Compression-Aware Video Super-Resolution

Yingwei Wang[1*] Takashi Isobe[2*] Xu Jia[1†] Xin Tao[3]
Huchuan Lu[1,4] Yu-Wing Tai[5]
[1]Dalian University of Technology [2]Xiaohongshu Inc.
[3]Kuaishou Technology [4]Peng Cheng Laboratory
[5]The Hong Kong University of Science and Technology
{xjia, lhchuan}@dlut.edu.cn {isobetakashi}@xiaohongshu.com
{yingweiwangapr2000, jiangsutx, yuwing}@gmail.com

Abstract

Videos stored on mobile devices or delivered on the Internet are usually in compressed format and are of various unknown compression parameters, but most video super-resolution (VSR) methods often assume ideal inputs resulting in large performance gap between experimental settings and real-world applications. In spite of a few pioneering works being proposed recently to super-resolve the compressed videos, they are not specially designed to deal with videos of various levels of compression. In this paper, we propose a novel and practical compression-aware video super-resolution model, which could adapt its video enhancement process to the estimated compression level. A compression encoder is designed to model compression levels of input frames, and a base VSR model is then conditioned on the implicitly computed representation by inserting compression-aware modules. In addition, we propose to further strengthen the VSR model by taking full advantage of meta data that is embedded naturally in compressed video streams in the procedure of information fusion. Extensive experiments are conducted to demonstrate the effectiveness and efficiency of the proposed method on compressed VSR benchmarks. The codes will be available at https://github.com/aprBlue/CAVSR

1. Introduction

Video super-resolution aims at restoring a sequence of high-resolution (HR) frames by utilizing the complementary temporal information within low-resolution (LR) frames. There have been many efforts [1–3, 7, 13, 15–18, 20, 24, 35, 38, 47] made on this task, especially after the rise of deep learning. Most of these methods, however, assume an ideal input such as directly taking either bicubicly downsampled frames or Gaussian smoothed and decimated ones as the degraded inputs. In real world, videos stored on mobile devices or delivered on the Internet are all in a compressed format with different compression levels [10, 11, 14, 25, 30, 33, 41, 45]. Unless the compression is very lightweight, directly applying an existing VSR model would give unsatisfactory results with magnified compression artifacts, as shown in Fig. 1(a). One straightforward solution is to first apply a multi-frame decompression method [5, 9, 41, 45] to remove blocking artifacts and then feed the enhanced frames to an uncompressed VSR model. However, as shown in Fig. 1(b), the performance is still not good with artifacts remained. In addition, the decompression network usually cannot handle video frames of different compression levels adaptively, which will cause over-smoothing

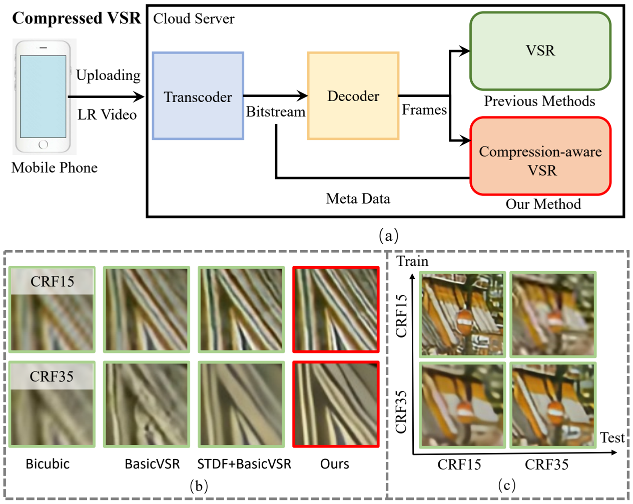


Figure 1. Motivation of this work. (a) Application scenario of the VSR task this work focuses on, (b) performance of existing VSR methods on compressed VSR task, and (c) performance of previous VSR models trained on videos of different compression.

[*] T. ISOBE and Y. WANG contributed equally to this work.
[†] Corresponding author.

and accumulate errors in super-resolution stage.

Recently a few pioneering works have been proposed to investigate the video super-resolution task on compressed videos. In [46], Yang et al. took into consideration complex degradations in real video scenarios and built a real-world VSR dataset using iPhone 11 Pro Max. A special training strategy is proposed to handle misalignment and illumination/color difference. In RealBasicVSR [3], Chan et al. synthesized real-world-like low-quality videos based on a combination of several degradations and proposed a two-stage method with an image pre-cleaning module followed by an existing VSR model. COMISR [22] and FTVSR [27] are proposed to address streamed videos compressed in different levels rather than degradations like noise and blur.

Although COMISR and FTVSR have improved performance on compressed videos, they are not specially designed to deal with videos of various levels of compression. Being aware of compression with input videos would allow a model to exert its power on those videos adaptively. Otherwise, video frames with less compression would be oversmoothed while the ones with heavy compression would still remain magnified artifacts, as shown in Fig. 1(c). These methods feed themselves with only compressed video frames as input, however, meta data such as frame type, motion vectors and residual maps that are naturally encoded with a compressed video are ignored. Making full use of such meta data and the decoded video frames could help further improve super-resolution performance on compressed videos.

Based on the above observations, we propose a *compression-aware video super-resolution* model, a compression encoder module is designed to implicitly model compression level with the help of meta data of a compressed video. It would also take into account both frames and their frame types in computing compression representation. A base bidirectional recurrent-based VSR model is then conditioned on that representation by inserting compression-aware modules such that it could adaptively deal with videos of different compression levels. To further strengthen the power of the base VSR model, we take advantage of meta data in a further step. Motion vectors and residual maps are employed to achieve fast and accurate alignment between different time steps and frame types are leveraged again to update hidden state in bidirectional recurrent network. Extensive experiments demonstrate that the specially designed VSR model for compressed videos performs favorably against state-of-the-art methods.

Our contributions are summarized as follows:

- A compression encoder to perceive compression levels of frames is proposed. It is supervised with a ranking-based loss and the computed compression representation is used to modulate a base VSR model.
- Meta data that comes naturally with compressed videos are fully explored in fusion process of spatial and temporal information to strengthen the power of a bidirectional RNN-based VSR model.
- Extensive experiments demonstrate the effectiveness and efficiency of the proposed method on compressed VSR benchmarks.

2. Related work

2.1. Video Super-Resolution

Most existing state-of-the-art VSR methods work on videos without much compression artifacts. They can be divided into two categories according to the way of temporal information aggregation: sliding-window-based and recurrent-based methods. Sliding-window-based methods [34, 35, 38, 42] compute optical flow or use deformable convolutions to align neighboring frames on either pixel-level or feature-level. Some methods [15, 18, 21, 48] also perform VSR without explicit motion compensation by using carefully designed networks based on 3D convolutions or non-local based modules. Recurrent-based VSR methods could exploit long-range temporal information accumulated over time, working in an online fashion. Uni-directional recurrent-based VSR methods [7, 12, 31] propagate history information to the current time step for restoration, working in an online fashion; while BasicVSR [1], GOVSR [47] and BasicVSR++ [2] aggregate temporal information from two directions for performance improvement. All these sliding-window-based and recurrent-based methods are trained and evaluated on bicubicly downsampled video frames without considering compression. In this work, we build our work on top of a bidirectional VSR model and achieve compression-aware VSR by modeling compression with the proposed compression encoder.

2.2. Compressed Video Quality Enhancement

There have been several attempts made to enhance the quality of compressed videos by removing the artifacts introduced by the codec. MFQE [9, 45] presents a module for peak quality frames (PQFs) detection and uses a lightweight multi-frame CNN to enhance other low-quality frames. It also alleviates the problem of quality fluctuation over frames with those single-frame-based methods [43, 44]. To reduce the influence of inaccurate optical flow, STDF [5] incorporates a spatial-temporal deformable convolution to aggregate temporal information. TSAN [41] aims at transcoded video restoration and achieves it using temporal deformable alignment and pyramidal spatial fusion.

Super-resolving compressed frames is a more challenging task, which requires both artifact removal and detail enhancement. A few pioneering works have made attempts at this task. COMISR [22] presents a bi-directional VSR

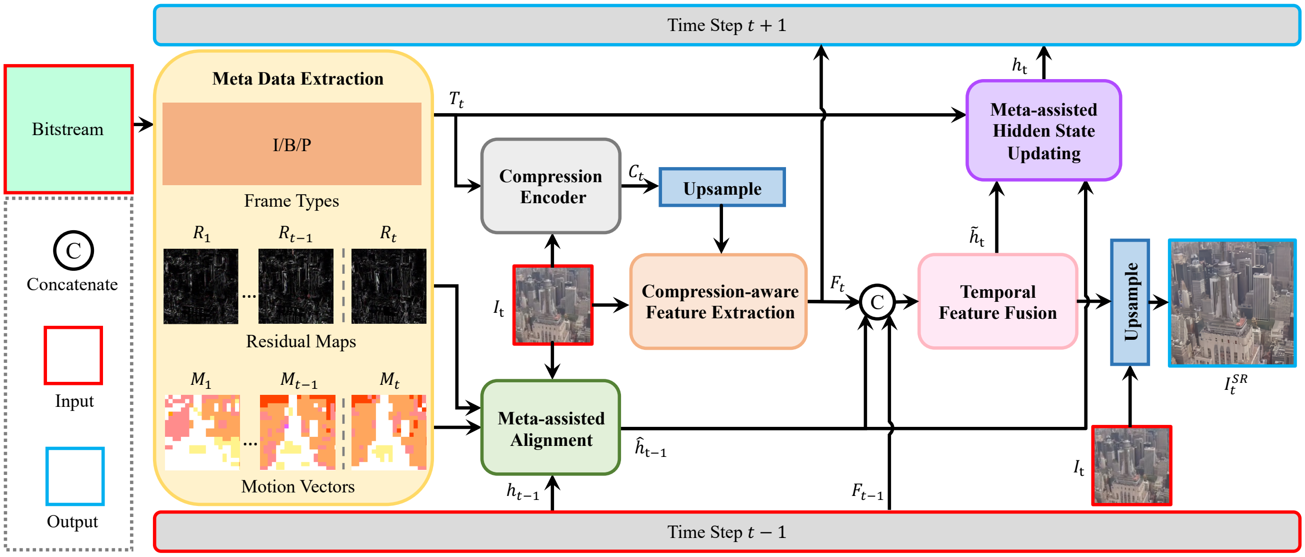


Figure 2. Pipeline of the proposed compression-aware VSR model. From the meta data of bitstream, we extract the frame types, motion vectors and residual maps. These additional information will be processed by our compression encoder to upsample the features of the current frame. Using the motion vector and residual maps, the meta-assisted alignment module aggregate information from the previous frame and merge it with the SR features of the current frame by the temporal feature fusion module. Finally, we obtain the SR results through the upsampling decoder. The meta data and the aggregated SR features of the current frame are also used to update the hidden state to assist the SR process of the next frame.

framework that include a detail-aware flow estimation module to recover HR flow, and a Laplacian enhancement module to add high-frequency details. FTVSR [27] projects video frames into the frequency domain and designs a Frequency-Transformer that conducts self-attention in joint space-time-frequency space to recover high-frequency details. RealBasicVSR [3] focuses on video shooting scenarios and takes into account noise and blur in addition to compression in their setting. A specially designed pre-cleaning module is added in front of BasicVSR for detail enhancement and artifact suppression. However, these methods are not specially designed to deal with input videos of various compression levels. In addition, rich metadata that is encoded in the bitstream could benefit the super-resolution process but has not been fully exploited. Very recently, CIAF [49] directly takes MVs to approximate optical flow for motion compensation and employs residual map to compute a mask for adaptive inference, which gives an initial attempt in employing metadata in the VSR task.

In this work, we focus on leveraging metadata to perform compression-aware video super-resolution, which is achieved by conditioning a base VSR model on an implicitly computed compression representation.

3. Method

In this work, we design a compression encoder to model compression of a frame in an implicit way and train it in a ranking-based manner. The computed compression representation is then employed to modulate a bidirectional recurrent-based VSR model such that it is able to adaptively deal with various compressions. To further strengthen the VSR model, metadata that is naturally encoded with a compressed video is fully explored to improve alignment between frames and hidden state update in bidirectional recurrent network. The overall architecture of the proposed framework could be found in Fig. 2.

3.1. Preliminary

We begin by briefly outlining several basic concepts about video codecs. Modern video encoders adopt Group of Pictures (GOP) as basic structure, which includes three different frame types: I-frames, P-frames and B-frames. Typically, I-frame (Intra-coded picture) is less compressed and its compression is similar to standalone image compression. P-frame (Predicted picture) holds only change in the image from the previous frame. Therefore, the encoder does not need to store the unchanging background pixels in P-frame, thus saving more space. B-frame (Bidirectional predicted picture) saves even more space by using the difference from both the preceding and following frames to encode its content. Block-wise motion vectors (MVs) and residual maps are stored in P- and B-frames for motion compensation and decoding. Obviously, even for the same video, different types of frames contain different levels of information and

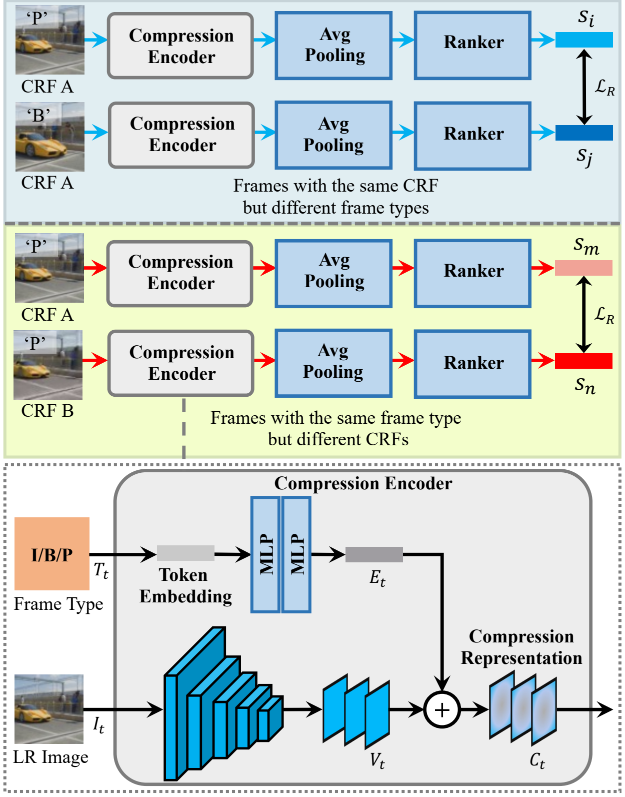


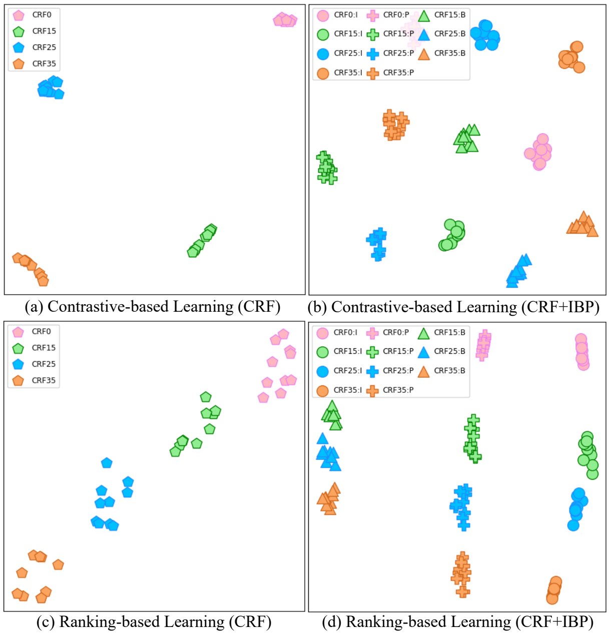
Figure 3. Compression encoder and its training strategy. With our pairwise training strategy and learning-to-rank paradigm, subtle compression differences can be learned effectively which allows our method to be sensitive to different compression levels.

compression.

Modern codecs also allow users to adjust compression levels by tuning its perceptual quality. A common way is to tune the Constant Rate Factor (CRF), which is a number in [0, 51]. CRF0 denotes lossless encoding and larger number means heavier compression and worse perceptual quality. Note that although we focus on the use of H.264 [40] in this work, the proposed method also applies to other codecs such as H.265 [32] and AVS [8].

3.2. Compression Encoder

To make the VSR model adaptive to various compression, we design a compression encoder to implicitly model compression in a video frame by taking into account both frame type and compressed perceptual quality. However, making an encoder sensitive to compression level is difficult because the differences across frame types and very closed CRF is very subtle. Thus, in this work compression representation learning is treated as a learning-to-rank task, as shown in Fig. 3. Specifically, video frame pairs are prepared in two ways in terms of compression. One subset is composed of frame pairs with the same CRF but with different frame types, and the other subset is composed of frame pairs with the same frame type but with different CRFs. The compression encoder is taught to become aware of the compression level of different frame types from the former subset and learn to distinguish compression caused by a small CRF from a large one.



(a) Contrastive-based Learning (CRF)

(b) Contrastive-based Learning (CRF+IBP)

(c) Ranking-based Learning (CRF)

(d) Ranking-based Learning (CRF+IBP)

Figure 4. Visualization of compression representation learned with different strategies using t-SNE [36]. Ten videos are compressed with a set of different CRFs. All compressed videos include I-, B- and P-frames, except that those compressed with CRF0 include only I- and P- frames.

It consists of two input branches, *i.e.*, frame type branch and frame branch. As for the frame type branch, a one-hot vector is assigned to each frame type and a token embedding is used to represent that information. As for the frame branch, the frame decoded from a video codec is fed to a few convolutional layers. The feature maps from the frame branch and the token embedding from the frame type branch are combined as the compression representation of this frame, which is denoted as $C_t \in \mathbb{R}^{\mathbb{C}}$.

To teach the encoder to be compression-aware, a pair of frames and their frame types are sent to a siamese-like architecture. In that architecture, a pair of compression representations are obtained from the shared compression encoder and two scores are further computed after a few shared ranker layers. The ranker predicts a ranking score s for an LR frame. The ranking loss [50] is applied to learn the ranking orders about compression level. For convenience, we define a score $Q_f = \{0, 1, 2\}$ for each frame type $\{I, P, B\}$ according to the amount of compression, and define another score $Q_c = CRF\ number$ for the other kind

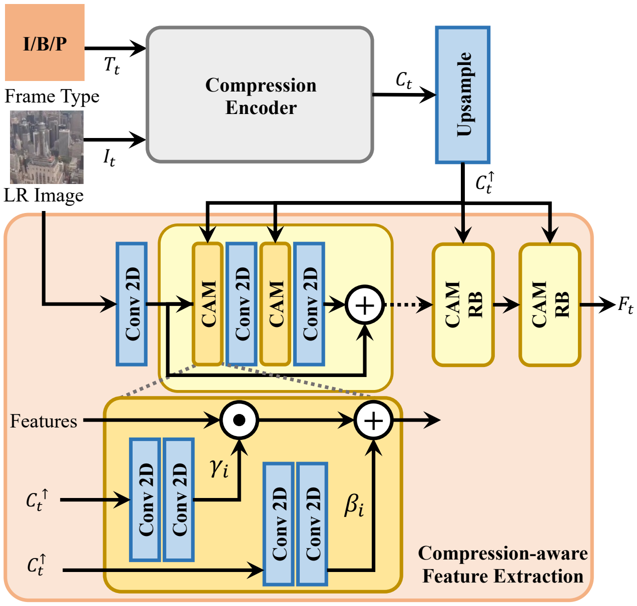


Figure 5. Illustration of compression-aware feature extraction.

of compression factor. The pairwise margin-ranking loss is adopted and is defined below.

$$\mathcal{L}_R = max(0, (s_i - s_j) * \kappa + \xi)$$
$$where \begin{cases} \kappa = 1 & if\ Q_{f/c}(i) < Q_{f/c}(j) \\ \kappa = -1 & if\ Q_{f/c}(i) > Q_{f/c}(j), \end{cases} \tag{1}$$

where κ indicates the ground-truth order between the pair, ξ is the margin, Q_f or Q_c is chosen depending on the used subset. By encouraging the ranking scores to have the correct order, degradation representation learns to encode information about degradation within it, which is the key to achieving compression-aware video super-resolution.

3.3. Compression-Aware VSR

Compression-aware modulation. Once compression representation is computed, it is employed to encourage a base VSR model to perform adaptively under various compression. Feature extraction component of the base VSR model is composed of several convolutional layers and residual blocks. In this work, a simple compression-aware modulation (CAM) module is inserted before each convolutional layer in the feature extraction process. Similar to [39], the modulation is instantiated as an affine transformation whose parameters γ_i and β_i are computed spatially adaptively based on the compression representation, as shown in Fig. 5 and Eq. 2.

$$\text{CAM}(\mathbf{F}|\gamma_i, \beta_i) = \gamma_i \odot \mathbf{F} + \beta_i, \tag{2}$$

where, $\mathbf{F}$ represents the features generated by the i-th convolutional layer. The spatial dimension of γ_i and β_i are the same as $\mathbf{F}$, and $\odot$ denotes element-wise multiplication. In this way, estimated compression information is injected into the base VSR model to achieve compression-aware video super-resolution.

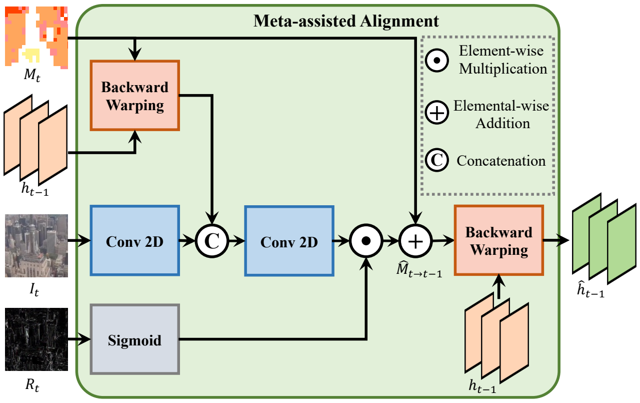


Figure 6. Illustration of the proposed meta-assisted alignment.

Meta-assisted alignment. Motion compensation plays a key role in the VSR process. A common practice is to estimate optical flow between frames and apply that in image or feature alignment. However, computing optical flow would bring in extra computation and inaccurate optical flow prediction would have negative influence on VSR performance. Optical flow estimation is challenging especially when video is seriously compressed. In this work, we make full use of two kinds of additional meta data that naturally come with compressed videos, *i.e.*motion vectors (MVs) and residual maps, in the alignment procedure. Directly taking MVs as an alternative of optical flow might not be optimal since they are computed block-wisely in the video codec. As shown in Fig. 6, here we take MVs as initial offsets and further refine them with the help of the input frame and the residual map. The estimated motion information is used to align hidden states to the current time step in the temporal feature fusion stage. Warped hidden state representation is then combined with frame feature to compute residual offsets. In the codec process, extracted residual maps represent how well the reconstruction is. An attention map is computed based on the residual map in order to refine motion represented by MVs. Regions with larger reconstruction errors imply larger alignment errors and rely more on refinement. The final motion $\hat{M}$ is computed as the sum of MVs M and the estimated residual offsets.

Meta-assisted Propagation. Since content in B-frame is heavily compressed, the hidden state computed for that frame may contain less information than others, hence causing performance degradation in the propagation process over time. To ameliorate this issue, we propose to update the hidden state of B-frame as a momentum-based moving

Table 1. Quantitative comparison on compressed video of Vid4 [23] for 4× VSR. The PSNR (dB) and SSIM are calculated on Y-channel. Red text indicates the best and blue text indicates the second best performance. The Runtime is calculated on an HR image size of 1280 × 720. These model are carefully trained using the provided code.

Method	Params	Runtime	Per clip with Compression CRF25				Average of clips with Compression		
	(M)	(ms)	calendar	city	foliage	walk	CRF15	CRF25	CRF35
EDVR [38]	20.6	378	21.76/0.676	25.74/0.665	24.12/0.624	27.42/0.811	26.53/0.794	24.76/0.694	22.39/0.544
IconVSR [1]	8.7	70	21.45/0.661	25.76/0.659	24.25/0.629	26.99/0.804	26.84/0.809	24.61/0.688	22.10/0.520
BasicVSR++ [2]	7.3	77	22.31/0.693	26.24/0.670	24.53/0.635	27.08/0.830	27.17/0.826	25.04/0.707	22.21/0.532
RealBasicVSR [3]	6.3	63	21.86/0.683	25.85/0.672	24.22/0.631	27.55/0.819	26.94/0.813	24.87/0.701	22.39/0.531
STDF [5]+BasicVSR [1]	7.0	95	22.20/0.682	25.93/0.658	24.60/0.633	27.15/0.819	26.82/0.805	24.97/0.698	22.52/0.540
COMISR [22]	6.2	73	22.87/0.718	25.95/0.657	24.72/0.662	27.02/0.814	26.66/0.801	25.14/0.713	22.62/0.546
FTVSR [27]	10.8	850[a]	23.09/0.745	26.43/0.693	25.07/0.659	27.45/0.831	27.50/0.826	25.51/0.732	22.79/0.561
Ours	8.9	93	22.99/0.747	26.48/0.709	24.92/0.668	28.20/0.842	27.42/0.833	25.65/0.742	22.84/0.574

[a] Since FTVSR takes much GPU memory, we follow a similar way as mentioned in the original paper and divide an image into 4 patches and pass each patch through the model. Therefore the final runtime is the total time of inference on all these patches.

average with its previous frame, as shown in Eq. 3.

$$\begin{cases} h_t = \alpha * \tilde{h}_t + (1-\alpha) * \hat{h}_{t-1} & \text{if } T_t = B \\ h_t = \tilde{h}_t & \text{otherwise,} \end{cases} \quad (3)$$

where, $\tilde{h}_t$ is the hidden state estimated in current time step t. $\hat{h}_{t-1}$ means the hidden state which is calculated in the previous time step $t-1$ and aligned to current time step t by using MVs. α means a momentum coefficient. We find that $\alpha = 0.5$ is a optimal value for different levels of compression.

4. Experiments

4.1. Implementation Details

Compressed Datasets. In this work, we use the popular Vimeo-90k dataset [42] for training. The training set consists of about 65K 7-frame video clips with various motion types. To generate the compressed LR frames, the HR frames are first smoothed by a Gaussian kernel with a standard deviation of 1.5 and downsampled by a scale of 4. Then, we use the H.264 encoder to generate the compressed video with popular FFmpeg 4.3 [6]. We set the CRF to 0, 15, 25, and 35, following [22] and [27]. To write out the bitstream, we modified the decoder of FFmpeg and will release the code publicly to encourage others to work with compressed video. Finally, we feed compressed frames as LR sequences and corresponding meta data to the VSR models to obtain super-resolution results. Following [22] and [27], we adopt the Vid4 dataset [23] as our test set and compress it with the same degradation and compression method as Vimeo-90K. The SR results are evaluated in terms of PSNR and SSIM on the Y channel of YCbCr space. Moreover, we further evaluate our method on the downloaded Vid4 from YouTube to simulate more realistic video streaming scenario, similar to [22].

Training Setting. The proposed CAVSR has 5 compression-aware modulation residual blocks (CAMRB) for compression-aware feature extraction and 25 residual blocks for temporal feature fusion. The batch size and the patch size of LR images are set to 16 and 64 × 64 during training. We also use random rotation, flipping and temporal reverse operation as the data augmentation technique during training to avoid overfitting. In our experiments, we set $\xi = 0.5$ and $\alpha = 0.5$. The overall network is trained in two stages The Cosine Annealing scheme [26] and Adam optimizer [19] with $\beta_1 = 0.9$ and $\beta_1 = 0.999$ are used. In stage one, we only train the compression encoder and ranker by optimizing Eq. 1 for 100K iterations. The initial learning rate is set to 1×10^{-4}. In stage two, we freeze the compression encoder and train the reset parts which are supervised by Charbonnier penalty loss function [4]. The initial learning rate is set to 1×10^{-4}. The total number of iterations is 400K. During training, videos with CRF0 and compressed videos with CRF15/25/35 are randomly fed to the VSR model with a probability of 0.5. All experiments are implemented with PyTorch on a server with V100 GPUs.

4.2. Comparison with State-of-the-Arts

In this section, we compare our method with several state-of-the-art VSR approaches, including EDVR [38], IconVSR [1], BasicVSR++ [2], RealBasicVSR [3], STDF [5] + BasicVSR [1], COMISR [22], FTVSR [27]. The first three methods are VSR methods originally proposed to deal with ideal degradation, *e.g.,* bicubic or Gaussian bicubic. The last five are the methods of working on compressed videos. The results of these methods are obtained by carefully training on our training set. Our method outperforms most of the previous VSR methods on the three compression levels both in PSNR and SSIM. Our method obtains comparable performance with latest FTVSR model. However, its model size is larger and its speed is much slower than ours because of its heavy computation and GPU

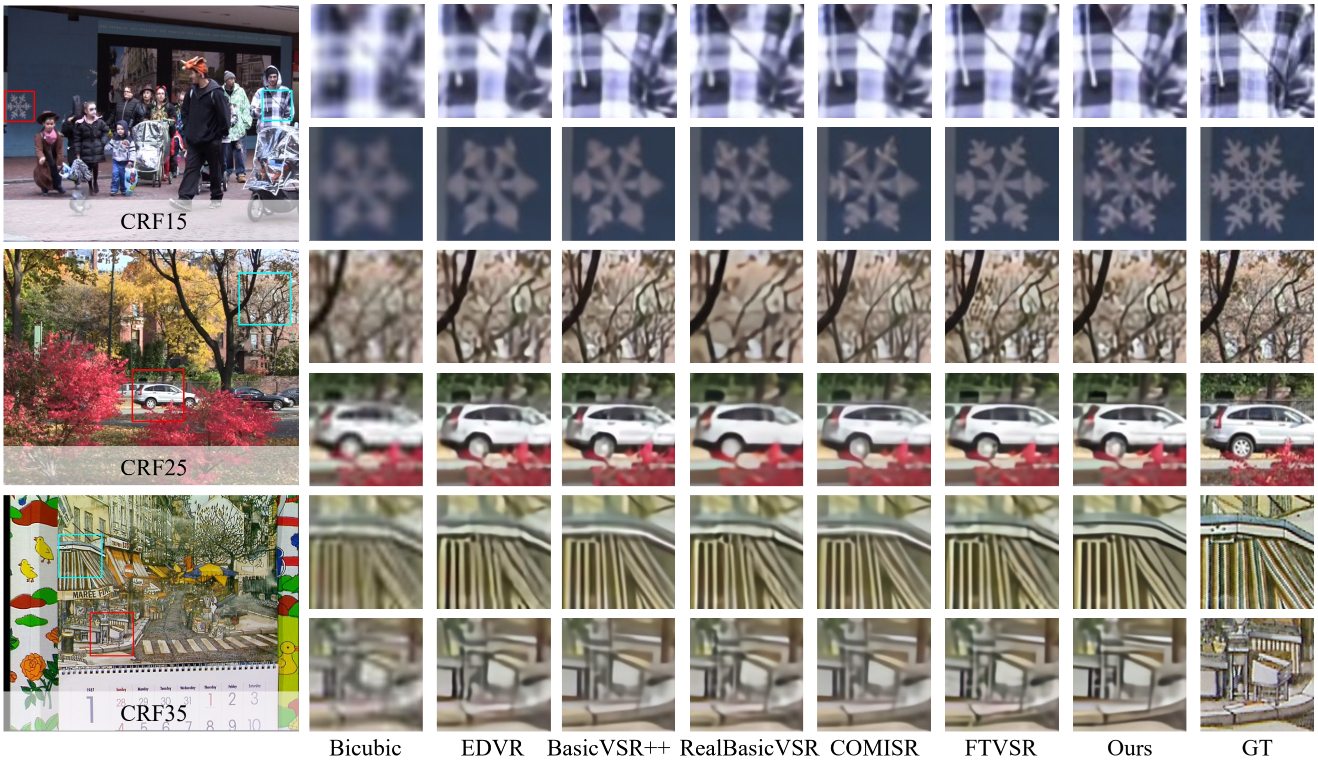


Figure 7. Qualitative comparison on the compressed **Vid4 [23]** test set for 4× VSR. Zoom in for better visualization.

memory usage. The qualitative comparison with other state-of-the-art methods is shown in Fig. 7. Our method produces higher quality HR image including finer details and sharper edges. Other methods are either prone to generate some artifacts (e.g., wrong stripes on clothes) or can not recover missing details (e.g., small windows of the building).

Experiments on real-world compressed videos. We also evaluate these methods on real-world compressed videos following [22]. Uncompressed videos are first generated from the raw frames, and are then uploaded YouTube. Various VSR models are applied to the downloaded videos for evaluation. Fig. 8 shows that the proposed method gives visually better results.

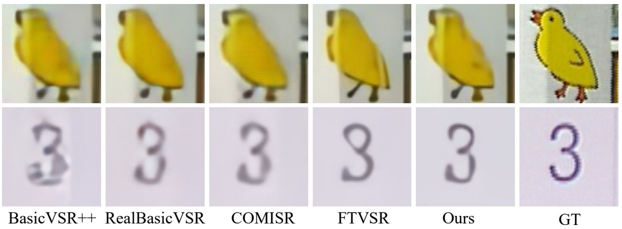


Figure 8. Qualitative comparison on Vid4 [23] downloaded from YouTube for × 4 VSR.

Table 2. Ablation studies of the proposed CAVSR on the compressed Vid4 with CRF15, CRF25 and CRF35. 'CAM' means the proposed compression-aware modulation. 'MA' and 'MH' represents the proposed meta-data-assisted alignment and hidden state updating, respectively.

Model #	CAM	MA	MH	CRF15	CRF25	CRF35
1				26.76	24.54	22.06
2	✓			27.25	25.41	22.74
3	✓	✓		27.40	25.60	22.80
4	✓	✓	✓	27.42	25.65	22.84

4.3. Ablation Study

Ablation on the components of CAVSR. In this section, we examine the effectiveness of each component of the proposed CAVSR on the compressed Vid4, as shown in Tab. 2. We adopt the BasicVSR [1] framework as our baseline (Model 1), which achieves 26.76dB, 24.54dB and 22.06dB on CRF 15, 25 and 35, respectively. By inserting the compression-aware modules before each convolutional layer in the feature extraction, Model 2 surpasses Model 1 by +0.49dB, +0.87dB and +0.68dB on CRF 15, 25 and 35, respectively. Such improvement could be attributed

Table 3. Ablation studies of the proposed CAM with different compression-aware representations. 'CL' and 'RL' means the training of compression-aware encoder with contrastive loss and ranking loss, respectively. 'CRF' represents the training data using the different kinds of CRF but the same frame types. 'IBP' is the training data using different kinds of frame types but the same CRF.

Model #	Loss		Data		CRF15	CRF25	CRF35
	CL	RL	CRF	IBP			
5					26.98	24.74	22.17
6	✓		✓		27.20	25.49	22.62
7	✓		✓	✓	27.32	25.58	22.77
8		✓	✓		27.27	25.54	22.69
9		✓	✓	✓	27.42	25.65	22.84

to the compression-aware design which allows the model to exert its power adaptively on different videos. With the proposed meta-assisted alignment (Model 3), we obtain extra performance gains, *e.g.,* from 27.25/25.41/22.74 to 27.40/25.60/22.80. By using the meta-assisted propagation, the performance of Model 4 further increases to 27.42/25.65/22.84. The performance gain of Model 3 and Model 4 implies that making full use of the information from meta data is beneficial to VSR.

Ablation on the compression-aware modulation. We try two kinds of the losses for compression encoder, 'CL' denotes the contrastive learning loss in [37] while 'RL' denotes the proposed ranking loss. We also use two kinds of input data for training the compression encoder. (1) 'CRF' means the input frame pair have different CRF level but the same frame types. (2) 'IBP' means the input frame pair are the different frame types but the same CRF level. Tab. 3 shows the results of using compression-aware representations obtained with two different losses in the modulation. Specifically, the compression encoder is supervised by either contrastive loss or the proposed ranking loss. Model 5 is a base model with the proposed meta-assisted alignment and propagation modules. We observe that as long as the compression-aware modulation is conducted, the VSR performance would be improved no matter whether the compression encoder is trained with contrastive loss or ranking loss. But we also notice that different kinds of learning strategies for compression representation do have an influence on performance. Improvements from Model 8 to Model 9 show that informing the model about the ranking order is more helpful in compression-aware modulation than only teaching it to distinguish one from the other. We also find that using more diverse input pairs can improve the performance, *e.g.,* Model 6 vs. Model 7 and Model 8 vs. Model 9.

Table 4. Ablation studies of the proposed Meta-assisted Alignment. 'OF', 'MV' and 'RMV' represents the Optical Flow [28], Motion Vector and the proposed Refined Motion Vector, respectively. The Runtime(ms) is calculated on an HR image size of 1280×720.

OF	MV	RMV	#Param.	Runtime	CRF15	CRF25	CRF35
✓			9.9M	114	27.53	25.69	22.74
	✓		8.5M	87	27.36	25.57	22.79
	✓	✓	8.9M	93	27.42	25.65	22.84

Ablation on the meta-assisted q1. We examine our method with different alignment methods, as shown in Tab. 4. We adopt the SPyNet [29] as an alternative way to perform alignment at the feature level. We can observe that the optical flow indeed boosts the performance on CRF 15 and CRF 25 by a large margin, while it suffers a performance drop on larger compression level, *i.e.,* CRF 35. The explanation for that is the larger compression level would result in severe blocking in the frame which would influence the accuracy of optical flow estimation. The model with optical flow still comes with considerable computational expense. Compared to that, MVs are not only cheap to obtain but also bring consistent improvement on all testing CRFs. Although such block-wise alignment is not accurate as the pixel-wised optical in CRF 15 and CRF 25, the proposed refined MVs with minimal computation increase could help achieve comparable performance with the optical flow.

5. Conclusion

In this work, we focus on addressing the task of video super-resolution on compressed videos. A novel compression-aware video super-resolution approach is proposed to deal with various compression. The key of this method is the design of a compression encoder, which can implicitly model compression in a video frame into a compression representation. The learning of compression representation is treated as a learning-to-rank task on the constructed pairs. A compression-aware modulation modules conditioned on that representation are inserted to base VSR models to achieve compression-aware VSR. Metadata such as MVs and residual maps, which are naturally encoded in the video, are leveraged in motion compensation and propagation modules to further improve the VSR performance. Extensive experiments demonstrate the effectiveness of the proposed method.

Acknowledgement The research was partially supported by the Natural Science Foundation of China, No. 62106036, and the Fundamental Research Funds for the Central University of China, DUT21RC(3)026.

References

[1] 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*, 2021. 1, 2, 6, 7

[2] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 2022. 1, 2, 6

[3] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *CVPR*, 2022. 1, 2, 3, 6

[4] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 6

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

[6] FFmpeg. FFmpeg h.264 video encoding guide. In *https://trac.ffmpeg.org/wiki/Encode/H.264*. 6

[7] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCVW*, 2019. 1, 2

[8] Wen Gao, Siwei Ma, Li Zhang, Li Su, and Debin Zhao. AVS video coding standard. In *Intelligent Multimedia Communication: Techniques and Applications*, volume 280 of *Studies in Computational Intelligence*, pages 125–166. Springer, 2010. 4

[9] 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

[10] Takashi Isobe, Jian Han, Fang Zhuz, Yali Liy, and Shengjin Wang. Intra-clip aggregation for video person re-identification. In *ICIP*, 2020. 1

[11] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuang Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning. In *CVPR*, 2021. 1

[12] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *ECCV*, 2020. 2

[13] Takashi Isobe, Xu Jia, Xin Tao, Changlin Li, Ruihuang Li, Yongjie Shi, Jing Mu, Huchuan Lu, and Yu-Wing Tai. Look back and forth: Video super-resolution with explicit temporal difference modeling. In *CVPR*, 2022. 1

[14] Takashi Isobe, Dong Li, Lu Tian, Weihua Chen, Yi Shan, and Shengjin Wang. Towards discriminative representation learning for unsupervised person re-identification. In *ICCV*, 2021. 1

[15] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *CVPR*, 2020. 1, 2

[16] Takashi Isobe, Fang Zhu, Xu Jia, and Shengjin Wang. Revisiting temporal modeling for video super-resolution. In *BMVC*, 2020. 1

[17] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Turning frequency to resolution: Video super-resolution via event cameras. In *CVPR*, 2021. 1

[18] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 1, 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICIP*, 2014. 6

[20] Suyoung Lee, Myungsub Choi, and Kyoung Mu Lee. Dynavsr: Dynamic adaptive blind video super-resolution. In *WACV*, 2021. 1

[21] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, 2020. 2

[22] Yinxiao Li, Pengchong Jin, Feng Yang, Ce Liu, Ming-Hsuan Yang, and Peyman Milanfar. Comisr: Compression-informed video super-resolution. In *ICCV*, 2021. 2, 6, 7

[23] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):346–360, 2013. 6, 7

[24] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In *ICCV*, 2017. 1

[25] Yifan Liu, Yali Li, and Shengjin Wang. Disentangling based environment-robust feature learning for person reid. In *BMVC*, 2022. 1

[26] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *CoRR*, abs/1608.03983, 2016. 6

[27] Zhongwei Qiu, Huan Yang, Jianlong Fu, and Dongmei Fu. Learning spatiotemporal frequency-transformer for compressed video super-resolution. In *ECCV*, 2022. 2, 3, 6

[28] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 8

[29] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 8

[30] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G Anderson, and Lubomir Bourdev. Learned video compression. In *ICCV*, 2019. 1

[31] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, 2018. 2

[32] Gary J. Sullivan, Jens-Rainer Ohm, Woojin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *TCSVT*, 22(12):1649–1668, 2012. 4

[33] Yu-Wing Tai, Hao Du, Michael S Brown, and Stephen Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *TPAMI*, 32(6):1012–1028, 2009. 1

[34] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, 2017. 2

[35] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020. 1, 2

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(86):2579–2605, 2008. 4

[37] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *CVPR*, 2021. 8

[38] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPR*, 2019. 1, 2, 6

[39] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 5

[40] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *TCSVT*, 13(7):560–576, 2003. 4

[41] Li Xu, Gang He, Jinjia Zhou, Jie Lei, Weiying Xie, Yunsong Li, and Yu-Wing Tai. Transcoded video restoration by temporal spatial auxiliary network. In *AAAI*, 2022. 1, 2

[42] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 2, 6

[43] Ren Yang, Mai Xu, Tie Liu, Zulin Wang, and Zhenyu Guan. Enhancing quality for HEVC compressed videos. *TCSVT*, 29(7):2039–2054, 2019. 2

[44] Ren Yang, Mai Xu, and Zulin Wang. Decoder-side hevc quality enhancement with scalable convolutional neural network. In *ICME*, 2017. 2

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

[46] Xi YANG, Wangmeng Xiang, Hui Zeng, and Lei Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *ICCV*, 2021. 2

[47] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, Tao Lu, Xin Tian, and Jiayi Ma. Omniscient video super-resolution. In *ICCV*, 2021. 1, 2

[48] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, pages 3106–3115, 2019. 2

[49] 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*, 2022. 3

[50] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *ICCV*, 2019. 4