Compression-Omniscient Video Super-Resolution

Shuyun Wang, Yanbin Liu, Ming Lu, Zhuojie Wu, Senmao Tian, Yandong Guo, Xin Yu

***Abstract*—Current compressed video super-resolution methods have achieved promising performance but they often assume that an input video is compressed under low-delay configurations. However, under random access configurations, those methods might struggle to leverage the metadata effectively due to the large variations of metadata in different compression configurations. In this work, we propose a general Compression-Omniscient Video Super-Resolution (COVSR) method that can address video super-resolution for both low-delay and random-access configurations. Specifically, we first introduce an efficient compression-aware propagation (ECAP) module that dynamically adjusts propagation routes in accordance with the compression configurations. Since existing methods require reconstructing frames in a frame-by-frame manner, it is difficult to achieve efficient parallelization. However, we found that by slightly sacrificing temporal dependencies, our ECAP can significantly improve inference speed. Furthermore, considering that ECAP may bring challenges in cross-frame alignment, we designed a metadata-driven alignment (MDA) module to refine the motion vectors rather than calculating cross-frame offsets from scratch. MDA first transforms motion vectors into coarse optical flows and then iteratively refines them over several scales into dense feature-level optical flows. In this way, MDA significantly improves the quality of motion vectors while achieving faster alignment speed by exploiting metadata. Extensive experimental results demonstrate that our COVSR not only achieves efficient and superior super-resolution performance but also is generalizable to various compression configurations. Our code will be available and the project page is at https://covsr.github.io.**

***Index Terms*—Compressed video super-resolution, propagation scheme, metadata, optical flow.**

I. Introduction

CURRENT video platforms and capture devices enable people to create, share, and watch videos much more easily. These videos are typically compressed using compression standards like AVC/H.264, which offer various compression configurations. These standards and configurations, however, often lead to compression distortions and loss of details. When video super-resolution (VSR) methods [5], [21], [22], [24], [32], [37], [40]–[42] are applied directly to these compressed videos, the results are often not visually pleasant. This is because VSR models tend to recognize compression artifacts as textures and then amplify them into unnatural textures. In addition, the lack of specific modifications to the design of networks under the compression context and the insufficient use of metadata would adversely impact the overall performance.

Recently, several works [1]–[4], [19] have been proposed to investigate the compressed video super-resolution task. COMISR [1] and FTVSR [2] mitigate the information loss from compression by perceiving and inferring high-frequency details from a decoded image sequence. CIAF [4] and CAVSR [3] further utilize metadata to assist and enhance feature fusion for super-resolution, achieving a better trade-off between speed and performance. These compressed video super-resolution methods leverage the *recurrent propagation* scheme from VSR. Although recurrent propagation performs well in VSR, directly applying it to the compressed video super-resolution task may lead to two problems: (i) The strong temporal dependency of recurrent propagation limits its capability to jointly update multiple features simultaneously, thus restricting parallel processing. (ii) A compressed video has distinct frame types (I-, B-, and P-frames) with different dependencies among them. Recurrent propagation treats all frames identically and propagates adjacent frames straightforwardly, which cannot be generalized to various compression configurations. In Fig. 1(a), there are α interval frames between the reference frame and the current frame in the metadata. Direct use of recurrent propagation will result in a mismatch of reference states as shown in Fig. 1(b). This can further lead to feature misalignment and inferior results, as seen in Fig. 1(c).

To overcome those problems, we propose a **Compression-Omniscient Video Super-Resolution (COVSR)** method, which can address compressed video super-resolution for both low-delay and random-access configurations. COVSR mainly consists of four modules: feature extraction module, efficient compression-aware propagation, metadata-driven alignment module, and upsampling module. Considering that decoding timestamps are designed for the video codec rather than for the VSR task, directly employing them as the propagation order could lead to suboptimal results. As a result, we design a novel *Efficient Compression-Aware Propagation (ECAP)* scheme to dynamically adjust propagation routes in accordance with the compression configurations. Specifically, our ECAP consists of three main components: *decoding-based dual-way propagation*, *long short-term connections*, and *joint update mechanism*. The decoding-based dual-way propagation matches reference frames and current frames according to the decoding timestamps and dependencies between different frame types in dual ways. Such a design enables our model to perceive global

Shuyun Wang, Zhuojie Wu, and Xin Yu are with the School of Electrical Engineering and Computer Science, University of Queensland, Brisbane 4067, Australia (e-mail: shuyun.wang@uq.edu.au; zhuojie.wu@uq.edu.au; xin.yu@uq.edu.au). *(Corresponding author: Xin Yu.)*

Yanbin Liu is with the School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand (e-mail: csyanbin@gmail.com).

Ming Lu is with Intel Lab China, Beijing 100876, China (e-mail: lu199192@gmail.com).

Senmao Tian is with the School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: smtian1204@gmail.com).

Yandong Guo is with AI^2 Robotics, Shenzhen, China (e-mail: yandong.guo@live.com).

Manuscript received ...; revised ...

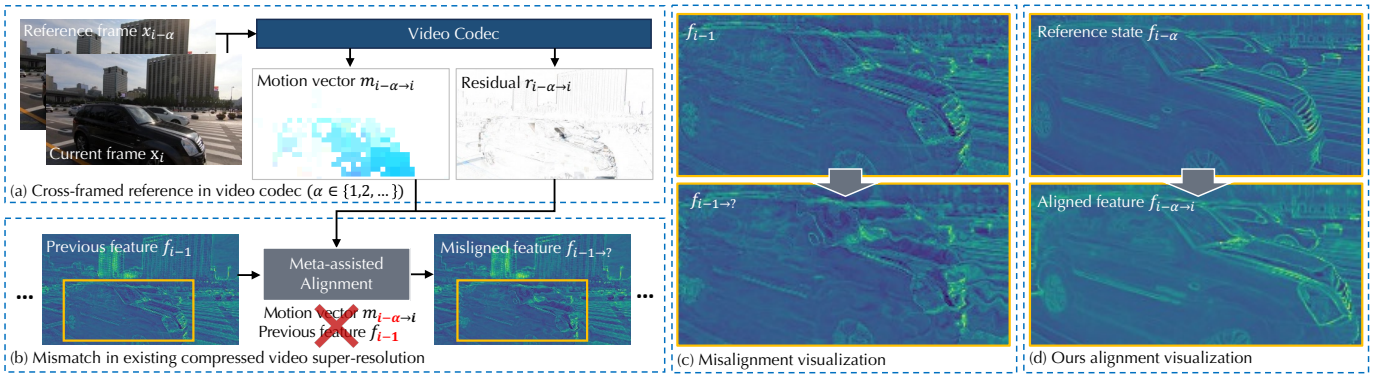

Fig. 1. **Comparison of our method and the state-of-the-art compressed video super-resolution method (CAVSR [3]).** (a) At time step i, metadata is generated by a current frame x_i and a reference frame $x_{i-\alpha}$, where the value of α is decided by the codec process. (b) Since α is not always equal to 1, using mismatched metadata (motion vector $m_{i-\alpha\to i}$) in CAVSR to align the previous adjacent feature f_{i-1} will cause misalignment. (c) Some visualizations of misalignment in CAVSR. (d) Our alignment method accurately models the moving car and retains better contours thanks to the effective use of metadata.

information while maintaining compatibility with metadata. Moreover, the long short-term connections further enlarge the gathering range of reference states for I- and P-frames by additionally incorporating B-frame states into propagation. More importantly, we find that by slightly sacrificing temporal dependencies, we can reorganize the propagation order to jointly update the features of B-frames and achieve better parallelization capability during dual-way propagation, thus significantly accelerating inference speed.

Existing alignment modules in VSR [20], [23], [26], [34]–[36], [38], [39] often struggle to balance speed and accuracy in cross-frame alignments. Alternatively, integrating motion vectors embedded in the video bitstream seems promising as they provide coarse motion information among frames. Directly employing motion vectors [4] or augmenting them with sparse residual maps [3] results in blocky artifacts. Leveraging the correlation matrix to refine motion vectors [9] imposes a severe computational burden for a VSR network. In order to effectively utilize the motion vector for rapid optical flow generation and accurate cross-frame alignment, we propose a *Metadata-Driven Alignment (MDA)* module. MDA first transforms motion vectors to the course optical flow in the lower resolution and iteratively refines the motion information across multiple scales. Here, deep supervision [45] is adopted to stabilize the learning process of MDA across different scales. Since MDA is introduced to align cross-frame features, we free the optical flow supervision when the entire network starts converging. In this fashion, MDA significantly improves the motion information provided by motion vectors while achieving faster alignment speed with the help of metadata.

Extensive experiments demonstrate that our method not only achieves efficient and superior super-resolution performance but also is generalizable to various compression configurations. Our contributions are summarized as follows:

- We propose a general Compression-Omniscient Video Super-Resolution (COVSR) method that takes rational advantage of metadata for enhanced parallelization and performance. By devising a novel efficient compression-aware propagation scheme, this work is the first to study propagation in the challenging compressed video super-resolution task.
- We design a lightweight metadata-driven alignment module in COVSR to efficiently utilize the motion information within motion vectors to achieve fast flow estimation and accurate cross-frame alignment.
- Extensive experiments demonstrate the effectiveness and efficiency of the proposed method on compressed video super-resolution benchmarks, where state-of-the-art results are achieved.

II. Related Work

In this section, we first review the video super-resolution methods related to our work. Then, we discuss the flow estimation techniques on compressed videos.

A. Video Super-Resolution

1) Uncompressed Video Super-Resolution: The recurrent structure is a widely embraced framework employed in diverse video processing tasks, which is mainly divided into unidirectional propagation and bidirectional propagation in VSR. Sajjadi *et al.* [5] integrate explicit optical flow into the unidirectional propagation, effectively reducing the additional computations arising from the repetitive calculations of the sliding window. Chan *et al.* [6] introduce a bidirectional propagation method that incorporates embedded optical flow, elevating the quality of recovery. Yi *et al.* [7] merge sliding window and recurrent architectures to use super-resolution outputs from both current and future frames. Chan *et al.* [8] extend bidirectional propagation to the second order and grid-like manner and employ deformable convolution-based alignment for each recurrent unit. Zhou *et al.* [42] use feature-level temporal continuity between adjacent frames to reduce redundant computations.

The increasing complexity of the alignment modules along with the lack of parallelism in the propagation design leads to their poor efficiency. In addition, the lack of specific modifications to the design of networks and the insufficient use of metadata results in poor performance when applying these methods to compressed videos.

2) Compressed Video Super-Resolution: Different from uncompressed video super-resolution, the main focus of compressed video super-resolution is to address issues caused by the compression process such as block artifacts and the loss of high-frequency details. Various approaches have been explored in this field. Chen *et al.* [43] is the first to jointly leverage coding priors and deep priors for VSR. Li *et al.* [1] introduce a detail-aware flow estimation module to compute optical flows from scratch and incorporate a Laplacian enhancement module, aiming to restore high-frequency details within a bidirectional framework. Qiu *et al.* [2] first reconstruct the video using a pre-trained transformer model and then employ a frequency transformer to recover high-frequency details of high-resolution frames bidirectionally. Zhang *et al.* [4] leverage motion vectors from the video bitstream for motion compensation and use residual maps to expedite inference in a unidirectional manner under the low-delay configuration. Wang *et al.* [3] sequentially leverage metadata embedded in compressed video streams and integrate compression encoders accommodate to various compression levels.

Effectively utilizing metadata embedded in the bitstream can greatly improve performance. However, existing methods leverage metadata under the low-delay configuration without taking the random access configuration into account. Therefore, in this paper, we propose an efficient compression-aware propagation scheme that can rationally take advantage of metadata under different configurations.

B. Flow Estimation on Compressed Videos

While most VSR methods [1], [6], [23] directly employ pretrained optical flow networks for alignment. However, there are differences between the generic optical flow and the VSR-specific optical flow, which leads to imprecise alignment. Some recent approaches [8], [16] utilize deformable convolution along with complex operations to enhance feature alignment, but the time complexity and the number of parameters are significantly increased. Zhang *et al.* [4] use motion vectors for alignment. However, since the motion vectors are block-wise and sparse, they are not good enough compared to optical flows. Wang *et al.* [3] attempt to correct motion vectors using residuals from the bitstream, but sparse residual maps can only transfer motion vectors to the coarse optical flow. Zhuo *et al.* [9] use a complex correlation matrix to refine motion vectors into the optical flow domain. However, the large computational cost associated with its complex design makes it difficult to deploy in compressed video super-resolution networks. Moreover, they all estimate the optical flow in neighboring frames and do not take cross-frame offsets into account. In contrast, our metadata-driven alignment module first transforms cross-frame motion vectors to coarse optical flows, and then refines them across multiple scales through a lightweight network, enabling the rapid generation of VSR-specific optical flow for cross-frame feature alignment.

III. Preliminary

A. Frame Types

Typically, a video bitstream contains multiple groups of pictures (GOPs), each of which contains multiple coded frames. Images between two neighboring I-frames form a GOP. Each frame type carries specific metadata that aids in video decoding and reconstruction. The I-frame contains the full image data and is transmitted entirely over the network. P-frames and B-frames, on the other hand, are transmitted with metadata describing differences from other decoded frames. The difference between a P-frame and a B-frame lies in their reference positions. A P-frame references only previous I-frames or P-frames, while a B-frame references both previous and future I-frames or P-frames. This referencing allows for efficient compression by minimizing redundant data. It is worth noting that the following discussion is limited to a single GOP, and a consistent scheme is applied across GOPs.

B. Video Coding Structures

The three types of mainstream video coding structures are all-intra, low-delay, and random access. (i) The *all-intra configuration* makes every frame in the video stream an I-frame, which allows decoding from any frame. It is preferred in professional video editing due to the ease of frame-by-frame manipulation. (ii) In the *low-delay configuration*, the first frame is an I-frame while the others are encoded as all P-frames or B-frames. This configuration is designed to minimize capture-to-display delays and is suitable for real-time scenarios. (iii) The *random access configuration* consists of I-frames, P-frames, and B-frames, which have a different decoding order than the presentation order. The future frame in presentation order can be encoded before the current frame and can also become the reference frame of the current frame. This configuration is commonly used for final encoding due to its superior coding efficiency compared to other compression configurations. Our approach mainly focuses on optimizing the super-resolution of compressed video under the random access configuration. For other compression configurations, we also achieve competitive results.

C. Timestamps

Decoding timestamps and presentation timestamps are components of bitstream metadata and are usually stored together. Along with other information such as frame types, these timestamps contribute to accurate video decoding and smooth display. Specifically, presentation timestamps indicate the time when the video frame is displayed on the screen. Decoding timestamps determine the order in which video frames are decoded. They guide the decoder on when to decode these frames, allowing it to engage motion compensation and pixel prediction on time. In random access configurations, decoding timestamps are often different from presentation timestamps.

D. Motion Vectors

In video compression, motion vectors describe the direction and magnitude of motion and measure the displacement of pixels over time. Motion estimation algorithms generate these vectors by comparing pixel differences between frames to find the best match. Frames are divided into macroblocks, and during the encoding process, each macroblock is matched to a

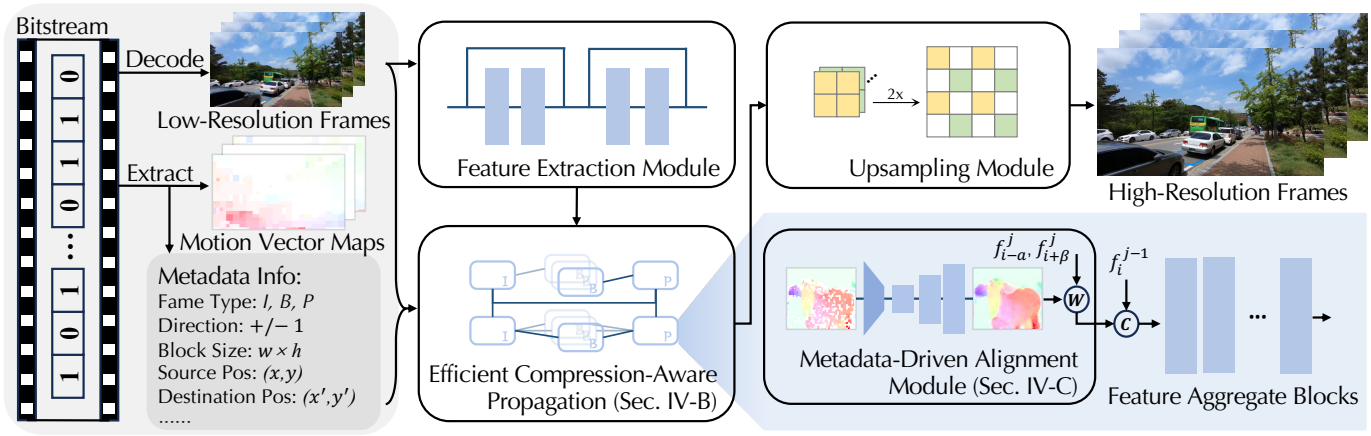

Fig. 2. **Overview of our Compression-Omniscient Video Super-Resolution.** Shallow features are first extracted from the feature extraction module. Then, we design two novel modules for enhanced video super-resolution: (1) **Metadata-driven alignment module** (Sec. IV-C) use a lightweight network to refine sparse motion vectors to more accurate dense feature-level optical flow, followed by the feature aggregate blocks. (2) **Efficient compression-aware propagation** (Sec. IV-B) is a novel scheme that dynamically adjusts propagation routes during propagation. Here, $f^j_{i-\alpha}, f^j_{i+\beta}$ and f^{j-1}_i represent features of the two dynamic reference states and the previous state, $\mathcal{W}$ is the warping operation, and $\mathcal{C}$ is the concatenation operation. After propagation, the final features are fed into the upsampling module to generate high-resolution frames.

similar-looking macroblock in a previously encoded frame. In this way, only the motion vectors need to be transmitted, rather than all the macroblocks, effectively reducing the amount of data transmitted or stored. To make full use of the motion information in the bitstream, our method computes the motion vectors in both forward and backward versions from metadata. These motion vectors provide critical information that enhances our understanding of pixel movement between frames, enabling more efficient and accurate video reconstruction.

IV. Methodology

A. Overview

The architecture of our Compression-Omniscient Video Super-Resolution (COVSR) is shown in Fig. 2. COVSR aims to restore the high-resolution (HR) frames from low-resolution (LR) frames with the assistance of metadata, such as frame types and block sizes. Specifically, COVSR consists of four modules: feature extraction module, Efficient Compression-Aware Propagation (ECAP), Metadata-Driven Alignment (MDA) module, and upsampling module.

Let x_i denotes the current frame at time step i, and $x_{i-\alpha}$, $x_{i+\beta}$ represent its two reference frames. In order to generate a high-resolution frame, x_i undergoes different states to extract rich features, *i.e.*, f^j_i represents the feature at the j-th ($j = 1, 2, 3$) state. The high-resolution frame of x_i can be generated using the following steps:

1) The shallow feature f^1_i of the frame x_i is obtained from the feature extraction module.
2) MDA first refine motion vectors between the reference frames and the current frame to obtain accurate feature-level optical flows $o_{i-\alpha\to i}$ and $o_{i+\beta\to i}$. Then, aligned reference states $f^j_{i-\alpha\to i}, f^j_{i+\beta\to i}$ are obtained by optical flow and warping operations.
3) The aligned reference states are fed into the feature fusion blocks along with the previous state f^{j-1}_i to generate the aggregation feature f^j_i. The f^j_i will then be propagated as a state in ECAP.
4) The final feature f^3_i is derived after propagation. We input it to the upsampling module to generate the high-resolution frame.

Details of the efficient compression-aware propagation and the metadata-driven alignment module are described in Sec. IV-B and Sec. IV-C, respectively.

B. Efficient Compression-Aware Propagation

According to the video codec, we propose an efficient compression-aware propagation scheme to dynamically adjust propagation routes under the compression context, as shown in Fig. 3(b). In our context, propagation is primarily used to specify the frame index of the current frame and reference frames and the spread of features. Each propagation step is accompanied by an update of the specific feature, which includes the alignment of reference states, and the aggregation of aligned features:

$$f^j_i = \mathcal{A}\Big(\mathcal{C}\big(f^{j-1}_i, f^j_{i-\alpha\to i}, f^j_{i+\beta\to i}\big)\Big), \tag{1}$$

where $\mathcal{A}$ represents feature aggregation blocks, $\mathcal{C}$ denotes concatenation along channel dimension, $f^j_{i-\alpha\to i}$ and $f^j_{i+\beta\to i}$ denote the reference states warped from the time step $i-\alpha$ and $i+\beta$ to i.

The existing recurrent propagation scheme is shown in Fig. 3(a). It sequentially updates features of all frame types (I-, B-, and P-frames) in both forward and backward processes. This simple scheme does not match the intrinsic decoding orders stored in metadata, therefore the metadata cannot be used effectively to enhance the performance. In contrast, our design is tailored for the compressed video scenario and is compatible with different compression configurations. In particular, our propagation includes a forward process and a backward process that are different from bidirectional propagation. In the forward process, the first step is to update the features of the I-frame and P-frames sequentially. The I-frame does not receive any reference state, while the P-frame

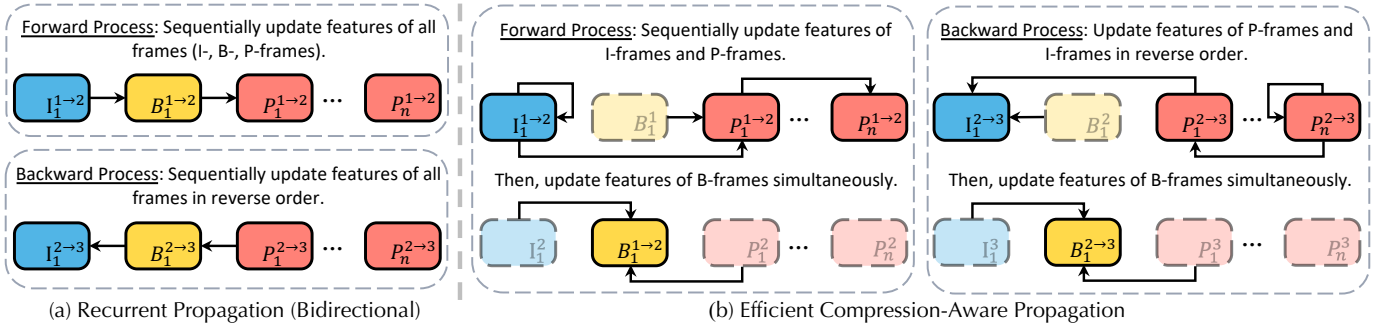

Fig. 3. **Illustration of different propagation schemes.** (a) **Recurrent Propagation (Bidirectional)**. It sequentially updates the features of all frames(I-, B-, P-frames) in both forward and backward processes. (b) **Efficient Compression-Aware Propagation**. In contrast, our efficient compression-aware propagation is specially designed for the compressed video scenario to improve the reconstruction performance. Here, I, P, and B denote the intra-frame, predicted frame, and bidirectional predicted frame. The subscript indicates the indexes of the different types of video frames (from 1 to n), while the superscript indicates the update state (from 1 to 3). For example, $P_1^{2\to3}$ represents the feature of the first P-frame updates from the second to the third state.

receives states from either the previous I-frame (or P-frame) and the previous adjacent frame. The second step is to jointly update the features of all B-frames using the reference states of the most recent I-frame or P-frames before and after. In the backward process, the first step is to update the features of the I-frame and P-frames in reverse order. The last P-frame does not receive any reference state, and other P-frames and the I-frame receive states of the next P-frame and the next adjacent frame. The second step is the same as the forward process.

Considering the video codec, our propagation follows three design principles to facilitate the application to the video compression and super-resolution scenario, aiming for better efficiency and performance:

1) Propagation needs to have *good compatibility with metadata* under different compression configurations.
2) Feature updates need to obtain a *sufficient number of reference states* to improve reconstruction performance.
3) *Jointly updating features* of specific frames to get a better trade-off between efficiency and performance.

For detailed implementation, we make several components to reflect the three practical design principles. With respect to the **first principle**, when updating the features of a current frame, we dynamically adjust the index of the reference state and the current state according to the decoding timestamp. Depending on the order in which the codec constructs the metadata, our propagation can find the correct reference frame for I-frames, B-frames, and P-frames to utilize the metadata, rather than only adjacent references that lead to metadata mismatches and feature misalignment. As for the **second principle**, we made two innovative components. (1) We design a decoding-based backward process according to decoding timestamps to enhance the global modeling capability, combined with a forward process, which we name (*decoding-based dual-way propagation*. (2) We design *long short-term connections* for all I-frames and P-frames to maximize the use of metadata and reference states by additionally incorporating B-frame states into propagation. These two components are implemented by using metadata in both forward and backward versions to incorporate the inverse routes into propagation. This allows the P-frame to reference not only the previous P-frame or I-frame but also the next P-frame and the two adjacent frames. In this way, we can rationally make the I-frame and the P-frame receive sufficient reference states to ensure a robust restoration. We modify the current frame order to achieve the **third principle** while being compatible with metadata. Alternating the reconstruction of P-frames and B-frames in an order of the decoding timestamp results in sub-optimal reconstruction efficiency. We find that by sacrificing slightly temporal dependencies, we can achieve a better trade-off between inference speed and performance. Specifically, we first update all the features of the I-frames and P-frames and store them in the cache. Then, we exploit these features as reference states to jointly update the features of all B-frames. This *joint update mechanism* greatly improves speed and makes it possible to use multiple computational resources for further acceleration, which is not available in existing recurrent propagation.

C. Metadata-Driven Alignment Module

For a frame x_i, the propagation requires to align the reference states $f_{i-\alpha}^j, f_{i+\beta}^j$ to the current time step i, according to Eq. 1. Therefore, we design a lightweight metadata-driven alignment module that can compute the optical flows $o_{i-\alpha\to i}$ and $o_{i+\beta\to i}$ between reference frames $x_{i-\alpha}$, $x_{i+\beta}$ and frame x_i, and then complete the alignment by warping:

$$f_{i-\alpha\to i}^j = \mathit{MDA}(x_i,\ x_{i-\alpha},\ f_{i-\alpha}^j,\ m_{i-\alpha\to i}),$$

where $m_{i-\alpha\to i}$ represent the motion vector map from $x_{i-\alpha}$ to x_i, similar for $f_{i+\beta\to i}^j$. The initial representation of the motion vector is a group of vectors, each of which records compressed positions, sizes, and motion offsets of the block. We convert the motion vector into two different directions of motion vector maps by filling the pixels in each block with the same motion offset.

1) Optical Flow Estimation: There is a significant discrepancy in the distribution between the motion vector map and the optical flow, posing a challenge for contemporary optical flow estimation modules [3], [4] to harness this data effectively. In response, MDA integrates sparse block-wise motion vector maps into dense pixel-wise optical flows, as shown in Fig. 4. MDA draws inspiration from two fundamental

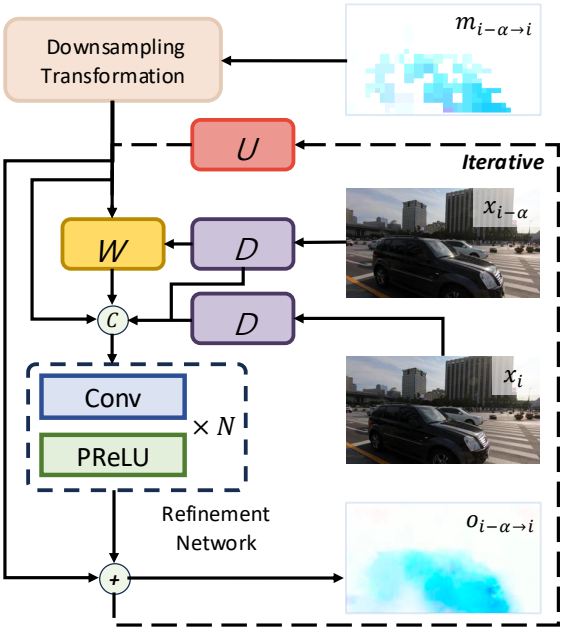

Fig. 4. **Optical flow estimation of our MDA.** 'W': warp operation, 'D': bilinear downsampling interpolation, and 'U': upsampling. To leverage the motion vector for fast flow estimation, MDA first transforms motion vectors into the optical flow domain at a lower resolution through downsampling transformation. Then, the motion information is refined iteratively across multiple scales to generate more accurate optical flows.

aspects: (1) Motion vectors contain inaccurate motions in some areas, resulting from a mismatch between the coarse macroblock division and the motion of the object. (2) Different levels of compression and compression configurations affect the quality of the motion vectors themselves. Therefore, we convert the motion vector map from the macroblock domain to the pixel domain through a downsampling transformation. Then, we utilize a coarse-to-fine multiscale refinement network to gradually refine the sparse pixel domain motion vectors to dense feature-level optical flow domains.

Specifically, input the motion vector maps through a downsampling transformation to obtain a coarse optical flow map $o^L_{i-\alpha\to i}$. The downsampling transformation consists of a $L\times$ bilinear downsampling interpolation followed by a convolutional layer. Lower resolution makes it easier to capture large amounts of movement, while higher resolution allows for more focus on detailed movement. To refine the motion information, we enhance the multiscale optical flow by exploring the motion between the input frame and the warped frame through a Refinement Network (RN). Initially, we warp the downsampling reference image $x^l_{i-\alpha}$ using the coarse optical flow to obtain warped frames $x^l_{i-\alpha\to i}$. Subsequently, we input the coarse optical flow $o^l_{i-\alpha\to i}$, downsampled reference frame, downsampled current frame, and warped frame into the refinement network to obtain a refined optical flow $o^l_{i-\alpha\to i}$:

$$o^l_{i-\alpha\to i} = U_{\uparrow}(RN(o^{l+1}_{i-\alpha\to i}, x^{l+1}_{i-\alpha\to i}, x^{l+1}_{i-\alpha}, x^{l+1}_{i}) + o^{l+1}_{i-\alpha\to i}), \quad (2)$$

where $l \in [0, L]$ is the scale level, RN is the refinement network and $U_{\uparrow}$ represents the bilinear upsampling interpolation. We upsample the fine optical flow map to obtain the next level of coarse optical flow. This process is repeated iteratively until the final optical flow is obtained. It is noteworthy that our final level performs refinement without upsampling.

2) Distillation with Deep Supervision: Given that our MDA differs from previous VSR methods [1], [6], [8], [23] that rely on pre-trained optical flow estimation, we incorporate distillation loss with deep supervision to enhance the prediction accuracy of optical flows. These loss are employed to fine-tune the multiscale flow, allowing MDA to learn the distribution of the optical flow domain. Our training objective is to minimize both the end-point-error (EPE) loss and Charbonnier loss [14] between the input frame and the corresponding warp frame on multiple scales. Specifically, we utilize the teacher model [44] to generate multiscale image-level optical flow pseudo-labels $o^{pseudo}_{i-\alpha\to i}$. The total loss function of MDA is:

$$\begin{aligned} L_{MDA} = \lambda \sum_{l=0}^{L} EPE(o^l_{i-\alpha\to i}, o^{pseudo}_{i-\alpha\to i}) \\ + \eta \sum_{l=0}^{L} Charbonnier(x^l_{i-\alpha\to i}, \mathcal{W}(o^l_{i-\alpha\to i}, x^l_{i-\alpha})), \end{aligned} \quad (3)$$

where EPE calculates per-pixel end-point-error, λ and η are weights of the loss. The underlying assumption of the above loss is that the optical flow is identical to the flow used for image alignment. However, the optical flow used for aligning features is structurally very similar to the aforementioned optical flow but slightly different in detail. Therefore, we halt loss after a certain iteration and release the constraints on the entire network to better accommodate the compressed video-super-resolution task.

V. Experiments

A. Implementation Details

1) Compressed Datasets: Following previous works [1]–[3], we utilize REDS [10] and Vimeo-90k [11] for training. The REDS dataset comprises 270 videos, each containing 100 frames at a resolution of 1280 × 720. To ensure a fair comparison, as in previous work [2], four sequences are reserved for testing, referred to as REDS4, while the remaining clips are used for training. Vimeo-90k consists of approximately 65K 7-frame video clips, each having a resolution of 448 × 256. Similar to previous work [3], we use Vid4 [12] as test sets alongside Vimeo-90K. To generate compressed LR frames, we utilize the H.264 encoder and the popular FFmpeg 4.3. The CRF values were set to 0, 15, 25, and 35, following [1]–[3].

2) Architecture Design: Our feature extraction module and upsampling module follow the design of previous methods, where the feature extraction module adopts a single convolutional layer and the upsampling module uses two stacked convolution and pixel shuffling operations. The proposed refinement network has 3 combinations of convolution and PReLU to ensure that the entire metadata-driven alignment module is lightweight. Our COVSR uses three Multi-Frame Self-Attention Blocks (MFSABs) [15] as the feature aggregate blocks and has one shortcut connection for every 3 MFSABs. We apply second-order propagation similar to BasciVSR++ [8], where features are propagated forward and backward twice in an alternating fashion. All ablation studies are tested on REDS4 and CRF25, where the feature aggregation blocks equal 7 residual blocks, and only one forward

TABLE I
QUANTITATIVE COMPARISON (PSNR/SSIM) ON COMPRESSED VIDEOS OF VID4 AND REDS4 FOR THE 4× VSR SETTING. THE RESULTS OF VID4 ARE CALCULATED ON THE Y-CHANNEL WHILE THE RESULTS OF REDS4 ARE CALCULATED ON THE RGB CHANNELS. THE LATENCY IS CALCULATED ON A GROUP OF PICTURES (GOP) OF 100 FRAMES, 75% B-FRAMES, AND HR IMAGE SIZE OF 1280×720.

Method	Params (M)	Latency (ms)	Average on Compressed REDS4			Average on Compressed Vid4		
			CRF15	CRF25	CRF35	CRF15	CRF25	CRF35
DUF [41]	5.8	974	25.61/0.775	24.19/0.692	22.17/0.588	24.40/0.773	23.06/0.660	21.27/0.515
FRVSR [5]	5.1	137	27.61/0.784	25.72/0.696	23.22/0.579	26.01/0.766	24.33/0.655	22.05/0.482
EDVR [16]	20.6	378	28.72/0.805	25.98/0.706	23.36/0.600	26.34/0.771	24.45/0.667	22.31/0.534
RSDN [18]	6.2	94	27.66/0.768	25.48/0.679	23.03/0.579	26.58/0.781	24.06/0.650	21.29/0.483
BasicVSR [6]	6.3	63	29.05/0.814	25.93/0.704	23.22/0.596	26.56/0.780	24.28/0.656	21.97/0.509
IconVSR [6]	8.7	70	29.10/0.816	25.93/0.704	23.22/0.596	26.65/0.782	24.31/0.657	21.97/0.509
BasicVSR++ [8]	7.3	77	29.75/0.822	26.30/0.703	23.58/0.596	27.08/0.796	24.39/0.620	22.07/0.517
VRT [32]	35.6	243	29.69/0.819	26.36/0.704	23.59/0.595	27.25/0.800	24.41/0.663	22.06/0.516
RVRT [31]	10.8	183	29.89/0.825	26.36/0.705	23.59/0.596	27.40/0.802	24.43/0.663	22.07/0.517
MIA-VSR [42]	16.5	822	29.72/0.821	26.31/0.703	23.57/0.595	-	-	-
COMISR [1]	6.2	73	28.40/0.809	26.47/0.728	23.56/0.599	26.43/0.791	24.97/0.701	22.35/0.509
FTVSR [2]	43.3	850	30.51/0.853	28.05/0.776	24.82/0.657	27.40/0.811	25.38/0.706	22.61/0.540
CAVSR [3]	8.9	93	-	-	-	26.74/0.798	25.11/0.712	22.56/0.556
COVSR (Ours)	11.5	223	**30.64/0.860**	**28.46/0.788**	**25.13/0.671**	**27.96/0.837**	**25.98/0.753**	**22.90/0.580**

and backward process is performed. Latency is calculated on a GOP of 100 frames, 75% B-frames with the same frame size as REDS4.

3) Training Details: During training, the batch size and patch size of the LR images are set to 16 and 64×64, respectively. We use random rotation and flipping operations as data enhancement techniques. In our experiments, we set $\lambda = 0.33$, $\eta = 0.33$ and $L = 3$. Using Adam optimiser [13] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, we optimize the motion vector refinement module by performing 100K iterations of Eq. 3 in the first stage with the initial learning rate set to 2×10^{-4}. In the second stage, we use only the reconstruction loss, and the learning rate of the motion vector refinement module is reduced to 5×10^{-5}. We use the Charbonnier loss [14] as the reconstruction loss, and the total number of iterations is 400K. All experiments are implemented using PyTorch on a server with V100 GPUs.

B. Comparison with the State-of-the-Art (SoTA)

In this section, we compare our method with thirteen state-of-the-art VSR methods, including DUF [41], FRVSR [5], EDVR [16], RSDN [18], BasicVSR [6], IconVSR [6], BasicVSR++ [8], VRT [32], RVRT [31], MIA-VSR [42], COMISR [1], FTVSR [2] and CAVSR [3]. Following the compressed settings as COMISR [1] and FTVSR [2], we compress the test videos with several compression rates (CRF15, CRF25, CRF35) and evaluate the compressed videos *w.r.t.* PSNR and SSIM, as shown in Table I.

For REDS [10] dataset, although recent MIA-VSR [42] and RVRT [31] achieve state-of-the-art results on uncompressed videos, they do not perform well on the compressed videos. For example, RVRT achieves 26.36dB and 23.59dB in PSNR of compression CRF25 and CRF35. Meanwhile, MIA-VSR, which performs better than RVRT on uncompressed videos, only obtains 26.31dB and 23.57dB in PSNR of compression CRF25 and CRF35. This phenomenon shows that directly applying the ideal super-resolution method for compressed video is unsatisfactory. In contrast, our COVSR method is tailored for the compressed video scenario, significantly outperforming uncompressed baselines and achieving state-of-the-art.

For Vid4 [12] dataset, all results are obtained by fine-tuning on compressed videos as the same training settings of COMISR [1]. On compressed videos with a compression rate of CRF15, 25, and 35, our COVSR achieves 27.96dB, 25.98dB, and 22.90dB in PSNR, respectively. These results surpass all competitors and demonstrate the huge potential of COVSR on the task of compressed video super-resolution.

Overall, our COVSR achieves state-of-the-art results across three compression levels on both REDS4 and Vid4. Besides, COVSR demonstrates competitive running speed and parameter efficiency. Compared with the strong competitor FTVSR [2], COVSR excels in both the number of parameters and latency by 3∼4 times. Although CAVSR [3] has slightly better efficiency than our method, it scarifies the performance by a large amount (*e.g.*, 25.98dB vs. 25.11dB on CRF25 of Vid4). Our COVSR method attains a good balance of both efficiency and effectiveness, promising for real application of compressed video super-resolution.

The qualitative comparison with SoTA methods is illustrated in Fig. 5. Our method excels in recovering higher-quality HR images, capturing finer details and sharper edges. Notably, our methods are the only one to successfully restore a clear outline of the number in the Calendar clip. In contrast, other methods either struggle to recover missing details or introduce artifacts.

C. Comparison on the Low-delay Configuration

CIAF [4] is conducted under the low-delay configuration. In order to compare with CIAF and show our robustness in different configurations, we perform experiments following

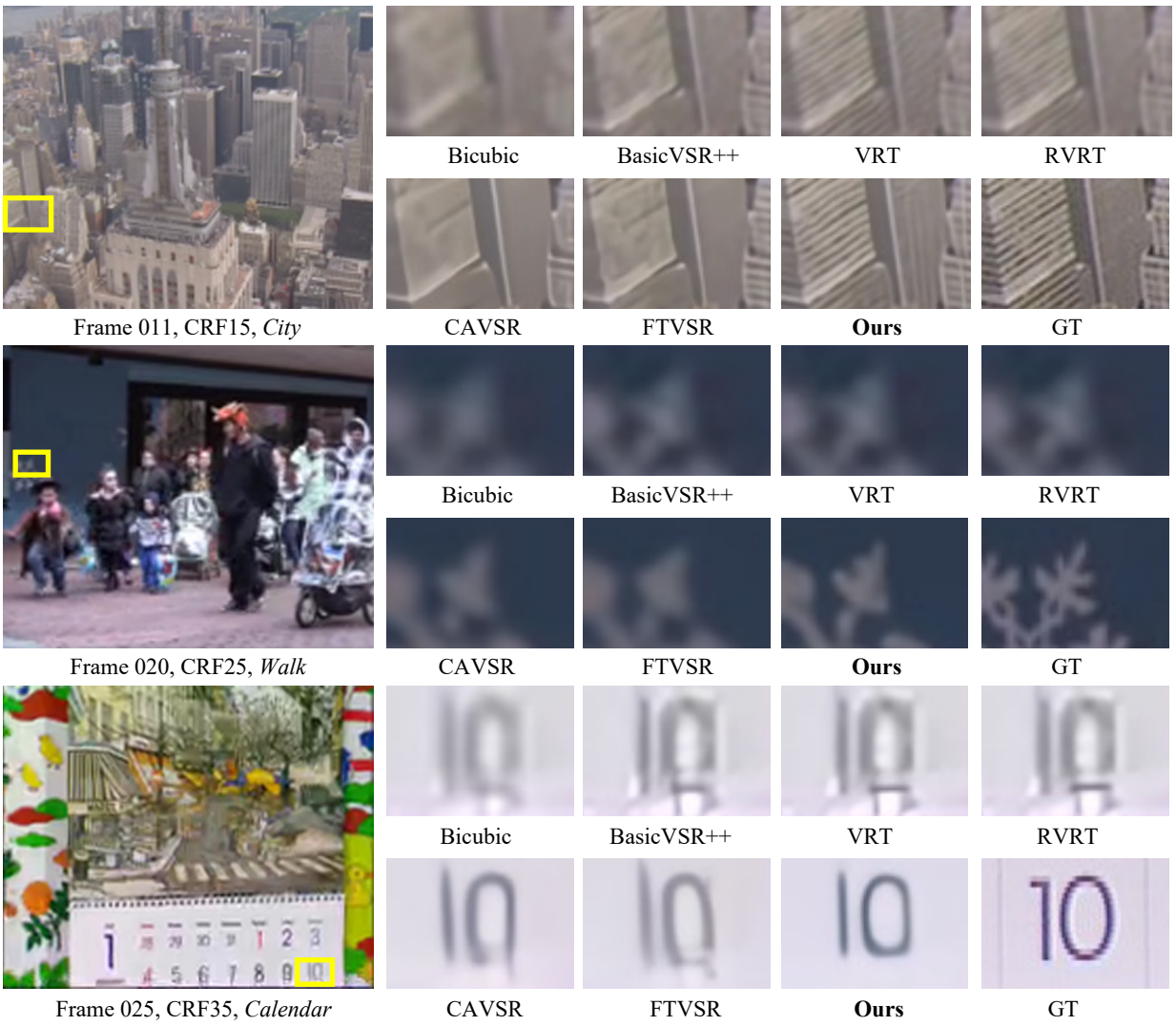

Fig. 5. **Qualitative comparison on the compressed Vid4 [12] test set for the 4× VSR setting.** Zoom in for better visualization.

the CIAF setup as shown in Table II. In this configuration, the frame types are only I-frames and P-frames. Our propagation degrades into recurrent propagation in this situation. However, the block-level motion vectors are inaccurate and sparse, preventing the features aligned by motion vectors from being effectively utilized by the CIAF. As for CAVSR [3], augmenting them with sparse residual maps will also result in misaligned features. In contrast, we propose an MDA that can successfully utilize the motion information stored in the bitstream and greatly improve the reconstruction performance.

TABLE II
THE QUANTITATIVE RESULTS (PSNR/ SSIM) OF CIAF [4], CAVSR [3] AND COVSR ON THE VID4 TEST SET UNDER THE LOW-DELAY-P CONFIGURATION. PSNR IS CALCULATED ON THE Y-CHANNEL; SSIM IS CALCULATED ON THE RGB CHANNEL. 4× UPSAMPLING IS PERFORMED.

Model	CRF18	CRF23	CRF28
CIAF [4]	25.13/0.6990	24.20/0.6355	23.01/0.5557
CAVSR [3]	26.55/0.7568	25.82/0.7100	24.32/0.6151
Ours	**27.34/0.7903**	**26.19/0.7331**	**24.78/0.6466**

D. Ablation Study

1) Ablation on Propagation Schemes: We compare the same backbone network under *Unidirectional* propagation, *Bidirectional* propagation, and *Efficient Compression-Aware* propagation (denoted as UniP, BiP, ECAP). The PSNR of ECAP is significantly better than UniP and BiP, as shown in Table III, highlighting the effectiveness of leveraging codec prior. In terms of parameters, FLOPs, and latency, ECAP has similar FLOPs to BiP, but the latency is lower than UniP and approximately 40% lower than BiP, thanks to our ability to perform parallel inference. Since the pre-trained optical flow is based on the presentation order, it is not well-adapted to the propagation structure of the codec. The performance can be further improved by replacing OF [20] with the optical flow estimation of MDA. Moreover, the latency is also decreased due to the lightweight design of MDA.

We further examine the effect of each component in our proposed propagation scheme, and show the results in Ta-

TABLE III
COMPARISON OF DIFFERENT PROPAGATION SCHEMES. FLOPS ARE COMPUTED ON AN LR FRAME OF 180×320 PIXELS. 'UNIP', 'BIP', 'ECAP', AND 'OF' REPRESENT UNIDIRECTIONAL PROPAGATION, BIDIRECTIONAL PROPAGATION, EFFICIENT COMPRESSION-AWARE PROPAGATION, AND OPTICAL FLOW MODULE [20], RESPECTIVELY.

Propagation			Flow Estimation		Params	FLOPs	Latency	PSNR
UniP	BiP	ECAP	OF	MDA	(M)	(G)	(ms)	(dB)
✓			✓		4.0	0.22	33.94	27.10
	✓		✓		4.0	0.36	56.28	27.37
		✓	✓		4.1	0.35	32.89	27.44
		✓		✓	3.3	0.33	26.58	27.57

TABLE IV
ABLATION STUDY OF OUR PROPAGATION SCHEME. DBDP REPRESENTS THE DECODING-BASED DUAL-WAY PROPAGATION, LSTC, AND JUM INDICATE THE LONG SHORT-TERM CONNECTIONS AND THE JOINT UPDATE MECHANISM.

DBDP	LSTC	JUM	FLOPs(G)	Latency(ms)	PSNR(dB)
✓			0.23	34.92	27.36
✓	✓		0.33	37.13	27.59
✓	✓	✓	0.33	26.58	27.57

ble IV. Our decoding-based dual-way propagation can achieve 27.36dB. With the long short-term connections, our model gains +0.23 dB without significant latency improvement. This proves that gathering states from different temporal locations significantly enhances the recovery of the entire sequence. Moreover, the joint update mechanism enables parallel inference, allowing us to enhance reconstruction efficiency with similar performance.

2) Ablation on Metadata-Driven Alignment Module: As shown in Table V, we examine the optical flow estimation of our MDA with different optical flow modules. We compare with a most common baseline SPyNet [20] and denote it as OF. Since motion vectors are inaccurate, sparse, and distinct from feature-level optical flows, OF outperforms the motion vectors significantly. Our refinement network, although much lighter in parameter than OF, can achieve similar results to OF even with only image input. Furthermore, after feeding motion vectors into our network, it surpasses OF on various metrics. This improvement can be attributed to our ability to refine coarse motion vectors to dense optical flows for feature alignment efficiently.

VI. CONCLUSION

We propose a Compression-Omniscient Video Super-Resolution (COVSR) method, specifically designed to enhance the resolution of random access compressed videos efficiently. Two innovative designs are introduced in our method: efficient compression-omniscient propagation and metadata-driven alignment module. By taking the video codec metadata and compressed video scenario into account, efficient compression-omniscient propagation offers several advantages over existing recurrent propagation, such as comprehensive compatibility of decoding timestamps, the inverse connection, and the incorporation of parallel computation. Metadata-driven alignment module enables more accurate estimation of optical flow with a lightweight network, enhancing both performance and latency. These two designs together address the issues of underutilized metadata, mismatched reference state, and long latency that are commonly found in traditional compressed video super-resolution methods. Experimental results demonstrate that our method not only outperforms state-of-the-art compressed VSR methods but also achieves a large speedup in inference. Moreover, it can be used in general low-delay configuration and random access configuration.

TABLE V
ABLATION STUDY OF THE PROPOSED OPTICAL FLOW ESTIMATION OF METADATA-DRIVEN ALIGNMENT MODULE (MDA). MV AND IM INDICATE THE MOTION VECTORS AND IMAGES THAT WILL BE FED INTO MDA.

OF	MV	IM	Params(M)	FLOPs(G)	Latency(ms)	PSNR(dB)
✓			4.1	0.35	32.89	27.44
	✓		2.7	0.32	23.54	27.21
		✓	3.1	0.33	26.43	27.41
	✓	✓	3.3	0.33	26.58	27.57

REFERENCES

[1] Y. Li, P. Jin, F. Yang, C. Liu, M.-H. Yang, and P. Milanfar, "Comisr: Compression-informed video super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2543–2552.

[2] Z. Qiu, H. Yang, J. Fu, and D. Fu, "Learning spatiotemporal frequency-transformer for compressed video super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 257–273.

[3] Y. Wang, T. Isobe, X. Jia, X. Tao, H. Lu, and Y.-W. Tai, "Compression-Aware Video Super-Resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2012–2021.

[4] H. Zhang, X. Zou, J. Guo, Y. Yan, R. Xie, and L. Song, "A codec information assisted framework for efficient compressed video super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 220–235.

[5] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6626–6634.

[6] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "Basicvsr: The search for essential components in video super-resolution and beyond," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4947–4956.

[7] P. Yi, Z. Wang, K. Jiang, J. Jiang, T. Lu, and J. Ma, "Omniscient video super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4429–4438.

[8] K. C. K. Chan, S. Zhou, X. Xu, and C. C. Loy, "Basicvsr++: Improving video super-resolution with enhanced propagation and alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5972–5981.

[9] S. Zhou, X. Jiang, W. Tan, R. He, and B. Yan, "MVFlow: Deep Optical Flow Estimation of Compressed Videos with Motion Vector Prior," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 1964–1974.

[10] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. M. Lee, "Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 0–0.

[11] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, pp. 1106–1125, 2019.

[12] C. Liu and D. Sun, "On Bayesian adaptive video super resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 346–360, 2013.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. 1st Int. Conf. Image Process.*, 1994, vol. 2, pp. 168–172.

[15] S. Shi, J. Gu, L. Xie, X. Wang, Y. Yang, and C. Dong, "Rethinking alignment in video super-resolution transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36081–36093, 2022.

[16] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 0–0.

[17] K. C. K. Chan, S. Zhou, X. Xu, and C. C. Loy, "Investigating tradeoffs in real-world video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5962–5971.

[18] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, "Video super-resolution with recurrent structure-detail network," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 645–660.

[19] P. Chen, W. Yang, M. Wang, L. Sun, K. Hu, and S. Wang, "Compressed domain deep video super-resolution," *IEEE Trans. Image Process.*, vol. 30, pp. 7156–7169, 2021.

[20] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4161–4170.

[21] T. Isobe, X. Jia, X. Tao, C. Li, R. Li, Y. Shi, J. Mu, H. Lu, and Y.-W. Tai, "Look back and forth: Video super-resolution with explicit temporal difference modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17411–17420.

[22] C. Liu, H. Yang, J. Fu, and X. Qian, "Learning trajectory-aware transformer for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5687–5696.

[23] T. Isobe, S. Li, X. Jia, S. Yuan, G. Slabaugh, C. Xu, Y.-L. Li, S. Wang, and Q. Tian, "Video super-resolution with temporal group attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8008–8017.

[24] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.

[25] T. Isobe, F. Zhu, X. Jia, and S. Wang, "Revisiting temporal modeling for video super-resolution," *arXiv preprint arXiv:2008.05765*, 2020.

[26] W. Li, X. Tao, T. Guo, L. Qi, J. Lu, and J. Jia, "Mucan: Multi-correspondence aggregation network for video super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 335–351.

[27] D. Fuoli, S. Gu, and R. Timofte, "Efficient video super-resolution through recurrent latent space propagation," in *Proc. 2019 IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3476–3485.

[28] B. N. Chiche, A. Woiselle, J. Frontera-Pons, and J.-L. Starck, "Stable long-term recurrent video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 837–846.

[29] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1015–1028, 2017.

[30] B. Xia, J. He, Y. Zhang, Y. Wang, Y. Tian, W. Yang, and L. Van Gool, "Structured sparsity learning for efficient video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22638–22647.

[31] J. Liang, Y. Fan, X. Xiang, R. Ranjan, E. Ilg, S. Green, J. Cao, K. Zhang, R. Timofte, and L. Van Gool, "Recurrent video restoration transformer with guided deformable attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 378–393, 2022.

[32] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, "Vrt: A video restoration transformer," *IEEE Trans. Image Process.*, 2024.

[33] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3897–3906.

[34] J. Lin, Y. Huang, and L. Wang, "FDAN: Flow-guided deformable alignment network for video super-resolution," *arXiv preprint arXiv:2105.05640*, 2021.

[35] J. Cao, J. Liang, K. Zhang, W. Wang, Q. Wang, Y. Zhang, H. Tang, and L. Van Gool, "Towards interpretable video super-resolution via alternating optimization," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 393–411.

[36] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "Understanding deformable alignment in video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 2, pp. 973–981, 2021.

[37] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4472–4480.

[38] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "Tdan: Temporally-deformable alignment network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3360–3369.

[39] L. Wang, Y. Guo, L. Liu, Z. Lin, X. Deng, and W. An, "Deep video super-resolution using HR optical flow estimation," *IEEE Trans. Image Process.*, vol. 29, pp. 4323–4336, 2020.

[40] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3106–3115.

[41] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3224–3232.

[42] X. Zhou, L. Zhang, X. Zhao, K. Wang, L. Li, and S. Gu, "Video Super-Resolution Transformer with Masked Inter&Intra-Frame Attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 25399-25408.

[43] P. Chen, W. Yang, M. Wang, L. Sun, K. Hu, and S. Wang, "Compressed domain deep video super-resolution," *IEEE Trans. Image Process.*, vol. 30, pp. 7156-7169, 2021.

[44] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," in *Proc. 16th European Conference on Computer Vision (ECCV)*, Glasgow, UK, Aug. 2020, pp. 402–419.

[45] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *in Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, pp. 562-570, 2015.