Plug-and-Play Versatile Compressed Video Enhancement –Supplementary Materials–

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

This supplementary document is organized as follows:

– Section 1 provides a detailed explanation and pseudo-code to clarify the procedure for enhancing compressed frames.

– Section 2 reports quantitative comparisons for quality enhancement in highly compressed scenarios (*i.e.*, CRF40, CRF45 and CRF48) to demonstrate the robustness of the proposed method.

– Section 3 provides more qualitative comparisons on quality enhancement (Section 3.1) and downstream tasks (Section 3.2), including video super-resolution, optical flow estimation, video object segmentation, and video inpainting.

– Section 4 presents results of extending the proposed framework to compressed video super-resolution to demonstrate its applicability across various domains.

– Section 5 provides visual results of incorporating MV alignment and region-aware refinement, analyzing the number of experts and impact of frame adaption for improving the temporal consistency.

– Section 6 introduces details of experimental settings, including the dataset preparation, baseline methods, and implementation details.

– Section 7 discusses related works that also focus on downstream vision tasks, and further analyzes applicable scenarios of these works and the proposed method.

1. Procedure of Quality Enhancement

The goal of compressed video enhancement is to reconstruct high-quality outputs $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_T\}$ from compressed inputs $\{x_1, x_2, ..., x_T\}$. Our proposed framework achieves this through two key components: the compression-aware adaptation (CAA) network, denoted as $\mathcal{G}_\phi$, and the bitstream-aware enhancement (BAE) network, denoted as $\mathcal{F}_{\theta_i}$, which ensure adaptively handling different compression settings and reconstructing high-fidelity content, respectively. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Procedure of Enhancing Compressed Frames

```
Input: Sequence-wise $CRF_s$, Frame-wise $CRF_i$, Input frames $\{x_1, x_2, ... , x_n\}$, Motion vectors $MV$, Partition map $P_i$
Output: Enhanced high-quality frames $\{\hat{y}_1, \hat{y}_2, ... , \hat{y}_n\}$
1: Sequence adaptation
       $f_{\theta_s} \leftarrow \mathcal{G}_{\phi_s}(CRF_s, \{f_{\theta_1}, f_{\theta_2}, ... , f_{\theta_N}\})$
2: for $x_i \in \{x_1, x_2, ... , x_T\}$ do
3:     Frame adaptation
           $\mathcal{F}_{\theta_i} \leftarrow f_{\theta_i} \leftarrow \mathcal{G}_{\phi_i}(CRF_i, f_{\theta_s})$
4:     Motion vector alignment
           $\hat{x}_i \leftarrow [MV(h_i^p), MV(h_i^f), x_i]$
5:     Region-aware refinement
           $\hat{y}_i \leftarrow \mathcal{F}_{\theta_i}(\hat{x}_i, P_i)$
6: end for
7: return $\{\hat{y}_1, \hat{y}_2, ... , \hat{y}_n\}$
```

Compression-aware adaptation (CAA) network $\mathcal{G}_\phi$ focuses on hierarchical parameters adaptation, consisting of sequence-wise weight generator $\mathcal{G}_{\phi_s}$ and frame-wise parameters generator $\mathcal{G}_{\phi_i}$ to adaptively tailor the enhancement model to the characteristics of compressed frames (see Step 1 and Step 3). The obtained frame-wise expert layer f_{θ_i} further constructs the subsequent bitstream-aware enhancement network $\mathcal{F}_{\theta_i}$ (as shown in Step 3).

Bitstream-aware enhancement (BAE) network $\mathcal{F}_{\theta_i}$ frame-wisely applies techniques such as motion vector (MV) alignment (as shown in Step 4) and region-aware refinement (as shown in Step 5) to enhance temporal consistency and reconstruct fine-detailed results.

2. Quantitative Results

To assess the quality enhancement performance of each method in highly compressed scenarios, we conduct evaluations at CRF values of 40, 45 and 48 and summarize the results with PSNR and SSIM (the higher the better). Please note that the above CRF values are not included during training. The results of the REDS4 dataset [16] are reported

[*] Corresponding author.

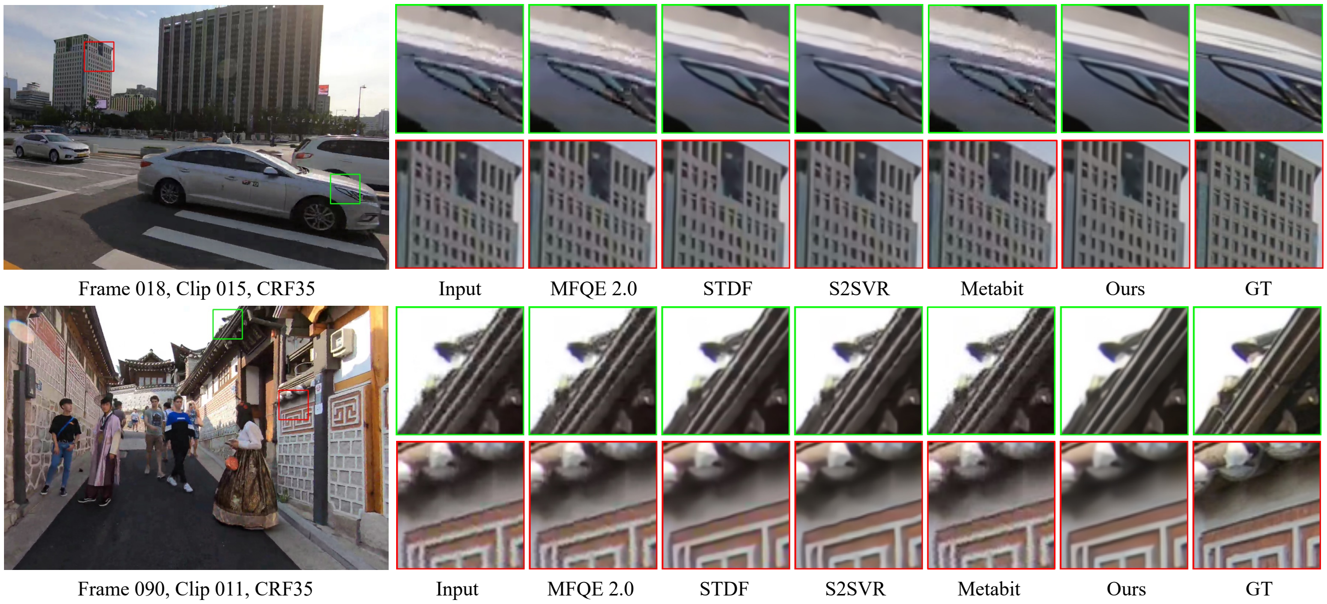


Figure 1. Qualitative results on quality enhancement, where the results are evaluated on the REDS4 dataset [16]. As can be seen, our method demonstrates its effectiveness in reducing compression artifacts, resulting in visually appealing outputs with clear details. In contrast, the compared methods fail to fully suppress these artifacts, leaving noticeable distortions (*e.g.*, the car in the 1st row).

in Table 1. As can be seen, performing frame-wise adaptation with slice type (marked with grey) achieves a similar performance (less than 0.03 dB in terms of PSNR) to the original design. Additionally, the proposed method shows robust performance in enhancing the highly compressed inputs, achieving PSNR gains of 0.74 dB, 0.46 dB and 0.33 dB on CRF40, CRF45 and CRF48, respectively. In contrast, the other methods provide limited and even no improvement. For instance, STDF [3] and S2SVR [14] achieve a minor PSNR gain of 0.04 dB and 0.41 dB at CRF40, respectively. MFQE 2.0 [7] and Metabit [4] show no improvement on the highly compressed inputs, indicating their dependency on a well-designed training strategy to cope with a wide range of CRFs instead of a general mix-training strategy of various compression levels.

3. More Qualitative Comparisons

3.1. Quality Enhancement

We provide visual comparisons on the task of quality enhancement in Figure 1. As can be seen, MFQE 2.0 [7] and Metabit [4] fail in eliminating the compression artifacts, leading to the texture distortion (*e.g.*, the car in the 1st row). Despite STDF [3] and S2SVR [14] effectively refining the compressed frames, they struggle to eliminate the color distortion and provide artifact-free results (*e.g.*, the building in the 2nd row). In contrast, the proposed method effectively eliminates the compression artifacts and corrects the color distortion, achieving visually satisfying results.

Method	CRF40	CRF45	CRF48
	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑	PSNR↑ / SSIM↑
Input	26.69 / 0.7352	24.38 / 0.6452	23.17 / 0.5989
MFQE 2.0 [7]	26.69 / 0.7369	24.37 / 0.6466	23.16 / 0.6001
STDF [3]	27.03 / 0.7477	24.54 / 0.6544	23.26 / 0.6058
S2SVR [14]	27.10 / 0.7506	24.59 / 0.6575	23.30 / 0.6091
Metabit [4]	26.69 / 0.7352	24.38 / 0.6452	23.17 / 0.5988
Ours	27.42 / 0.7619	24.82 / 0.6697	23.47 / 0.6201
	27.43 / 0.7619	**24.84 / 0.6697**	**23.50 / 0.6215**

Table 1. Quantitative results on quality enhancement, where the evaluation is conducted in highly compressed scenarios (*i.e.*, CRF40, CRF45 and CRF48) and summarized with PSNR and SSIM (the higher the better). The best and second best results are highlighted with **bold** and underline. Results obtained by replacing frame-wise CRF_i with slice type are highlighted with grey .

3.2. Versatility Evaluation

Video super-resolution. As shown in Figure 2, it is challenging to apply video super-resolution (VSR) models that are tailored for clean data to compressed inputs, leading to the amplification of compression artifacts, as observed in the 1st column. Equipping the baselines with pre-enhancing methods such as MFQE 2.0 [7] and Metabit [4] provides limited quality improvement, and STDF [3] struggles to adequately suppress these artifacts (*e.g.*, the car in the 3rd row). In contrast, pre-enhancing with our method and S2SVR [14] achieves artifact-free results, preserving the sharp edges and details of the content. Notably, our approach outperforms S2SVR [14] in terms of

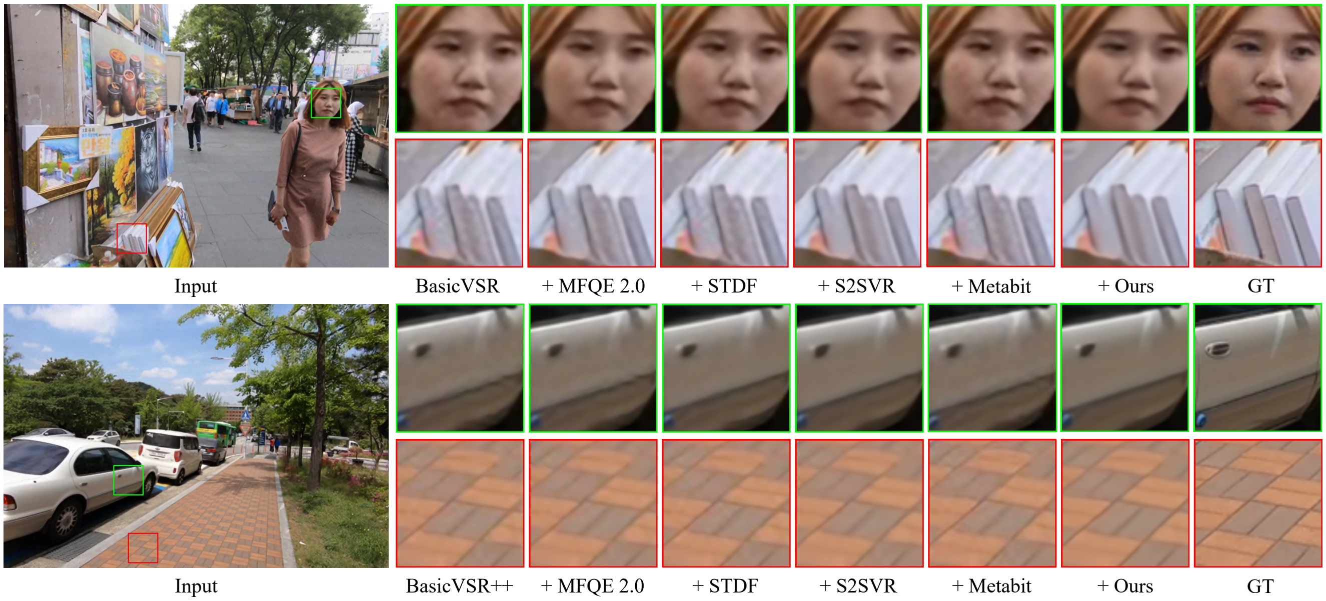


Figure 2. Qualitative results of ×4 video super-resolution on the REDS4 dataset [16]. As can be seen, pre-enhancing compressed frames with our method effectively prevents the amplification of compression artifacts. While the other enhancement methods struggle to eliminate the artifacts and even severe the distortions in some cases (*e.g.*, STDF [3] in the 4th row).

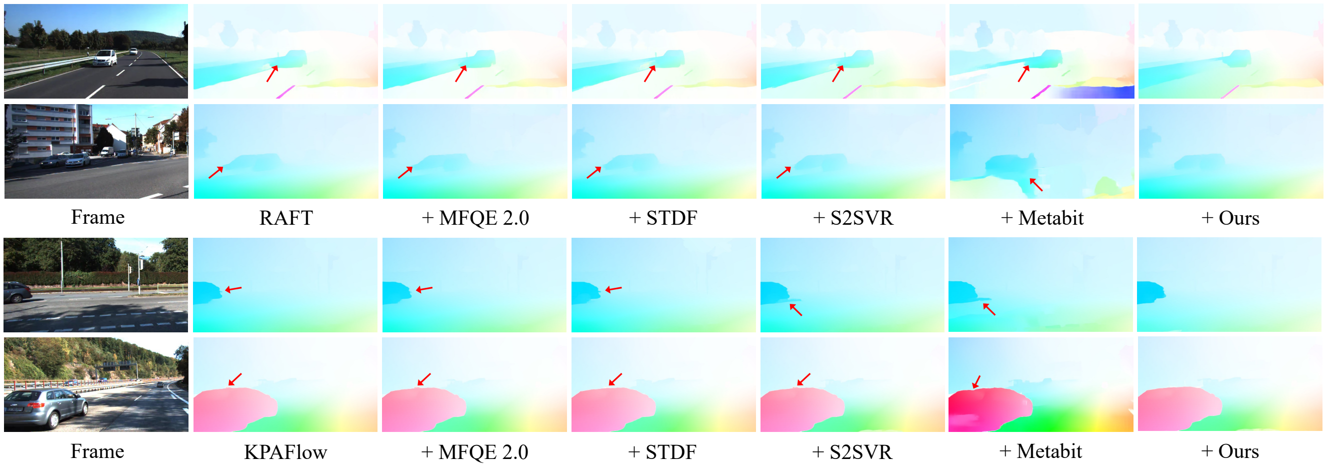


Figure 3. Qualitative results of optical flow estimation on the KITTI-2015 dataset [6], where we mark the inaccurate boundaries with red arrows. As can be seen, equipping the baseline models with our method effectively improves the accuracy at the boundaries of moving objects (*e.g.*, the moving car of the 1st row).

model complexity and computational efficiency, achieving significantly lower model complexity and faster processing speeds, as detailed in Tab. 1.

Optical flow estimation. Figure 3 presents the visualizations of predicted optical flow, with inaccurate boundaries highlighted by red arrows. As can be seen, when estimating optical flow from compressed inputs, the inaccuracy is particularly prominent near motion boundaries (*e.g.*, the front of the car in the 1st row). In contrast, the proposed method demonstrates superior performance in addressing these issues, delivering more accurate results in these challenging regions compared to other methods. For instance, in the 1st row, our method effectively corrects the optical flow errors produced by RAFT [20], whereas both MFQE 2.0 [7] and S2SVR [14] fail to provide notable improvements, and Metabit [4] perturbs the performance of downstream optical flow estimation. This highlights the effectiveness of our method in assisting the downstream optical flow estimation on compressed videos.

Video object segmentation. The results of video object segmentation are visualized in Figure 4. As can be seen, accurately segmenting the objects in compressed images is

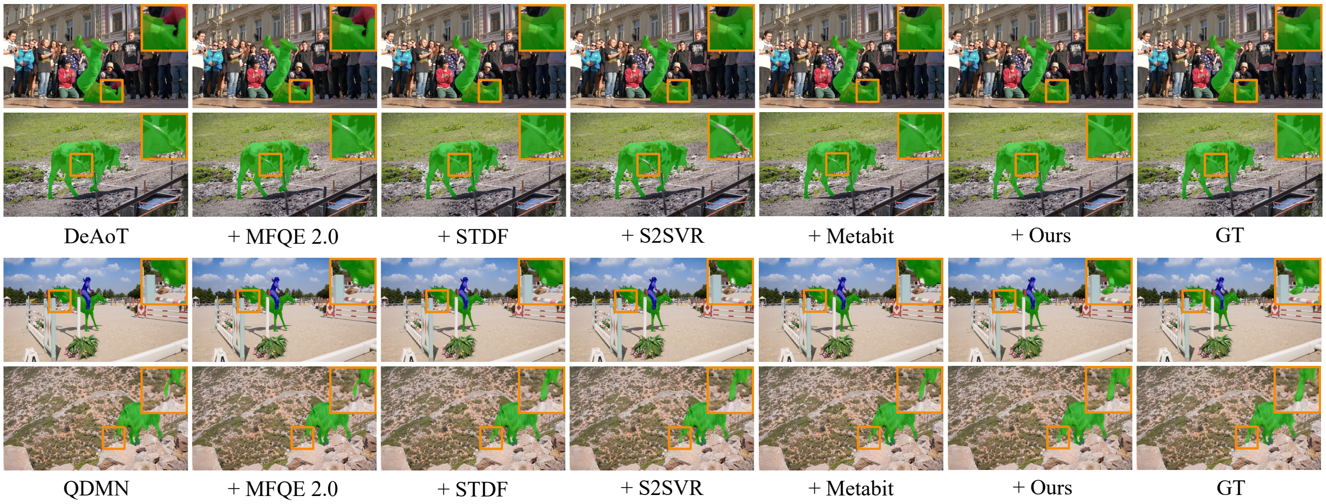


Figure 4. Qualitative results of video object segmentation on DAVIS-17 val dataset [17]. Directly performing VOS on compressed images often results in inaccurate masks (*e.g.*, results in the 1st column). In contrast, pre-enhancing the compressed inputs with our proposed method significantly improves mask accuracy (*e.g.*, the tail in the 4th row).

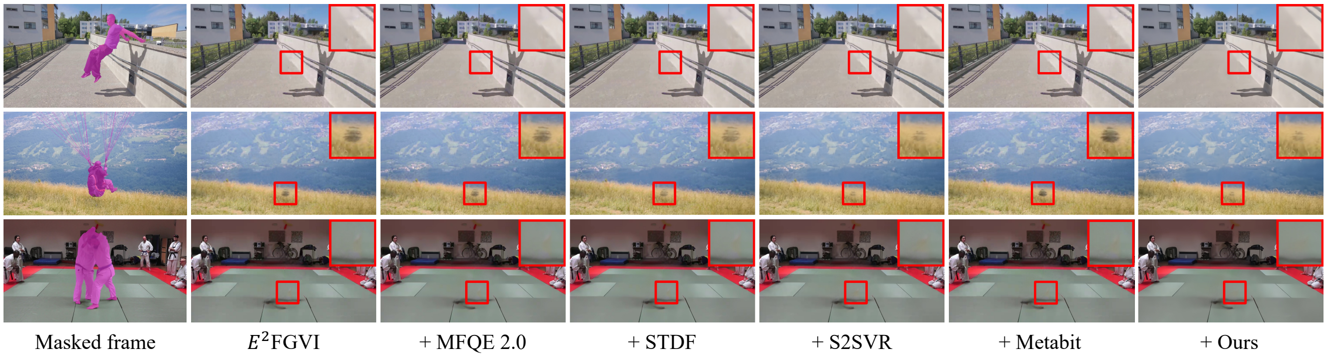


Figure 5. Visual results of video inpainting on the DAVIS-17 val dataset [17]. As can be seen, pre-enhancing the compressed inputs with the proposed method significantly reduces artifacts and color distortions in the removed regions (*e.g.*, the horse hoof in the 3rd row).

challenging for VOS baselines (*e.g.*, under-segmented mask of the tail predicted by DeAoT [21]). Nevertheless, such inaccuracy is not adequately -addressed by pre-enhancing the input videos with methods such as MFQE 2.0 [7], S2SVR [14], and Metabit [4]. In contrast, the proposed method effectively mitigates errors and improves mask accuracy, underscoring the effectiveness of our method in supporting VOS on compressed video data.

Video inpainting. To further investigate the versatility of our method, we extend the downstream task to video inpainting, a generative task that needs to handle blurred object boundaries due to image compression [22]. The results of removing the specified objects from compressed frames are shown in Figure 5. As can be seen, due to the misalignment between compressed objects and their masks, it is hard for E^{2}FGVI [13] to adequately remove the specified object, resulting in noticeable artifacts and color distortions in the removed region (*e.g.*, the wall in the 1st row). In contrast, pre-enhancing the compressed inputs using our proposed method substantially improves the inpainting results, effectively mitigating artifacts and delivering results with consistent structures, demonstrating our capability of enhancing generative tasks under compression conditions.

4. Compressed Video Super-Resolution

The proposed method is designed to be versatile, without any assumptions about downstream tasks, which ensures broad applicability across various domains. Yet, it can be readily adapted for specific applications when required. Here we demonstrate this adaptability with the application to 4× video super-resolution for compressed videos. By expanding 30 region-aware refinement-integrated residual blocks and incorporating a pixel shuffle layer at the end of the network, we convert the enhancement network into a VSR-specific one. We follow COMISR [11] to prepare the compressed training dataset and adopt the same train-

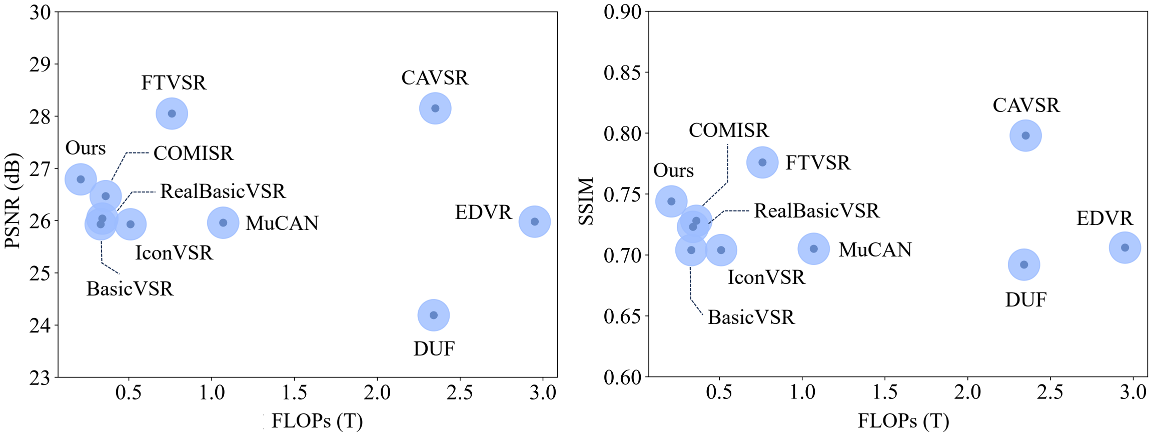
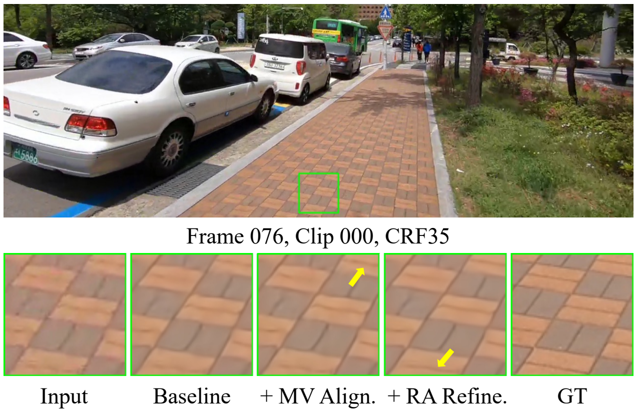


Figure 6. FLOPs and performance comparison of 4× compressed video super-resolution on the REDS4 dataset [16], where the compression level is set to CRF25. Despite not being tailored for VSR, the proposed method shows competitive performance.


Frame 076, Clip 000, CRF35

Input Baseline + MV Align. + RA Refine. GT

Figure 7. Qualitative results of the ablation study on MV alignment (MV Align.) and region-aware refinement (RA Refine.). As can be seen, incorporating the region-aware refinement effectively reduces distortions and enhances the textures.

ing configuration. The quantitative results at the compression level of CRF25 are summarized with PSNR/SSIM, and reported in Figure 6. As can be seen, although the proposed method is not tailored for VSR, it still provides competitive results with minimal computational complexity. For instance, the proposed method outperforms IconVSR [2] by 0.86 dB in terms of PSNR, costing only 0.41× of FLOPs. Additionally, our method achieves a PSNR gain over COMISR [11] (specifically designed for compressed VSR) by 0.23 dB, while taking 0.58× FLOPs. This indicates the versatility and potential of our method to serve as a general solution for leveraging codec information in specialized tasks.

5. Ablation Studies

In this section, we present visual results from ablation studies to assess the impact of incorporating MV alignment and region-aware refinement into the baseline model (as illustrated in Sec. 5.3 of the submission). Additionally, we analyze the effect of varying the number of experts (N) on model performance. These experiments are conducted on the REDS [16] dataset, with models trained for 50K iterations for fast evaluation. The results are summarized with PSNR and SSIM.

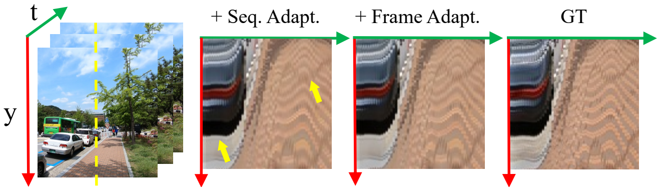


Figure 8. Visualization of the temporal profile, which tracks a specified column (marked with the yellow dotted line) over time.

MV alignment. As shown in Figure 7, aligning frames with motion vectors (denoted as + *MV Align.*) effectively improves the texture inconsistency, as highlighted by the yellow arrow. This demonstrates the effectiveness of MV alignment in aligning and propagating high-quality reference frames, therefore improving the overall quality of compressed videos.

Region-aware refinement. As shown in Figure 7, refining features with the guidance of partition map (denoted as + *RA Refine.*) effectively reduces distortions and enhances the fine details (*e.g.*, the boundary of bricks marked by the yellow arrow), obtaining results with coherent textures.

Frame adaptation. To assess its impact on temporal consistency, a comparison of the temporal profile is included in Figure 8. As can be seen, frame-wise adaptation helps to adaptively enhance each frame, resulting in a smoother temporal transition (as indicated by the yellow arrows).

Number of experts. We investigate the number of ex-

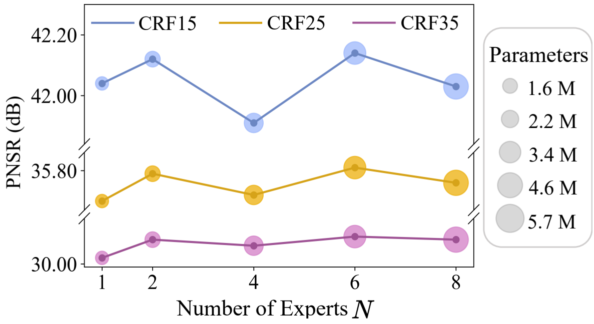


Figure 9. Ablation study on the number of experts. The design of mixing experts leads to notable performance improvement, and the configuration of 6 experts is selected to balance the performance and model complexity.

perts by setting different values for N. As shown in Figure 9, compared to a simple single-expert network, increasing N effectively improves the performance but does not yield consistent performance gains. Based on the results, we adopt $N = 6$ as it achieves optimal results with manageable model complexity.

6. Experimental Settings

Dataset preparation. We adopt the widely-used H.264 [19] standard and FFMPEG to generate compressed videos by specifying the CRF values (*i.e.*, 15, 25 and 35). The CRF_s value and slice type of each compressed sequence are extracted from the header. MVmed [1] is applied to extract motion vectors and partition maps.

Compared methods and downstream models. For the task of quality enhancement, we follow the official suggestions to locate keyframes with slice types for MFQE 2.0 [7]. For STDF [3], we adopt the STDF-R3L variant. Since Metabit [4] only addresses I/P frames, we reimplement it to adapt the adopted dataset that contains I/P/B frames. For the task of video object segmentation (VOS), we adopt the SwinB-DeAOT-L variant from DeAoT [21] to ensure strong VOS performance.

Implementation details. In practice, expert layers are implemented with convolutional layers initialized with Kaiming initialization [9]. The sequence-wise weight generator is constructed with two fully connected layers followed by a softmax activation. The parameters re-weighting is implemented with dynamic parameters mechanism [8]. The frame-wise parameters generator is constructed with two fully connected layers and a sigmoid normalization. Introducing parameters $\triangle\theta_i$ for f_{θ_s} is implemented with dynamic transfer mechanism [12]. The bitstream-aware enhancement network is constructed with 8 region-aware refinement-integrated residual blocks. Each block contains 64 channels. The FLOPs and inference speed are computed with an input size of 320×180 on a GeForce GTX 1080 Ti GPU. We merge the training splits of the REDS [16] and DAVIS [17] datasets for training, and further augment the dataset by downsampling the REDS dataset [16] using the Bicubic interpolation at a scaling factor of 4. During training, input frames are sampled from uncompressed data and compressed data with probabilities of 0.2 and 0.8, respectively. The compressed input frames are sampled from CRF15, CRF25 and CRF35 with equal probability. These frames are then randomly augmented with horizontal flips, vertical flips, and rotations. The length of input sequences is set to 15 and the batchsize is set to 10. The input patch size is set to 128×128. We adopt the Adam optimizer [10] with $\beta1 = 0.9$, $\beta2 = 0.99$. The initial learning rate is set to 2×10^{-4} and adjusted with the Cosine Annealing scheme [15]. The whole training takes iterations of 250K. We use 2 Nvidia GeForce RTX 3090 GPUs to complete these experiments.

7. Discussions

We explore the role of video enhancement in improving the performance of downstream tasks. Recent advancements in video codecs also introduce task-aware encoding [5] and decoding [18] frameworks to better support downstream tasks. However, these approaches typically require joint training of the compression model and target downstream tasks. In contrast, our approach serves as a plug-and-play adapter to enhance the performance of downstream models, making our method more practical, particularly in scenarios where the downstream task is unknown or subject to change. A promising strategy would be prioritizing our approach when the downstream task is ambiguous or not specified, while leveraging the aforementioned methods when the task is well-defined and can directly benefit from the integrated task-aware compression.

References

[1] L. Bommes, X. Lin, and J. Zhou. Mvmed: Fast multi-object tracking in the compressed domain. In *ICIEA*, pages 1419–1424, 2020. 6

[2] 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. 5

[3] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. Spatio-temporal deformable convolution for compressed video quality enhancement. In *AAAI*, pages 10696–10703, 2020. 2, 3, 6

[4] 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. 2, 3, 4, 6

[5] Xingtong Ge, Jixiang Luo, Xinjie Zhang, Tongda Xu, Guo Lu, Dailan He, Jing Geng, Yan Wang, Jun Zhang, and Hong-

wei Qin. Task-aware encoder control for deep video compression. In *CVPR*, pages 26036–26045, 2024. 6

[6] 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. 3

[7] 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. 2, 3, 4, 6

[8] 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. 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 6

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[11] 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. 4, 5

[12] Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. Dynamic transfer for multi-source domain adaptation. In *CVPR*, pages 10998–11007, 2021. 6

[13] 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. 4

[14] 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. 2, 3, 4

[15] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[16] 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. 1, 2, 3, 5, 6

[17] 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. 4, 6

[18] Xihua Sheng, Li Li, Dong Liu, and Houqiang Li. Vnvc: A versatile neural video coding framework for efficient human-machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 6

[19] 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. 6

[20] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 3

[21] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022. 4, 6

[22] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava, and Ser-Nam Lim. Deep video inpainting detection. *arXiv preprint arXiv:2101.11080*, 2021. 4