–2039. IEEE, 2022. 2

[62] Zhengdong Zhang and Vivienne Sze. Fast: A framework to accelerate super-resolution processing on compressed videos. In *CVPRW*, pages 19–28, 2017. 2

[63] Hengrun Zhao, Bolun Zheng, Shanxin Yuan, Hua Zhang, Chenggang Yan, Liang Li, and Gregory Slabaugh. Cbren: Convolutional neural networks for constant bit rate video quality enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4138–4149, 2022. 2

[64] Qiang Zhu, Jinhua Hao, Yukang Ding, Yu Liu, Qiao Mo, Ming Sun, Chao Zhou, and Shuyuan Zhu. Cpga: Coding priors-guided aggregation network for compressed video quality enhancement. In *CVPR*, pages 2964–2974, 2024. 2