Continuous Space-Time Video Resampling with Invertible Motion Steganography

Supplementary Material

1. Dataset and implementation details

1.1. Datasets

Vimeo90k Vimeo90k [21] is a commonly used dataset for video frame interpolation, video super-resolution, and spatiotemporal video super-resolution. It contains 64,612 training clips and 7,824 testing clips, with each clip consisting of 7 consecutive frames at a spatial resolution of 448 × 256. We use Vimeo90K as the training and testing set for fixed temporal and fixed resolution resampling tasks, specifically including Time 2× Space 1×, Time 2× Space 4×, and Time 2× Space 1×. Additionally, to investigate whether different motion magnitudes affect the quality of downsampling results (motion steganography), we follow [17] and split the Vimeo90k test set into fast motion, medium motion, and slow motion subsets, containing 1,225, 4,977, and 1,613 video clips, respectively. We also remove several video clips from the test set that contain consecutive all-black frames, as they can lead to infinite PSNR values.

Vid4 Vid4 [8] is a classic dataset for evaluating video super-resolution methods. It contains 4 video clips (calendar, city, foliage, and walk), each with at least 34 frames (720 × 480). We follow the experimental settings of [16] and use the 4 sequences in the Vid4 dataset to compare different resampling models.

Adobe240FPS The Adobe240FPS [9] dataset contains 133 720P hand-held videos captured at 240 FPS. Of these, 100 sequences are used for training, and 17 sequences are used to test non-integer frame rate conversion. We use this dataset, captured under high-frame-rate conditions, as the training set for the continuous spatiotemporal resampling model. Specific details of the dataset preparation are provided in the following sections.

SPMCS SPMCS [13] includes 32 videos and is widely used as a benchmark for video super-resolution. It contains rich textures and is sensitive to different scaling factors. We use it to evaluate the performance of continuous spatiotemporal resampling.

1.2. Training and testing details

Details for Fixed Spatialtemporal Resampling. As mentioned in the main text, the training process consists of two stages. In the first stage, we train the resampling model using Charbonnier loss in an end-to-end manner, supervising only the output of the upsampler. The loss function is formulated as:

$$loss_{rec} = \frac{1}{T}\sum_{i=1}^{T} L_{char}(I_i^{GT}, I_i^{SR}), \tag{1}$$

where I_i^{GT} and I_i^{SR} represent the ground truth and the reconstructed frames from the upsampler, respectively. In general, the downsampling results directly output by the model are in floating-point format, while in practical applications, images are typically quantized to unsigned 8-bit integers. Since the quantization process is non-differentiable during backpropagation, we adopt the differentiable quantization layer from STAA [16] to enable quantization-aware training. In the second training stage, which aims to embed motion information into low-frame-rate videos, we use the odd frames (e.g., 1^{st}, 3^{rd}, 5^{th}, 7^{th}) as ground truth to supervise the output of the IMSM. If spatial downsampling is involved, we use the bicubically downsampled results as low-resolution (LR) guidance. During training, we apply standard augmentation techniques, such as rotation, flipping, and random cropping. The training patch size is set to 192×192, and we train the network for 280,000 iterations using the Adam optimizer. The learning rate is initialized at 2×10^{-4} and decayed to 1×10^{-7} using cosine annealing. In the ablation study, to evaluate the effectiveness of the Invertible Motion Steganography Module (IMSM), we simultaneously supervised the output of both the downsampler and the upsampler. The total loss function is formulated as:

$$\begin{aligned} loss_{total} = \frac{1}{T}\sum_{i=1}^{T} L_{char}(HR_i, SR_i) + \\ \frac{1}{T'}\sum_{k=1}^{T'} L_{char}(LR_k^{Bic}, LR_k), \end{aligned} \tag{2}$$

where SR_i and LR_k denote the i^{th} super-resolved frame and the k^{th} downsampled frame, respectively. LR_k^{Bic} represents the k^{th} bicubically downsampled LR guidance.

Details for Continuous Spatiotemporal Resampling. To achieve controllable spatiotemporal resampling, we construct various combinations of resampling factors during training for spatiotemporal generalization using the high-frame-rate dataset [9]. Specifically, we perform temporal resampling with factors of (5/6, 4/5, 3/4, 2/3, 1/2), and spatial resampling from 2.0 to 4.0 with a step size of 0.2 (i.e., 2.0, 2.2, ..., 4.0). To facilitate understanding, we provide two sets of examples in Figure 1.

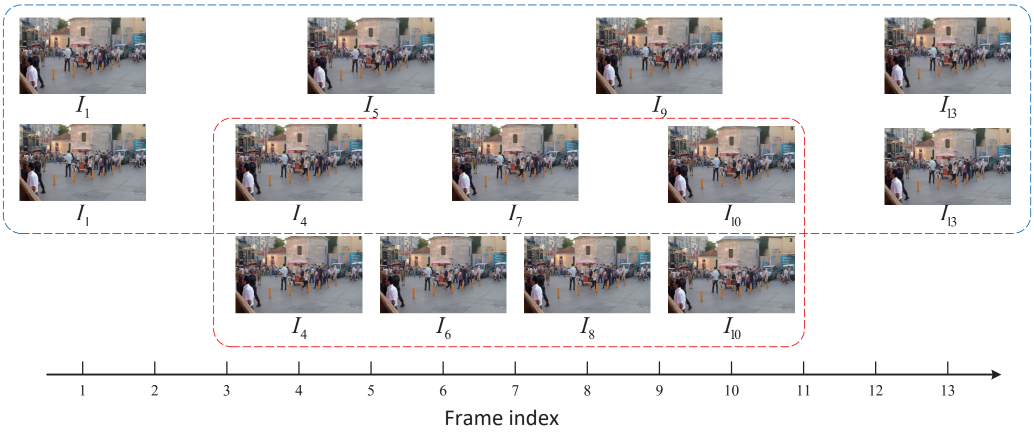


Figure 1. To construct a training set for non-integer frame rate conversion, we sample high-frame-rate videos at different intervals and align the timestamps to create training pairs. The red and blue boxes correspond to the cases of 3/4 and 4/5 frame rate conversions, respectively.

Table 1. Quantitative comparison with state-of-the-art methods for space 4 × resampling on benchmark datasets, metrics: PSNR(dB)/SSIM. The best and the second-best results are in red and blue, respectively.

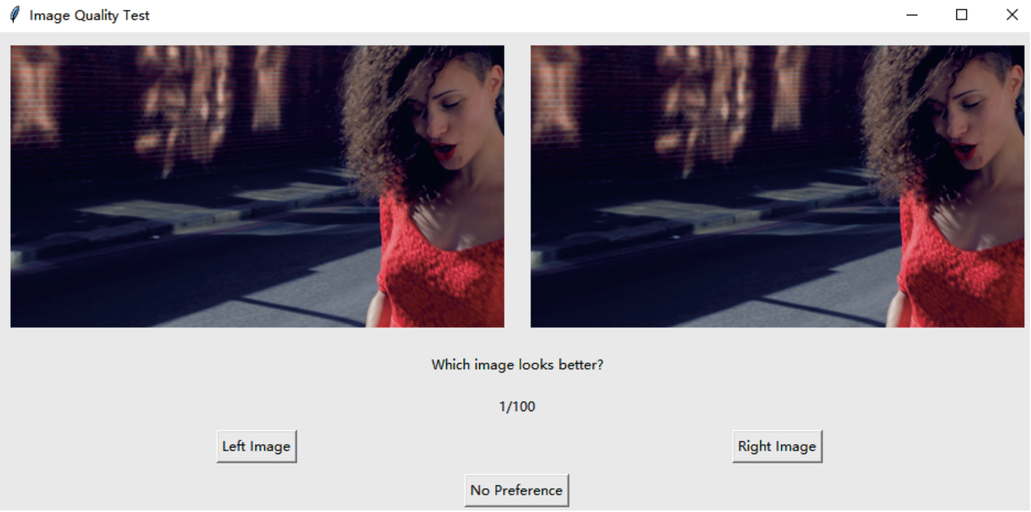
Method type	Methods	Vimeo90k		Vid4		Params
		RGB-Space	Y-Space	RGB-Space	Y-Space	
Image resampling	CAR [12] + EDSR [7]	35.64/0.9331	37.73/0.9484	26.77/0.8221	28.32/0.8423	52.9 M
	IRN[18]	37.68/0.9586	40.68/0.9731	29.18/0.8986	31.24/0.9184	4.4 M
	AIDN[19]	38.06/0.9591	40.93/0.9725	28.70/0.8880	30.66/0.9071	3.8 M
	HCFlow [6]	38.08/0.9618	41.17/0.9755	29.59/0.9065	31.71/0.9257	4.4 M
Video resampling	SelfC-small [14]	-	40.68/0.9756	-	31.61/0.9317	1.8 M
	SelfC-large [14]	38.50/0.9660	41.46/0.9784	30.31/0.9249	32.32/0.9404	3.3 M
	LSTM-VRN [2]	-	41.42/0.9764	-	32.24/0.9369	9.0 M
	MIMO-VRN [2]	-	43.26/0.9846	-	33.79/0.9577	19.2 M
	MIMO-VRN-C [2]	-	42.53/0.9820	-	33.40/0.9537	19.2 M
	Ours	40.02/0.9738	43.23/0.9858	32.33/ 0.9496	34.51/0.9628	17.4 M

2. Comparison for Spatial Video Resampling

Since the proposed method demonstrates significant advantages in temporal resampling, a natural question arises: does the motion steganography module also work effectively for spatial resampling alone? To verify this, we adopt a similar training strategy to spatiotemporal resampling, but with downsampling performed only in the spatial dimension by a factor of 4. Our comparison methods include image resampling techniques [6, 12, 18, 19] and video resampling methods [2, 14]. The comparison results are presented in Table 1. Despite not incorporating specific designs for spatial sampling (e.g., feature propagation or frame alignment), our method remains highly competitive, outperforming the comparison methods on the majority of metrics in both Vimeo90k and Vid4. Aligned with the experimental results in the main text, we also present ablation experiments for IMSM and different model complexities in Table 2 and Table 3. The results show that IMSM remains effective in cases where only spatial resampling is performed. This effectiveness can be attributed to the fact that we did not impose direct constraints on the downsampling results. In other words, we did not explicitly require the downsampling results to closely resemble any particular traditional sampling model (e.g., bicubic). This looser constraint actually benefits the upsampling recovery process. From another perspective, compared to the increasingly complex frame alignment module designs in video super-resolution tasks [20, 22], it may be more effective for video resampling to conduct end-to-end optimization of the entire joint downsampling-upsampling process. We hope these findings can inspire researchers to design more effective spatial resampling algorithms.

3. More Evaluations of the Downsampling Results

For resampling tasks, evaluating the quality of downsampling results is crucial. This distinguishes the task from video compression, as we expect the downsampled results to maintain a comparable visual quality to traditional downsampling methods. However, existing spatiotemporal re-



Which image looks better?

1/100

Figure 2. Interface for subjective visual experiments, where participants are asked to compare the visual quality of the two images on the left and right. They can select the one they find more visually appealing or choose ”no preference”.

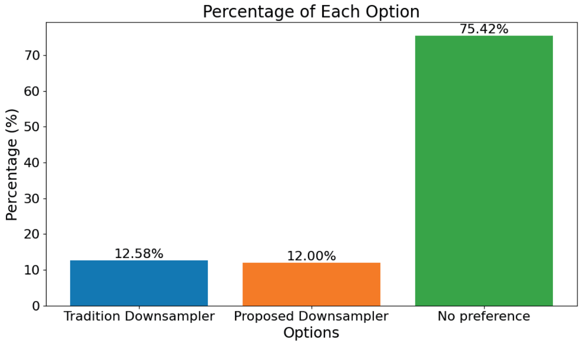


Figure 3. A subjective visual preference comparison between images containing embedded motion information and those generated by traditional downsampling methods. Most users cannot perceive any noticeable difference between the two.

Table 2. Ablation study of w/wo Invertible Motion Steganography Module(IMSM) on benchmark dataset for spatial 4× resampling, Metrics: PSNR-Y(dB)/SSIM-Y

Upscale rate	Vimeo90k		Vid4	
	w/o IMSM	w/ IMSM	w/o IMSM	w/ IMSM
T×1 S×4	41.83/0.9798	43.23/0.9858	32.92/0.9472	34.51/0.9628

Table 3. Ablation experiments on model performance under different complexities for spatial 4 × resampling, metrics: PSNR-Y(dB)/SSIM-Y.

Downsampling Rate	block num	Vimeo90k PSNR-Y SSIM-Y	Vid4 PSNR-Y SSIM-Y
T×1 S×4	12	41.99/0.9825	33.53/0.9550
	18	42.25/0.9834	33.83/0.9580
	24	43.23/0.9858	34.51/0.9682

Table 4. Full reference quantitative Comparison of downsampled image and its corresponding counterpart with traditional downsampling method (metrics: PSNR-Y/SSIM-Y). We apply bicubic sampling for spatial 4× downsample.

Downsampling Rate	Vimeo-Fast	Vimeo-Medium	Vimeo-Slow	Vimeo-Total
T×1 S×4	47.42/0.9970	46.15/0.9965	44.63/0.9953	46.04/0.9963

Table 5. No reference quantitative Comparison of downsampled image and its corresponding counterpart with traditional downsampling method (metrics: NIQE ↓).) We apply bicubic sampling for spatial 4× downsample.

Downsampling Rate	Vimeo-Fast		Vimeo-Medium		Vimeo-Slow		Vimeo-Total	
	Bicubic	Ours	Bicubic	Ours	Bicubic	Ours	Bicubic	Ours
T×1 S×4	12.05	11.64	14.30	12.86	6.48	6.34	12.33	11.32

sampling methods STAA [16] only impose constraints on the numerical range of downsampling results, lacking both qualitative and quantitative assessments. Therefore, we conduct a comprehensive evaluation of the downsampling results, including full-reference metrics, no-reference metrics, and subjective experiments. Consistent with the main text, the objective metrics for 4× spatial resampling are presented in Table 4 and Table 5. From the objective evaluation metrics, all downsampling results across various motion magnitudes and spatiotemporal combinations demonstrate high PSNR and SSIM values. And the results produced by our downsampling method also demonstrate superior NIQE scores. In addition to objective evaluation metrics, we conduct subjective experiments to verify that the hidden motion information does not affect subjective visual perception. We randomly select 100 pairs of images from Vimeo90K for evaluation and invite 24 participants to take part in the experiment. The evaluation interface is shown in Figure 2, where traditional spatiotemporal downsampled images are displayed alongside the proposed model's downsampled images containing hidden motion information—randomly shuffled. Participants are asked to choose which image look better based on ”clarity,” ”color saturation,” and ”overall visual quality.” The options include ”left image,” ”right image,” and ”no preference.” The results, summarized in Figure 3, indicate that in the vast majority of cases, participants could not perceive any significant differences between the two images and choose ”no preference.” In other cases, participants could not clearly distinguish which downsampling result is better. To statistically analyze the participants' choices, we employ the Chi-Squared Test. The null hypothesis (H0) for the Chi-Squared Test states that participants' preferences for images A and B are independent, indicating no significant difference. The alternative hypothesis (H1) posits that there is a significant difference in the participants' preferences for images A and B. The results of the Chi-Squared Test show that the Chi-Squared statistic is 0.178, while the p-value is 0.672. Based on a significance level of 0.05, we fail to reject the null hypothesis since the p-value exceeds this threshold. This implies that there is no significant difference in participants' choices between images A and B, indicating that the subjective visual difference between A and B is very small, making it nearly impossible to distinguish which image is visually better.

4. Frequency Analysis for Motion Steganography

In addition to comparing low-bit residuals, frequency analysis is a common method in image steganalysis. We use the Discrete Cosine Transform (DCT) to analyze images in the frequency domain. Specifically, we perform DCT transformations on the host image I_0 (original image) and the stego image $\hat{I}_0$ (image with hidden motion), dividing the results into eight frequency bands, from low to high frequency. The results are shown in Figure 4. We present two scenarios: in subfigure (a), there is noticeable movement between I_0 and I_1, while in subfigure (b), the movement is minimal. Frequency analysis reveals that differences between the host and stego images are larger in lower frequency bands (band 1, band 2) and minimal in higher bands (band 7, band 8), where the red and blue curves almost overlap. Comparing subfigure (a) and subfigure (b), we observe that images with greater movement exhibit larger differences in their DCT coefficients, indicating that complex motion significantly alters the frequency distribution. DCT-based image steganography typically hides information in mid to low-frequency bands, as modifying coefficients in these areas helps maintain hidden information stability. These frequency coefficients are less likely to be reduced during compression (such as quantization) compared to high-frequency coefficients. Additionally, slight changes to low-frequency coefficients generally go unnoticed, as they do not significantly alter the image's overall appearance. In contrast, changes to high-frequency coefficients may introduce noticeable noise or distortion, making them easier to detect.

Table 6. Comparison of the performance of the proposed method and FFmpeg in performing non-integer frame rate conversion on the Adobe dataset, metrics: PSNR-Y (dB)/SSIM-Y)

	20FPS → 24 FPS	24 FPS → 30 FPS
FFmpeg	27.53/0.7756	28.88/0.8067
Ours	38.31/0.9501	40.15/0.9626

5. Applications

Non-integer Frame Rate Upsampling. Given that many video interpolation algorithms can achieve integer frame rate conversions, this study focuses on non-integer frame rate conversions. To the best of our knowledge, no learning-based methods currently support multiple non-integer frame rate conversions using a single model. In practice, non-integer frame rate conversion often relies on tools such as FFmpeg, which typically employ techniques like linear interpolation, frame repetition, and timestamp adjustments to modify the frame rate. To validate the superiority of the learnable approach over traditional methods, we adopt a strategy similar to that illustrated in Figure 1, generating pairs of non-integer low-to-high frame rate data from high frame rate sources in the Adobe dataset to serve as both inputs and outputs for the upsampling module. We train an upsampler and test it on the Adobe test set, evaluating two configurations: (1) 20 FPS to 24 FPS, and (2) 24 FPS to 30 FPS. The experimental results, presented in Table 6, indicate that the proposed method significantly outperforms FFmpeg. Moreover, in certain cases, FFmpeg resorts to frame repetition to adjust the source video to the target frame rate, which can lead to noticeable stuttering during continuous playback. In contrast, our method avoids this issue, providing a smoother playback experience.

Efficient Video Resampling with Compression. Unlike the results of image downsampling, which are often saved in lossless PNG format, video downsampling typically undergoes lossy compression before being transmitted as a bitstream, primarily to reduce bandwidth from a practical application perspective. To verify the compatibility of the proposed method with existing compression frameworks, we replace the simple quantization operation following downsampling with a real H.265 codec [10]. The compression is performed using the default settings of FFmpeg x265 for all commands. Given that the x265 compression process

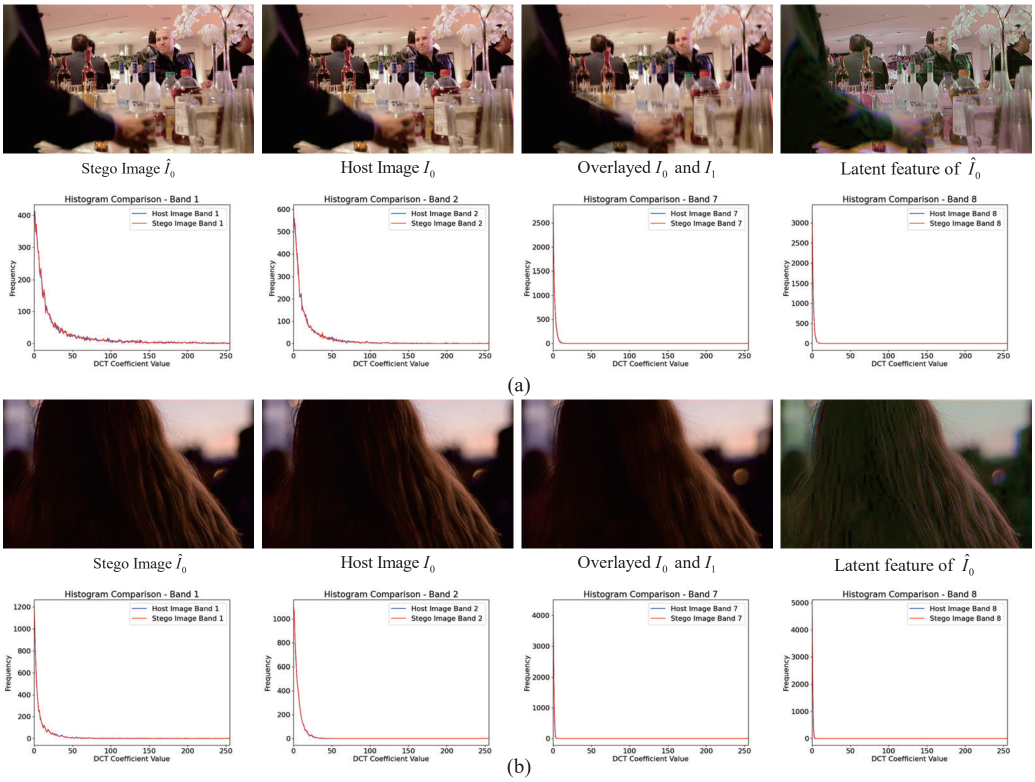


Figure 4. Analysis of the DCT coefficient histograms of different frequency bands after performing DCT transformation on the stego image and the corresponding host image. Two sets of images are provided, where (a) shows significant motion between adjacent frames, while (b) exhibits relatively little motion.

Table 7. BDBR (%) Results with the anchor of AIDN calculated by MS-SSIM.

Method	SelfC	x265	Ours
Rate saving	-12.50	-23.75	-33.54

is non-differentiable, during backpropagation, we directly pass the gradient from before the compression codec to the upsampler. Following previous work [14], we train a space 2× time 1× model and compare the experimental results with those obtained using the x265 encoder and two other learnable resampling methods [14, 19]. The results from x265 represent direct compression of the video with its original resolutions. From the rate-distortion curves in Figure 5 and the BDBR values in Table 7, it can be observed that, at similar bits per pixel (bpp), our results achieve better MS-SSIM performance. This also demonstrates that our motion steganography-based resampling scheme is robust to compression, which aligns with the conclusions drawn from the visualization results in Figure 4. Specifically, steganography primarily affects the low-frequency components of the DCT, which are not easily impacted by compression.

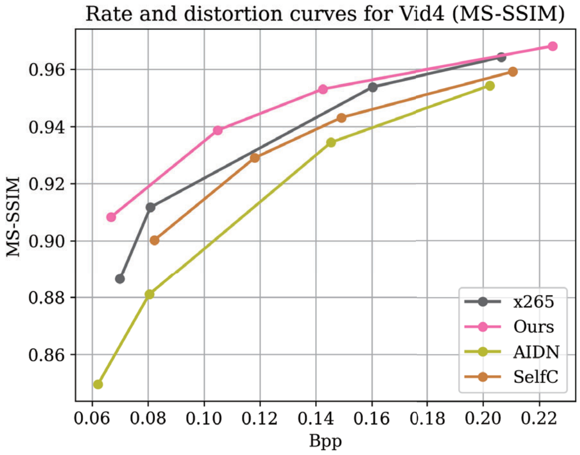


Figure 5. The rate-distortion performance of our approach compared with H.265 (x265 default) and the recent learned image/video resampling approaches on Vid4.

6. Differences from Existing Work

Due to constraints on the length of the main text, we will further discuss the differences between the proposed method and some existing works here.

Differences from MV-based Video Steganography. Motion vector (MV) based video steganography [3, 4] is an important approach within video steganography that has garnered increasing attention from researchers. This is because MV-based steganography offers a large embedding capacity due to the significant amount of motion vectors present in compressed video. However, our approach fundamentally differs from these MV-based methods. From a motivational perspective, the goal of MV-based video steganography is to maximize the capacity for hiding information within motion vectors and to enhance the reliability of information recovery. In contrast, our method aims to achieve high-fidelity temporal resampling of videos. From a task-oriented viewpoint, MV-based video steganography seeks to embed information within motion vectors, treating them as the host for hidden information. In our method, we embed motion information into low-frame-rate downsampled videos, where the motion information itself serves as the object to be concealed. In terms of specific implementation, MV-based methods often rely on complex, manually designed steps, such as selecting candidate MV sets and calculating embedding costs. These operations are deeply coupled with the overall video encoding process, requiring strict adherence to encoding standards [1, 11, 15]. In contrast, our method does not explicitly define motion vectors or optical flow but instead designs an end-to-end learnable model that implicitly implements the steganographic process.

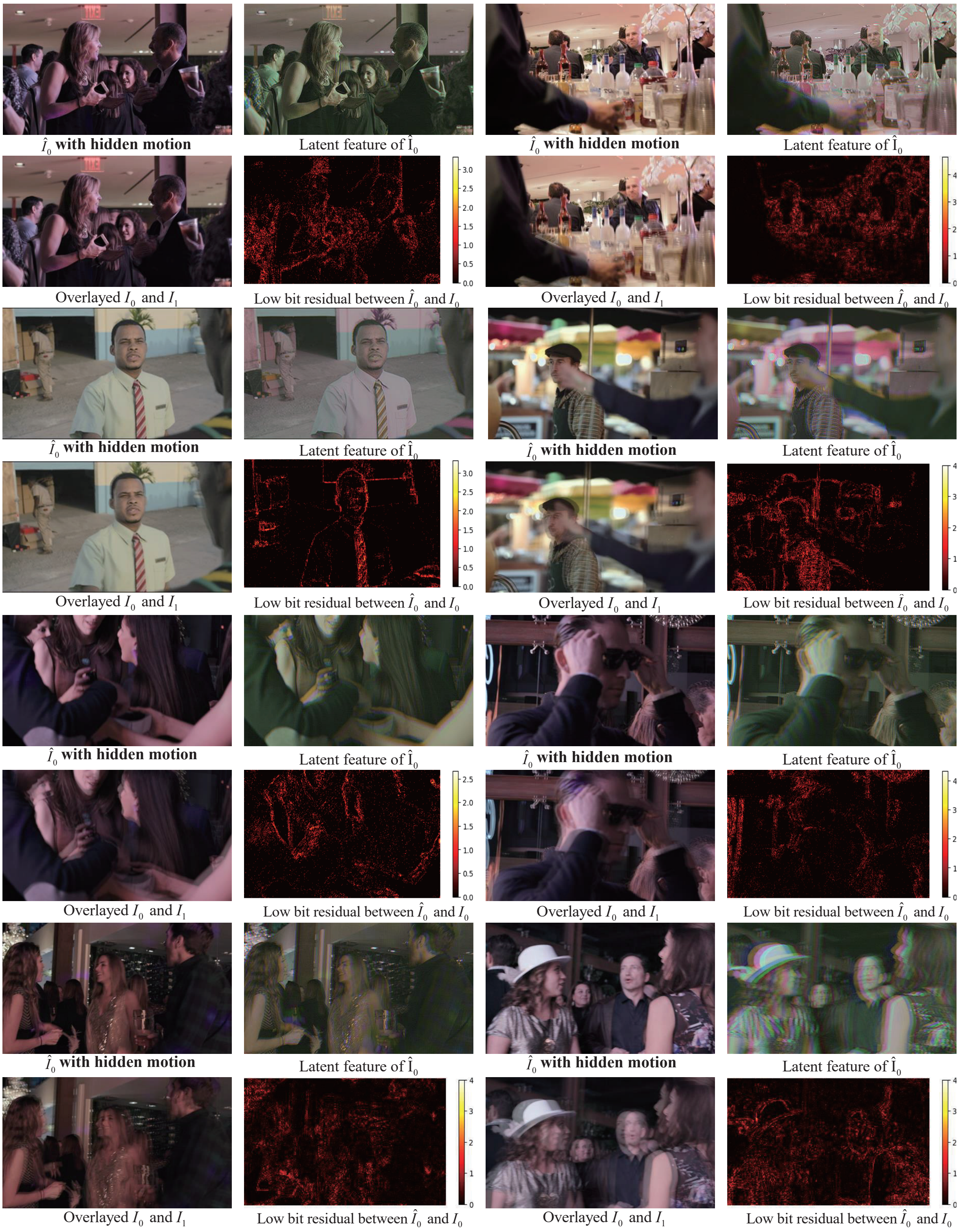
Differences from FGRN. FGRN [5] introduces an additional neural network to transform features into visually pleasing images during downsampling and reverses the process during upsampling. Although FGRN is similar in form to our approach, it differs significantly in terms of tasks addressed, motivations, model design, and optimization objectives. From a task perspective, FGRN focuses on the resampling of a **single image**, considering only the spatial correlation of downsampling results, whereas our method targets the spatiotemporal resampling of videos, with a primary focus on temporal resampling. In terms of motivation, FGRN aims to optimize the end-to-end resampling process of a single image to reduce direct constraints on the downsampling results. In contrast, our method's main objective is to imperceptibly embed high-frame-rate motion information into low-frame-rate videos to aid in high-frame-rate reconstruction. Regarding implementation, FGRN achieves only preset scaling for a single image, while our method enables the resampling of multiple frames at arbitrary frame rates, including optimizations tailored for video compression.

7. Limitations and Future Work

Although the proposed solution achieves excellent performance and high flexibility, there are still some limitations. First, similar to many existing video resampling methods, our approach has high complexity. Despite our efforts to adopt a lightweight design, such as using spatial-temporal separable 3D convolutions to form the Dense 3D Block, the inference complexity remains high and consumes a significant amount of GPU memory, making deployment on some resource-constrained edge devices challenging. Additionally, the capability of motion steganography for spatial upsampling has not been fully explored. We believe that further investigation into the relationship between motion estimation, motion compensation, and motion steganography could yield even higher performance.

8. More Visual Comparisions

Figure 6 provides additional visual results of low-bit steganalysis. In Figures 7, 8, 9 and 10, we present more visual comparisons of various methods under different spatiotemporal resampling combinations. Our method demonstrates considerable advantages across all spatiotemporal combinations, particularly in handling large motions and various nonlinear movements.



$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

$\hat{I}_0$ **with hidden motion**

Latent feature of $\hat{I}_0$

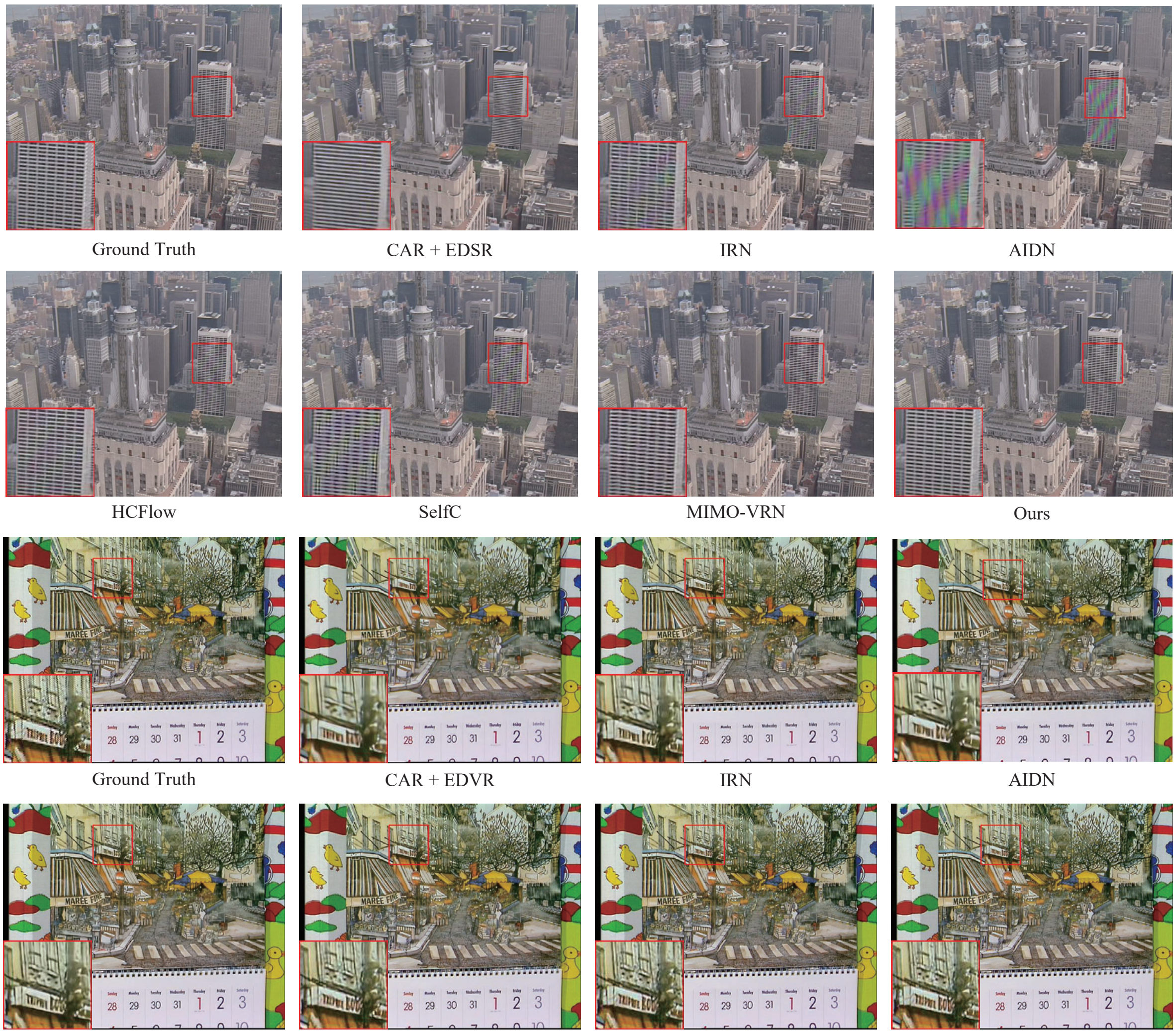
Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

Overlayed I_0 and I_1

Low bit residual between $\hat{I}_0$ and I_0

Figure 6. More visualizations of Low-Bit steganographic motion information.



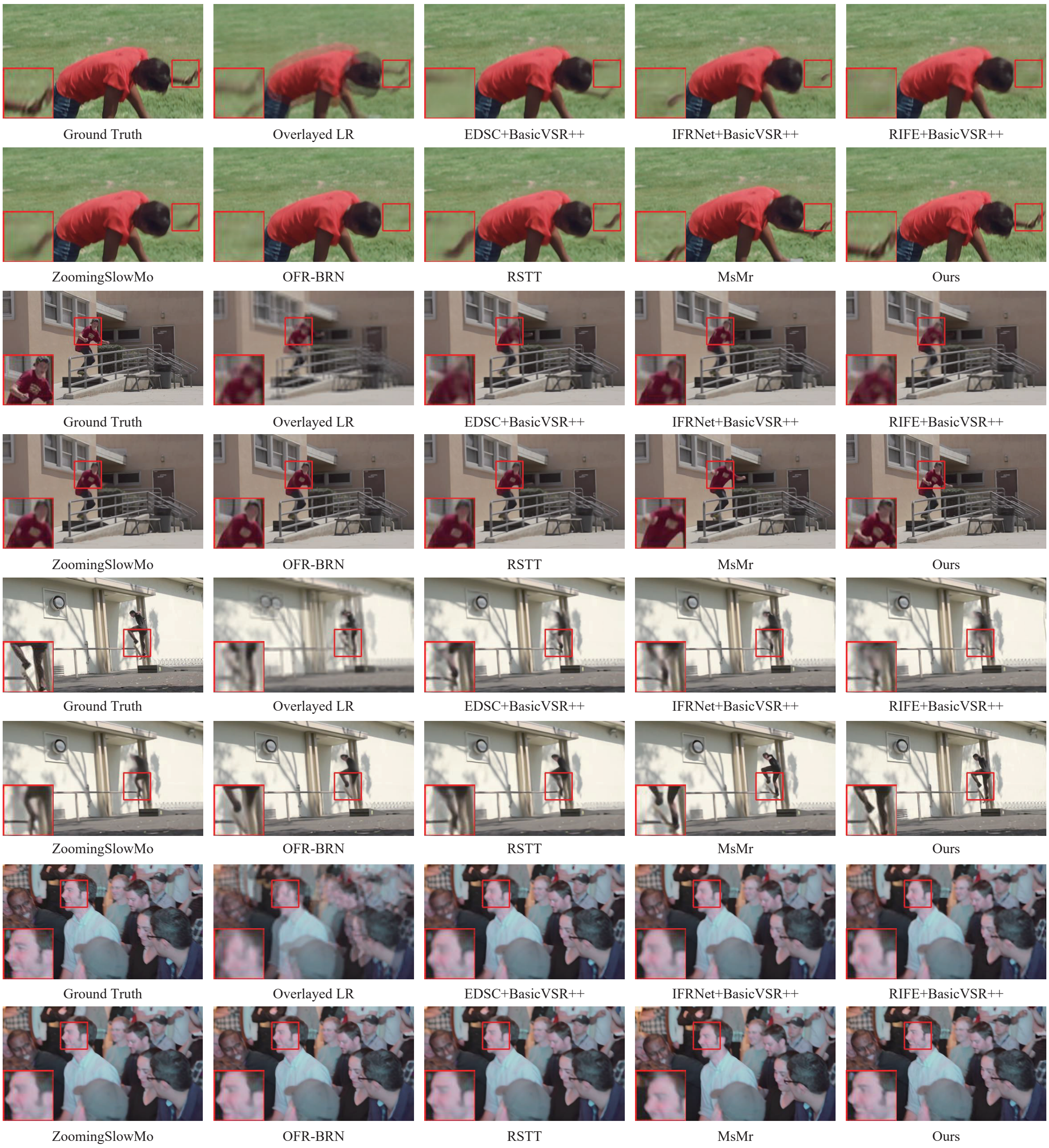
Ground Truth CAR + EDSR IRN AIDN

HCFlow SelfC MIMO-VRN Ours

Ground Truth CAR + EDVR IRN AIDN

HCFlow SelfC MIMO-VRN Ours

Figure 7. Visual qualitative comparisons on reconstruction of Space 4 × time 1 × case on Vid4 dataset.



Ground Truth

Overlayed LR

EDSC+BasicVSR++

IFRNet+BasicVSR++

RIFE+BasicVSR++

ZoomingSlowMo

OFR-BRN

RSTT

MsMr

Ours

Ground Truth

Overlayed LR

EDSC+BasicVSR++

IFRNet+BasicVSR++

RIFE+BasicVSR++

ZoomingSlowMo

OFR-BRN

RSTT

MsMr

Ours

Ground Truth

Overlayed LR

EDSC+BasicVSR++

IFRNet+BasicVSR++

RIFE+BasicVSR++

ZoomingSlowMo

OFR-BRN

RSTT

MsMr

Ours

Ground Truth

Overlayed LR

EDSC+BasicVSR++

IFRNet+BasicVSR++

RIFE+BasicVSR++

ZoomingSlowMo

OFR-BRN

RSTT

MsMr

Ours

Figure 8. Visual qualitative comparisons on reconstruction of Space 4 × time 2 × case on Vimeo90k dataset.

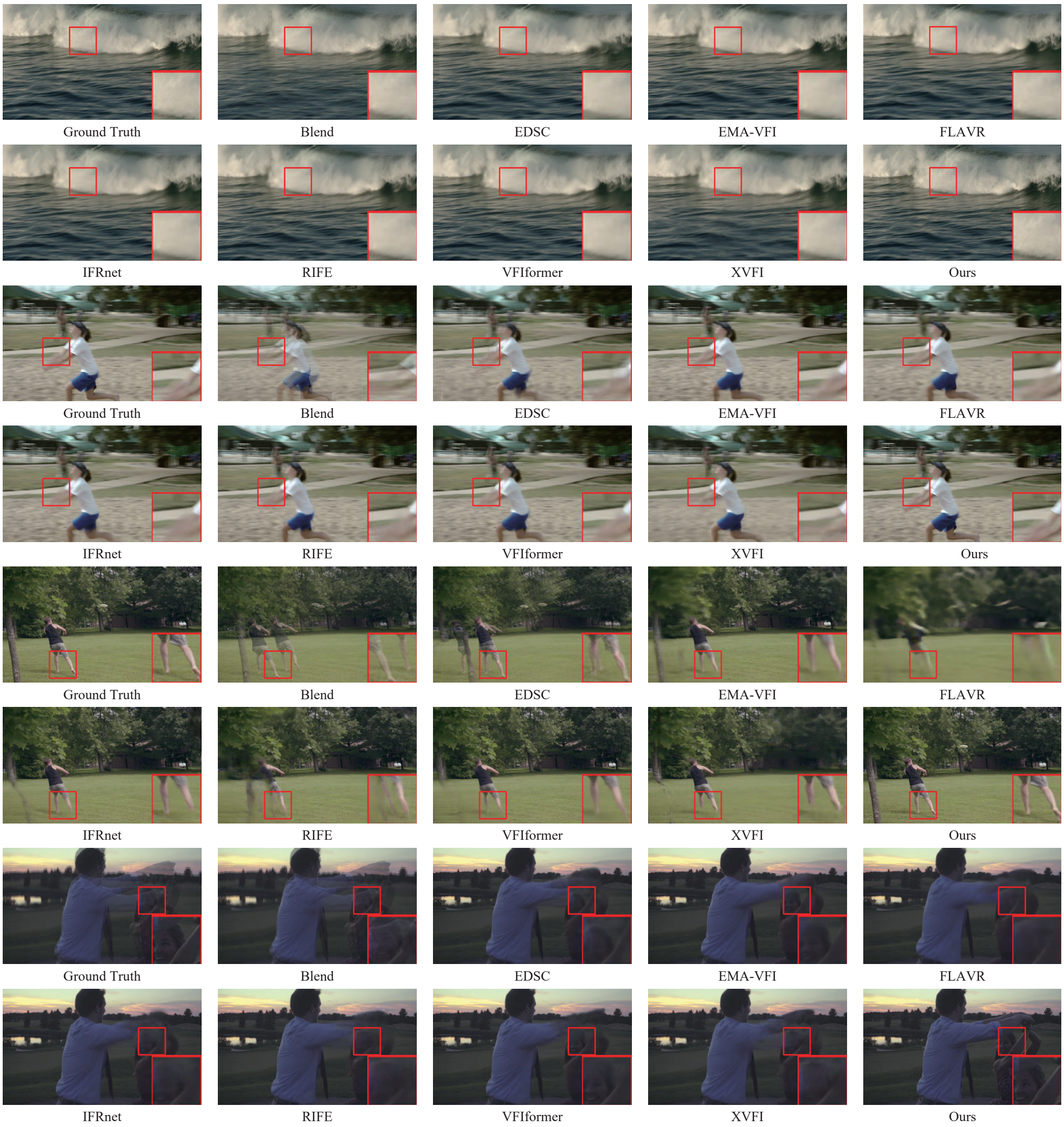


Figure 9. Visual qualitative comparisons on reconstruction of Space 1 × time 2 × case on Vimeo90k dataset.

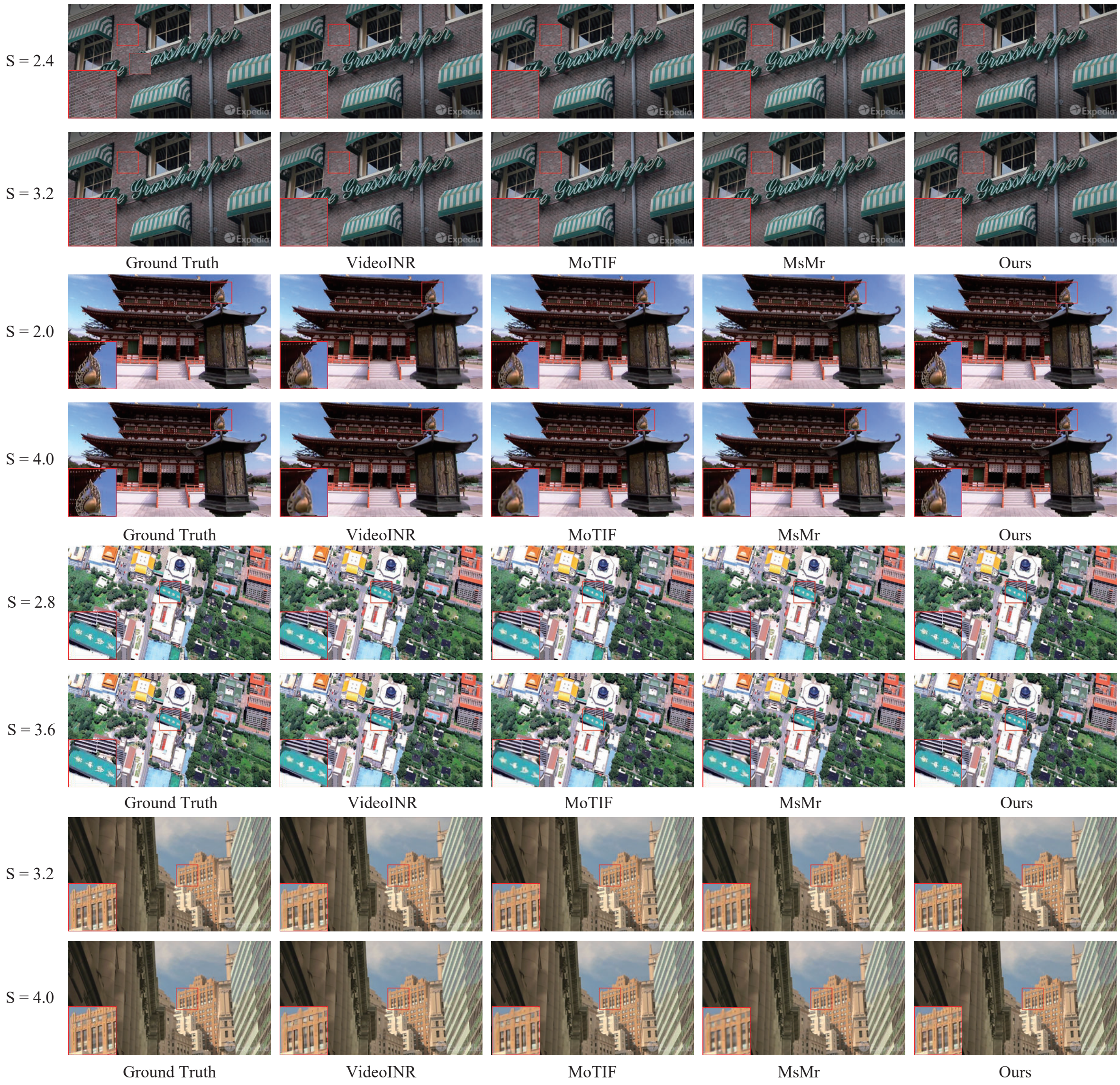


Figure 10. Visual qualitative comparisons of varying resampling fatcors for each method on the SPMCS dataset.

References

[1] 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.

[2] Yan-Cheng Huang, Yi-Hsin Chen, Cheng-You Lu, Hui-Po Wang, Wen-Hsiao Peng, and Ching-Chun Huang. Video rescaling networks with joint optimization strategies for downscaling and upscaling. In *Conference on Computer Vision and Pattern Recognition*, pages 3527–3536. 2

[3] Jun Li, Minqing Zhang, Ke Niu, and Xiaoyuan Yang. Investigation on principles for cost assignment in motion vector-based video steganography. *Journal of Information Security and Applications*, 73:103439, 2023. 6

[4] Jun Li, Minqing Zhang, Ke Niu, Yingnan Zhang, and Xiaoyuan Yang. Motion vector-domain video steganalysis exploiting skipped macroblocks. *arXiv preprint arXiv:2310.07121*, 2023. 6

[5] Shang Li, Guixuan Zhang, Zhengxiong Luo, Jie Liu, Zhi Zeng, and Shuwu Zhang. Approaching the limit of image rescaling via flow guidance. In *British Machine Vision Conference*, pages 1–13. 6

[6] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *International Conference on Computer Vision*, pages 4056–4065, 2021.

[7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.

[8] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *Conference on Computer Vision and Pattern Recognition*, pages 209–216, 2011. 1

[9] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 1

[10] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[11] 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.

[12] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020. 2

[13] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *International Conference on Computer Vision*, pages 4472–4480, 2017. 1

[14] Yuan Tian, Guo Lu, Xiongkuo Min, Zhaohui Che, Guangtao Zhai, Guodong Guo, and Zhiyong Gao. Self-conditioned probabilistic learning of video rescaling. In *International Conference on Computer Vision*, pages 4470–4479, 2021. 2, 5

[15] 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.

[16] Xiaoyu Xiang, Yapeng Tian, Vijay Rengarajan, Lucas D. Young, Bo Zhu, and Rakesh Ranjan. Learning spatio-temporal downsampling for effective video upscaling. In *European Conference on Computer Vision*, pages 162–181. 1, 3

[17] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *Conference on Computer Vision and Pattern Recognition*, pages 3370–3379, 2020. 1

[18] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *European Conference on Computer Vision*, pages 126–144. 2

[19] Jinbo Xing, Wenbo Hu, Menghan Xia, and Tien-Tsin Wong. Scale-arbitrary invertible image downscaling. *IEEE Transactions on Image Processing*, 32:4259–4274, 2023. 2, 5

[20] Kai Xu, Ziwei Yu, Xin Wang, Michael Bi Mi, and Angela Yao. Enhancing video super-resolution via implicit resampling-based alignment. In *Conference on Computer Vision and Pattern Recognition*, pages 2546–2555, 2024.

[21] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 1

[22] Xingyu Zhou, Leheng Zhang, Xiaorui Zhao, Keze Wang, Leida Li, and Shuhang Gu. Video super-resolution transformer with masked inter&intra-frame attention. In *Conference on Computer Vision and Pattern Recognition*, pages 25399–25408, 2024.