Compressed-Domain-Aware Online Video Super-Resolution

Supplementary Material

1. Video Coding Basics

Modern video codecs such as H.264 organize frames into different types according to their prediction structure. An *I-frame* (intra-coded frame) is encoded using only spatial prediction within the frame itself, without referencing other frames, and thus serves as a self-contained random-access point. A *P-frame* (predictive frame) is encoded using temporal prediction from previously decoded reference frames: each block is predicted through motion-compensated reference from a previously decoded frame, and only the corresponding residual and motion information are stored. In addition, codecs support *B-frames* (bi-directionally predictive frames), which are predicted from both past and future reference frames to further improve compression efficiency at the cost of increased delay and decoding complexity. Since we focus on online video scenarios with causal processing and low latency, we adopt an IP-only coding structure and do not use B-frames in our experiments.

In this work, we are mainly interested in inter-frame coding, which is the core mechanism behind temporal compression. Figure 1 provides a schematic overview of inter-frame encoding. For a given block in the current frame, the encoder searches the reference frame for the best-matching block. The displacement between the current and the matched reference block is encoded as a two-dimensional **motion vector (MV)**. Using the MV, the decoder can reconstruct a motion-compensated prediction by warping the reference frame toward the current time step. Because the prediction is not perfect, the encoder also computes the difference between the original block and its motion-compensated prediction as a **residual map**. During decoding, the reference frame is first reconstructed, then warped using the decoded MVs to obtain the prediction, and finally the decoded residual is added back to recover the current frame.

2. Dataset Generation Details

REDS and REDS4. The REDS dataset [2] is employed for training. This dataset contains 720p video sequences with diverse scenes and large inter-frame motion, which makes it particularly suitable for video super-resolution. For evaluation, the commonly used REDS4 dataset [2] is adopted. We first generate low-resolution (LR) video frames by applying $4\times$ downsampling in MATLAB.

```
I_lr = imresize(I_hr, 1/4, 'bicubic');
```

After downsampling, we compress the LR videos using FFmpeg 4.3.8 [5] with H.264 encoding [7]. The rate control

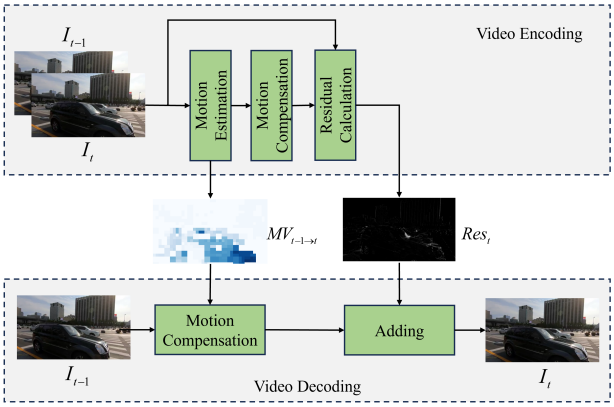

Figure 1. Schematic diagram of inter-frame encoding for video codecs.

mode is set to capped constant rate factor (Capped-CRF), which is widely used in online video streaming to balance quality and bandwidth [1, 3, 6]. Specifically, we use three CRF values $\{18, 23, 28\}$ and their corresponding maximum bitrates of 1, 0.5, and 0.3 Mbps, respectively, to simulate different compression levels.

For example, the FFmpeg command used for `CRF` $=$ 18 and `maxrate` $=$ `1M` is

```
ffmpeg -framerate 24 \
    -i <INPUT> \
    -c:v libx264 \
    -preset medium \
    -crf 18 \
    -maxrate 1M \
    -bufsize 2M \
    -keyint_min 25 \
    -g 25 \
    -bf 0 \
    -sc_threshold 0 \
    <OUTPUT>
```

Here, `-bf 0` disables B-frames and yields an IP-only coding structure, which matches our online and low-latency setting. The options `-g 25` and `-keyint_min 25` enforce a fixed GOP length of 25 frames, i.e., one I-frame every 25 frames. In addition, `-sc_threshold 0` turns off the scene-cut detector so that the encoder does not insert extra I-frames at scene boundaries, keeping a consistent prediction structure across all sequences.

We further extract motion vectors (MVs), residual maps, and frame types from the encoded bitstreams.

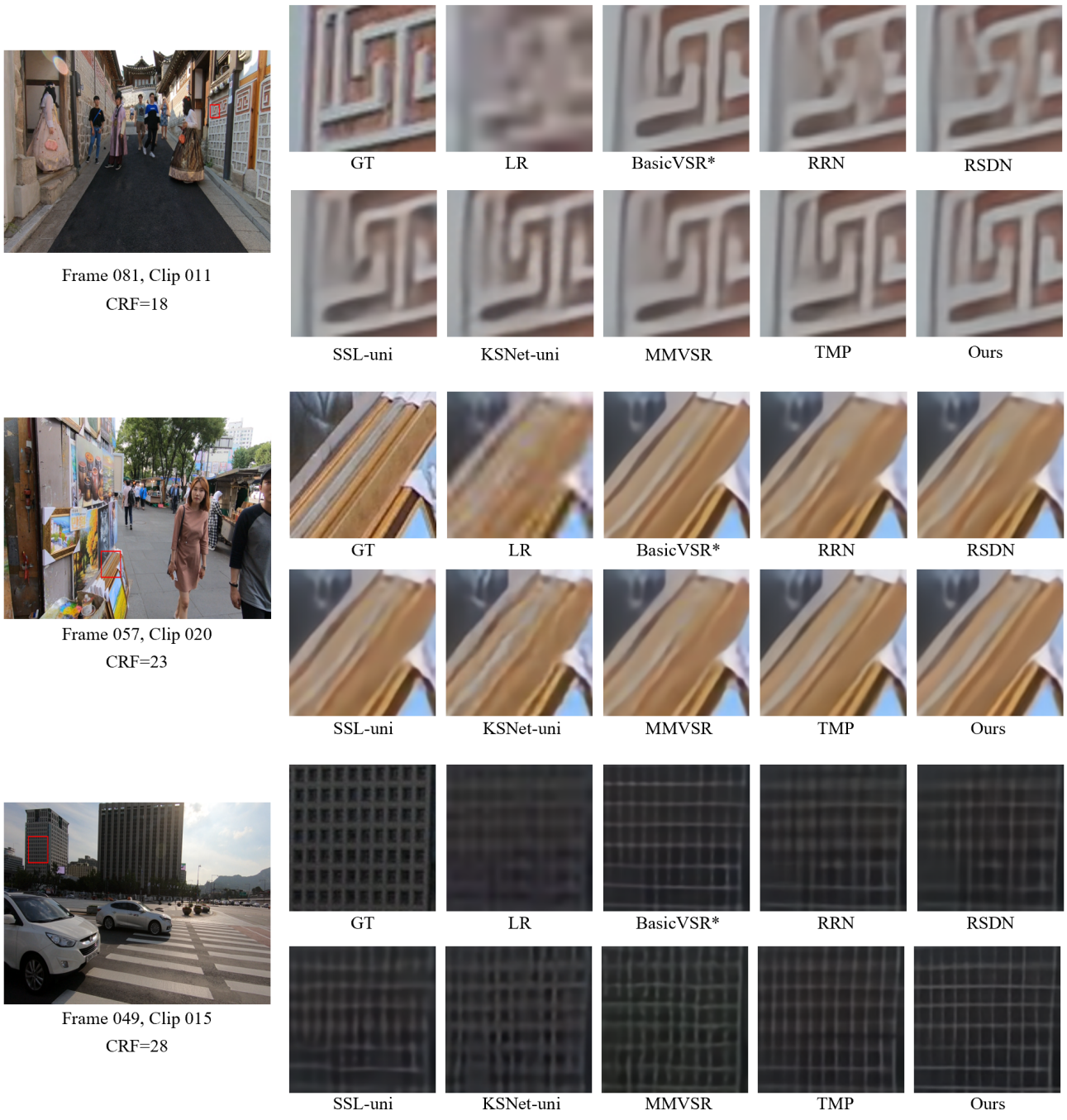

Figure 2. Qualitative comparison of different online VSR methods on the REDS4 dataset.

Inter4K. We further use the Inter4K dataset [4] to evaluate our method at higher resolutions. Inter4K is a high-quality dataset for video interpolation and super-resolution. It contains 1,000 ultra-high-resolution videos at 60 frames per second (fps), collected from online resources. The dataset provides standardized versions at multiple spatial resolutions, including ultra-high definition (UHD/4K), quad high definition (QHD/2K), full HD (FHD/1080p), HD (720p), quarter HD (qHD, 540p), and ninth HD (nHD, 360p). In our experiments, we select four clips from Inter4K, each containing 300 consecutive frames. For each selected clip, we use three spatial resolutions: 2K, 1080p,

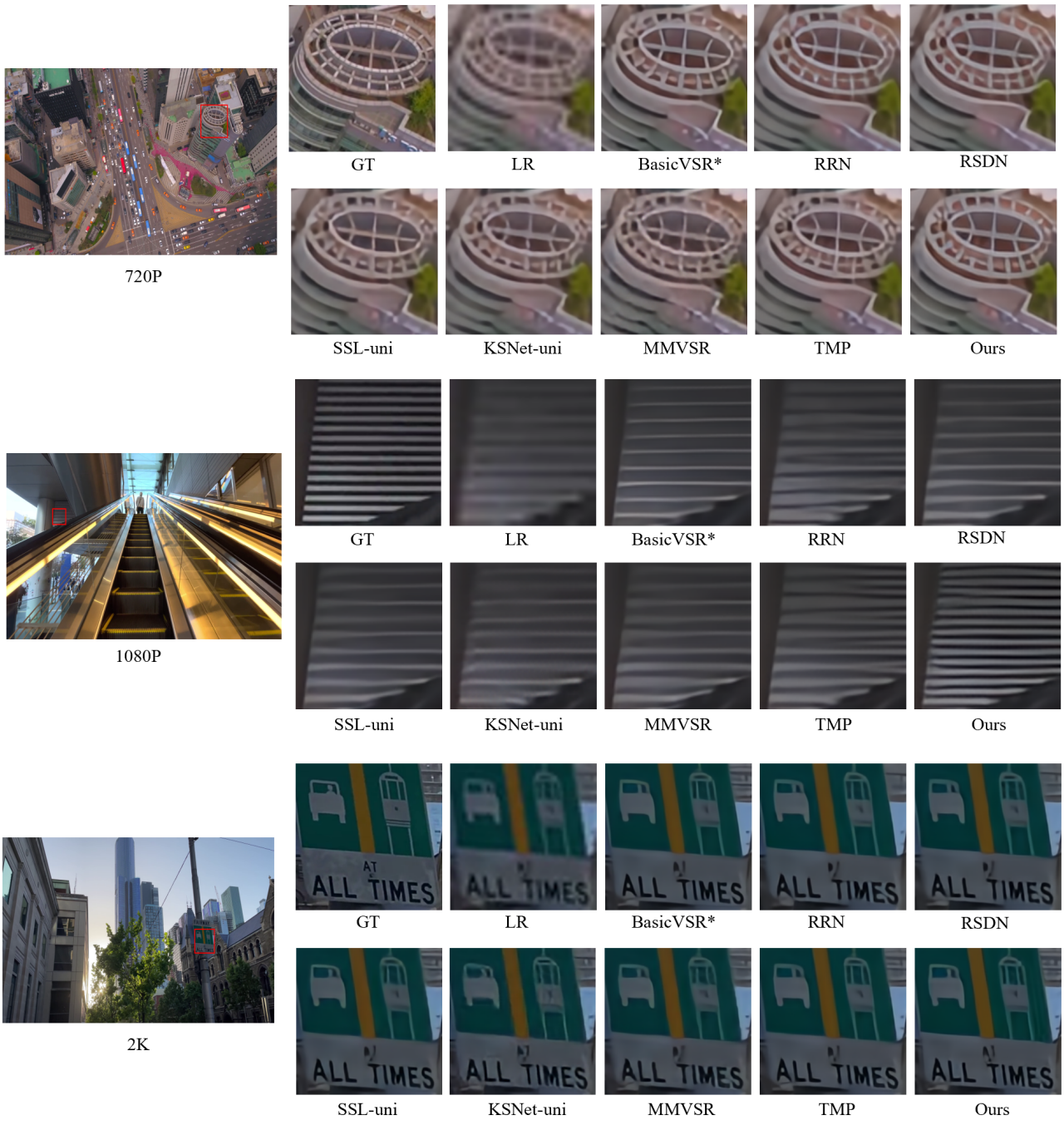

Figure 3. Qualitative comparison of different online VSR methods on the Inter4K dataset.

and 720p. We do not include native 4K resolution in our experiments because it significantly increases memory consumption and latency, making it difficult to satisfy our real-time constraint and to run all competing methods under the same single-GPU hardware budget. Moreover, 2K and 1080p already cover the most common resolutions in current online video streaming scenarios.

To reduce the experimental cost at higher resolutions, we use a single CRF value of 23 on Inter4K and vary only the bitrate caps according to the resolution. Following the same downsampling and compression pipeline as for REDS4, the Inter4K sequences are encoded with H.264 using Capped-

Table 1. Cross-Codec Generalization on HEVC (H.265).

Method	Runtime↓ (ms)	Params↓ (M)	PSNR(dB)↑ / SSIM↑ / LPIPS↓		
			CRF18	CRF23	CRF28
TMP	22.2	3.1	27.28/0.7650/0.3553	26.27/0.7279/0.3789	25.03/0.6832/0.4260
Ours	10.8	3.2	27.36/0.7683/0.3348	26.46/0.7340/0.3768	25.23/0.6888/0.4248

CRF mode with a CRF value of 23. To account for different spatial resolutions, we adopt resolution-dependent bitrate caps: for the 720p, 1080p, and 2K versions, the maximum bitrates are set to 0.5, 1, and 1.5 Mbps, respectively.

3. Additional Qualitative Results

Due to space limitations in the main paper, we provide additional qualitative results in this supplementary material. We present more visual comparisons on both the REDS4 and Inter4K datasets, covering a diverse set of motion patterns, textures, and compression levels, as shown in Figure 2 and 3. For better visual inspection, we display higher-resolution crops and enlarge key regions of interest, allowing clearer observation of differences in fine details between our method and competing approaches. Across all cases, our method consistently recovers sharper structures, cleaner textures, and fewer compression artifacts than prior online VSR methods, which is in line with the quantitative gains reported in the main paper. In particular, compared with the state-of-the-art method TMP [8], our approach delivers superior visual quality while running at over $2\times$ the inference speed (in FPS), achieving a more favorable trade-off between reconstruction accuracy and efficiency.

4. Codec Generalization Beyond H.264

Our method relies on *codec-agnostic* compressed-domain information (motion vectors, residual maps, and frame types) rather than any H.264-specific syntax. Such information is available in modern hybrid codecs (H.264/AVC, HEVC/H.265, and AV1), although their extraction and noise characteristics may vary. Therefore, the proposed modules are expected to generalize in principle. To support this claim, we additionally conducted a simple evaluation under HEVC (H.265). As shown in Table 1, our method consistently outperforms TMP.

5. Discussion of limitations and failure cases

Under extremely severe compression, motion vectors can become highly noisy or unreliable (e.g., large or inaccurate MVs) and residual maps may also be corrupted, which weakens temporal correspondence and can degrade performance. In such cases, our residual-gated fusion can suppress unreliable propagated information to mitigate error accumulation, but the benefit from temporal redundancy is reduced.

References

[1] Ashuthosh Dubey. Video streaming with capped CRF, 2025. 1

[2] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1996–2005. IEEE, 2019. 1

[3] NETINT Technologies. Optimizing video streaming with capped constant rate factor (CRF) encoding, 2024. 1

[4] Alexandros Stergiou and Ronald Poppe. Adapool: Exponential adaptive pooling for information-retaining downsampling. *IEEE Transactions on Image Processing*, 32:251–266, 2022. 2

[5] Suramya Tomar. Converting video formats with ffmpeg. *Linux journal*, 2006(146):10, 2006. 1

[6] Visionular-admin. What is capped CRF?, 2024. 1

[7] 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. 1

[8] Zhengqiang Zhang, Ruihuang Li, Shi Guo, Yang Cao, and Lei Zhang. Tmp: Temporal motion propagation for online video super-resolution. *IEEE Transactions on Image Processing*, 2024. 4