Compressed-Domain-Aware Online Video Super-Resolution

Yuhang Wang[1], Hai Li[1,2], Shujuan Hou[1,2,*], Zhetao Dong[1], Xiaoyao Yang[1]
[1]School of Information and Electronics, Beijing Institute of Technology, Beijing, China
[2]Terahertz Science and Application Center, Beijing Institute of Technology, Zhuhai, Guangdong, China
{yuhangwang, haili, shujuanhou, 3120230750, xyyang}@bit.edu.cn

Abstract

In bandwidth-limited online video streaming, videos are usually downsampled and compressed. Although recent online video super-resolution (online VSR) approaches achieve promising results, they are still compute-intensive and fall short of real-time processing at higher resolutions, due to complex motion estimation for alignment and redundant processing of consecutive frames. To address these issues, we propose a compressed-domain-aware network (CDA-VSR) for online VSR, which utilizes compressed-domain information, including motion vectors, residual maps, and frame types to balance quality and efficiency. Specifically, we propose a motion-vector-guided deformable alignment module that uses motion vectors for coarse warping and learns only local residual offsets for fine-tuned adjustments, thereby maintaining accuracy while reducing computation. Then, we utilize a residual map gated fusion module to derive spatial weights from residual maps, suppressing mismatched regions and emphasizing reliable details. Further, we design a frame-type-aware reconstruction module for adaptive compute allocation across frame types, balancing accuracy and efficiency. On the REDS4 dataset, our CDA-VSR surpasses the state-of-the-art method TMP, with a maximum PSNR improvement of ***0.13 dB*** *while delivering more than* ***double*** *the inference speed. The code will be released at* https://github.com/sspBIT/CDA-VSR.

1. Introduction

Video super-resolution (VSR) aims to reconstruct the high-resolution (HR) video sequence from low-resolution (LR) frames. With the rise of online applications such as video conferencing and live streaming, online VSR has garnered increasing attention [11, 45, 50, 53]. In online VSR, the current frame is enhanced using only past and current frames under a strict time budget.

*Corresponding author

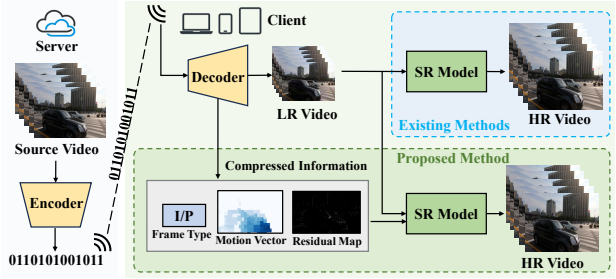

Figure 1. Comparison between existing online VSR methods and our proposed method. The server downsamples and encodes the source video, and then transmits the compressed stream. The client decodes and performs super-resolution. Our method uses compressed-domain information, including frame type, motion vectors, and residual maps, to guide alignment, fusion, and reconstruction, improving both accuracy and efficiency.

Recently, a series of online VSR methods have been proposed [12, 13, 27, 29, 34, 45, 57]. Flow-based alignment methods [29, 44] improve super-resolution (SR) quality by accurately aligning frames using optical flow, but optical flow estimation is computationally intensive. Implicit alignment methods [10, 12, 13] improve efficiency at the expense of reconstruction quality, most notably under large motions.

To better balance accuracy and efficiency, recent works explore efficient alignment and fusion modules [11, 14, 34, 45, 53]. DAP [11] uses a deformable attention pyramid to efficiently align features from the previous frame with those of the current frame. TMP [53] leverages the motion continuity between frames, propagates the offsets across frames and refines them locally, thus avoiding redundant computations. However, these methods still struggle with complex motion estimation and redundant processing of consecutive frames, leading to a heavy computational burden, particularly at higher resolutions such as 2K.

These methods typically rely solely on the LR video frames and do not exploit the valuable compressed-domain information such as motion vectors, residual maps, and frame types, readily available in the bitstream. Motion

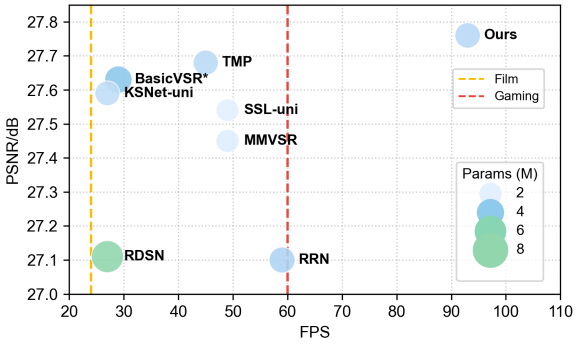

Figure 2. PSNR, FPS, and Parameters of different methods on REDS4 for 4× upscaling at CRF=18.

vectors describe coarse inter-frame motion, residual maps reflect high-frequency differences, and frame types determine inter-frame reference relationships. Leveraging this extra information can further improve both accuracy and efficiency in online VSR.

Building upon this idea, we propose a compressed-domain-aware online VSR framework (CDA-VSR), as illustrated in Figure 1. Our CDA-VSR is designed with three key modules that exploit the distinct characteristics of motion vectors, residual maps, and frame types. We propose a motion-vector-guided deformable alignment module (MVGDA) that uses motion vectors for coarse alignment, then initializes deformable offsets for local refinement. This design addresses the limitations of flow-based alignment methods (which are computationally expensive) and implicit alignment methods (which struggle with large motion). Then, instead of concatenating inter-frame features [36, 43, 53], which propagates mismatched details from the previous frame, we propose residual map gated fusion (RMGF). The residual map predicts spatial weights that suppress irrelevant regions and emphasize reliable structures to improve reconstruction quality. Moreover, we introduce frame-type-aware reconstruction (FTAR): a high-capacity path for I-frames and a lightweight path for P-frames. This frame-type adaptive allocation preserves keyframe fidelity, avoids redundant computation on P-frames, and significantly enhances real-time processing efficiency.

The main contributions are summarized as follows.

- We propose a compressed-domain-aware online VSR framework (CDA-VSR). By leveraging motion vectors, residual maps, and frame types to guide frame alignment, fusion, and reconstruction, CDA-VSR improves both accuracy and efficiency.
- We design a motion-vector-guided deformable alignment module (MVGDA) and a residual map gated fusion module (RMGF). MVGDA combines coarse motion-vector alignment with local deformable refinement, maintaining pixel-level accuracy with reduced complexity. RMGF uses residual maps to generate spatial weights, suppressing misaligned regions and enhancing detail reliability.
- We propose a frame-type-aware reconstruction strategy (FTAR). I-frames are processed with a high-capacity reconstruction module to preserve global fidelity, while lightweight modules are designed for P-frames to accelerate inference.
- Experiments on the REDS4 dataset show that our method achieves approximately 90 FPS while maintaining visual quality comparable to state-of-the-art methods and delivering $> 2\times$ the inference speed, as illustrated in Figure 2.

2. Related Works

Video Super-Resolution. VSR aims to reconstruct a HR video sequence from degraded LR inputs. Based on the paradigms, VSR methods [8, 19, 22, 30, 40, 46, 47, 49, 55] can be roughly grouped into sliding-window based VSR and recurrent based VSR. Sliding-window based VSR [1, 15, 18, 35, 38, 40, 48, 49] uses a fixed set of neighboring frames to reconstruct one or more target frames. The available information is constrained by the window size, so these methods can only exploit temporal details within a restricted subset of the video. To exploit a longer temporal context, recurrent based VSR methods [2, 3, 10, 12, 13, 21, 29, 31, 52] reuse information by propagating hidden states and reconstructed frames across time. Bidirectional propagation methods [2, 3, 32, 56] further boost accuracy by passing information from both past and future frames. By leveraging many support frames, these methods typically achieve higher accuracy at the cost of increased latency. Unidirectional propagation methods [10, 12, 13, 21, 29] aggregate the information from the past and current frames, as well as several cached future frames, thereby improving efficiency. When only past and current frames are available, unidirectional propagation methods are suitable for online VSR.

Online Video Super-Resolution. Online VSR requires real-time reconstruction of the current frame during video playback [45]. This imposes causal constraints (only past and current frames can be used) and demands low latency, in contrast to offline VSR, which can exploit bidirectional information. Early online strategies apply lightweight single-frame SR methods [6, 7, 20, 25, 37, 54]. By processing each frame independently, they fail to exploit temporal redundancy, leading to limited improvements in reconstruction quality. Later works explore unidirectional recurrent architectures that reuse information from previous frames, such as RSDN [12] and RRN [13]. Recent works [11, 14, 34, 45, 53, 57] improve online VSR mainly through enhanced alignment and fusion modules. KSNet [14] uses a multi-flow deformable alignment module and a kernel-split strategy. TMP [53] exploits inter-frame motion similarity, propagates estimated motion fields across frames,

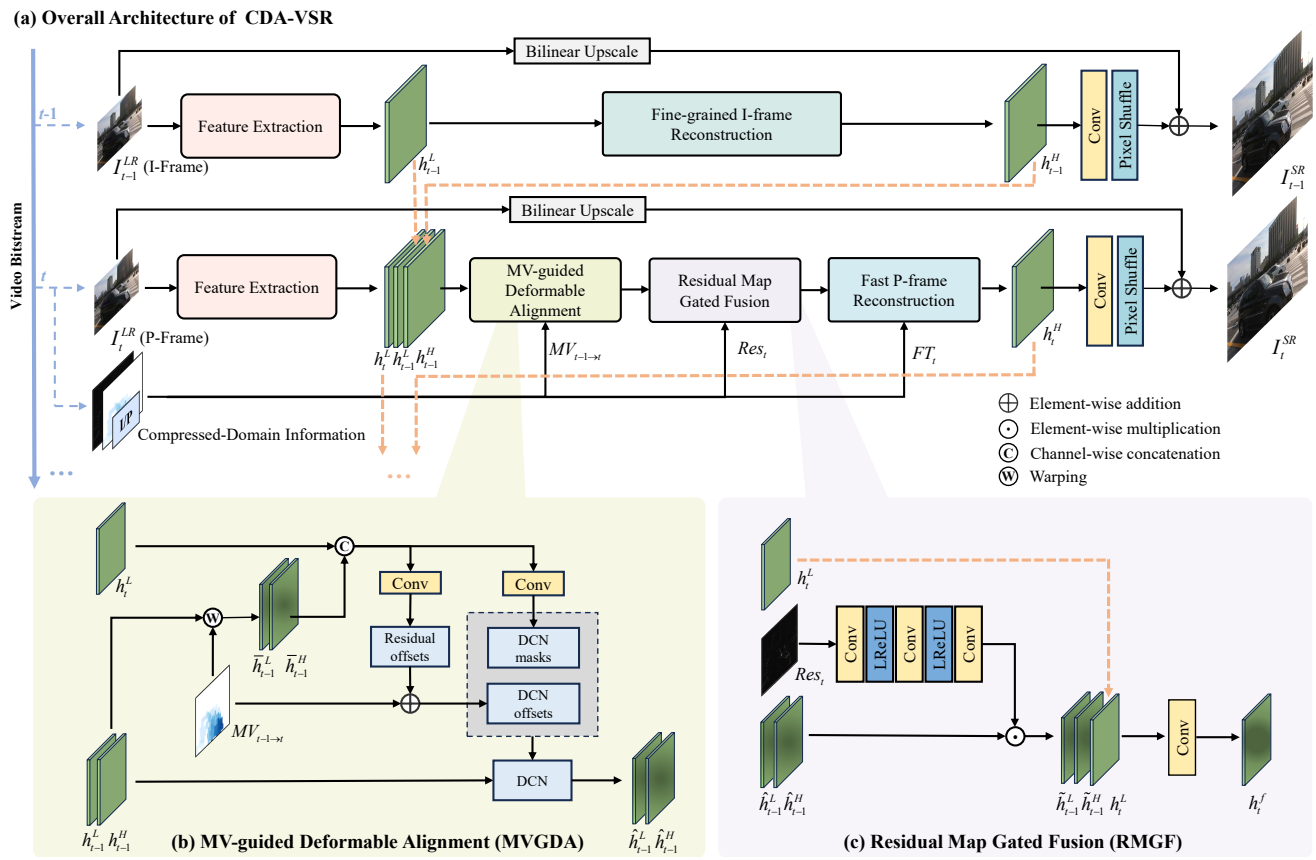

Figure 3. Overall architecture of the proposed Compressed-Domain-Aware VSR framework (CDA-VSR). Given LR frames and compressed-domain information (motion vectors, residual maps, and frame types), CDA-VSR reconstructs the corresponding HR frames through three key modules: (1) the MV-guided Deformable Alignment (MVGDA); (2) the Residual Map Gated Fusion (RMGF); (3) the Frame-Type-Aware Reconstruction (FTAR) with two branches, Fine-grained I-Frame Reconstruction and Fast P-Frame Reconstruction.

and incrementally refines them, thereby avoiding redundant motion estimation for each frame. MMVSR [34] performs dynamic-static decoupled alignment and fuses multi-memory streams to improve long-range temporal modeling. However, existing approaches still incur substantial compute and latency due to complex motion estimation for alignment and redundant processing across consecutive frames, which are amplified at higher resolutions (e.g., 2K).

Compressed-Domain Information for VSR. Recently, many vision tasks [4, 9, 24, 39] have benefited from compressed-domain information. Similarly, a few methods have attempted to incorporate such information into VSR [5, 41, 51]. CDVSR [5] fuses bitstream coding priors with deep SR models to better restore textures and details. CIAF [51] uses motion vectors as optical-flow proxies and residual maps to identify static regions, thereby skipping redundant processing and improving efficiency. CAVSR [41] employs a compression encoder to extract compression-level features and a modulation module to adapt SR to varying compression strengths. These methods show that incorporating compressed-domain information (e.g., motion vectors, residual maps) can improve VSR performance. However, as they are not designed for online VSR, their inference speed remains insufficient for real-time applications. In addition, existing methods have not fully explored dedicated designs for different types of compressed-domain information, which limits the potential gains achievable from such information. In contrast, CDA-VSR tailors specialized modules to the characteristics of each type of compressed-domain information, while explicitly satisfying the causality and real-time constraints of online VSR.

3. Methodology

3.1. Overall Architecture

Online VSR aims to reconstruct the HR reference frame $I_t^{HR} \in \mathbb{R}^{sH \times sW \times 3}$ from its corresponding LR frame $I_t^{LR} \in \mathbb{R}^{H \times W \times 3}$ and the N supporting frames $I_{[t-N:t-1]}^{LR}$, where s denotes the upsampling factor, H and W denote the height and width of LR frames, and t represents the times-

tamp of the video stream. Unlike existing online VSR methods that only exploit decoded LR frames, our CDA-VSR additionally leverages compressed-domain priors from the bitstream, including motion vectors $MV_{t-1\to t}$, residual maps Res_t, and frame types FT_t, as shown in Fig. 3. These priors provide motion cues for feature alignment, suppress misaligned regions for more reliable fusion, and distinguish frame types for adaptive reconstruction.

Building on earlier works [11, 53], CDA-VSR adopts a recurrent structure, which enhances computational efficiency and meets the requirements of real-time processing. Following existing works [11, 53], we adopt a shallow feature extraction network to map each decoded LR frame into latent features h_t^L. CDA-VSR then consists of three key modules designed to leverage the characteristics of different compressed-domain information. First, the MV-guided deformable alignment module (MVGDA) employs motion vectors for coarse alignment, followed by a lightweight deformable convolution to refine local misalignments. Second, the residual map gated fusion module (RMGF) generates spatially varying weights from residual maps to suppress irrelevant details and enhance reliable regions. Third, the frame-type-aware reconstruction (FTAR) adaptively allocates computational resources. I-frames are reconstructed using a fine-grained branch to preserve global fidelity, while P-frames are processed by a lightweight branch to improve inference speed. These three modules are described in detail in Sections 3.2, 3.3, and 3.4.

In summary, CDA-VSR exploits the complementary roles of motion vectors, residual maps, and frame types to achieve a superior trade-off between reconstruction quality and computational efficiency, enabling real-time online VSR.

3.2. MV-guided Deformable Alignment (MVGDA)

Accurate and efficient frame alignment is crucial in online VSR to exploit temporal redundancy and improve reconstruction quality. Flow-based alignment explicitly estimates optical flow to warp frames or features, but is computationally expensive and sensitive to flow errors, whereas deformable convolutions implicitly compensate motion via learned sampling offsets but still struggle with large or complex displacements due to unconstrained offsets.

To address these limitations, we leverage motion vectors (MVs) extracted directly from the video bitstream as temporal priors. MVs describe block-level displacements between adjacent frames and can be obtained essentially for free during decoding, providing a computationally efficient alternative to optical flow. Let h_{t-1} and h_t denote the features of the previous and current frames, respectively. We first perform coarse alignment by warping the previous-frame features h_{t-1} using MVs:

$$\overline{h}_{t-1} = \mathcal{W}(h_{t-1}; MV_{t-1\to t}), \tag{1}$$

where $\mathcal{W}(\cdot)$ denotes the warping operator. This step efficiently compensates large inter-frame motion.

Although MVs provide useful motion priors, their block-wise nature forces all pixels within a block to share a single motion vector, ignoring intra-block motion variations. As a result, MVs become unreliable near object boundaries and in regions with complex or non-rigid motion. To mitigate this, we embed MVs into a deformable convolution (DCN). Specifically, the DCN offsets are initialized with motion vectors o_{MV}, and a lightweight convolutional network predicts residual offsets Δo to further refine them. In parallel, a modulation mask m is predicted to adaptively reweight the sampling locations:

$$\Delta o = \mathcal{C}^o([h_t, \overline{h}_{t-1}]), \tag{2}$$

$$m = \sigma(\mathcal{C}^m([h_t, \overline{h}_{t-1}])), \tag{3}$$

where $\mathcal{C}^o$ and $\mathcal{C}^m$ denote convolutional subnetworks, $[\cdot,\cdot]$ represents channel-wise concatenation, and $\sigma(\cdot)$ is the sigmoid function. The aligned features $\hat{h}_{t-1}$ are then obtained by applying DCN to the previous unwarped features h_{t-1}:

$$\hat{h}_{t-1} = \mathcal{D}\big(h_{t-1}; o_{MV} + \Delta o, m\big), \tag{4}$$

where $\mathcal{D}$ denotes DCN. In this way, MVGDA uses motion vectors for efficient coarse alignment, while the deformable convolution only needs to learn local residual offsets, which simplifies offset learning and leads to more robust and efficient alignment under large and complex motions.

So far, we have described alignment for a single feature from the previous frame. In practice, our CDA-VSR exploits two complementary feature representations: coarse features h_{t-1}^L from the encoder and fine-grained features h_{t-1}^H from the reconstruction module. The former provide robust structural priors that are less affected by reconstruction noise, while the latter carry rich texture details that enhance fidelity. Both types of features are aligned using MVGDA.

$$\overline{h}_{t-1}^L, \overline{h}_{t-1}^H = \mathcal{W}(h_{t-1}^L, h_{t-1}^H; MV_{t-1\to t}), \tag{5}$$

$$\Delta o = \mathcal{C}^o\big([h_t^L, \overline{h}_{t-1}^L, \overline{h}_{t-1}^H]\big), \tag{6}$$

$$m = \sigma(\mathcal{C}^m\big([h_t^L, \overline{h}_{t-1}^L, \overline{h}_{t-1}^H]\big)), \tag{7}$$

$$\hat{h}_{t-1}^L, \hat{h}_{t-1}^H = \mathcal{D}\big(h_{t-1}^L, h_{t-1}^H; o_{MV} + \Delta o, m\big). \tag{8}$$

The two aligned features $\hat{h}_{t-1}^L$ and $\hat{h}_{t-1}^H$ are then jointly propagated to the current step, enabling the network to benefit from both stable global structures and detailed textures.

3.3. Residual Map Gated Fusion (RMGF)

After aligning features across frames with MVGDA, the next step is to effectively fuse information from the previous

and current frames. A simple strategy is to concatenate the aligned features along the channel dimension [36, 43, 53]. However, misaligned or inconsistent regions from the previous frame can propagate errors and degrade reconstruction quality. Therefore, it is crucial to selectively exploit reliable regions from the previous frame while suppressing misleading ones.

Residual maps Res_t obtained from the video bitstream represent the pixel-wise difference between the current frame and its motion-compensated prediction from reference frames. Large residual values indicate regions where motion compensation fails, such as occlusion boundaries or regions with complex local motion. Therefore, Res_t naturally highlights temporally inconsistent regions and provides a useful cue for guiding feature fusion.

To leverage this cue, we design a residual map gated fusion module (RMGF). A lightweight network $\mathcal{F}_{\text{res}}(\cdot)$ first transforms Res_t into a spatial gating map:

$$M_t = \sigma(\mathcal{F}_{\text{res}}(Res_t)), \tag{9}$$

where $\sigma(\cdot)$ denotes the sigmoid function. The gate M_t suppresses unreliable regions in the aligned features $\hat{h}^L_{t-1}, \hat{h}^H_{t-1}$, and the fused feature h^f_t is obtained as:

$$h^f_t = \mathcal{C}^f([M_t \odot \hat{h}^L_{t-1},\ M_t \odot \hat{h}^H_{t-1},\ h^L_t]), \tag{10}$$

where $\mathcal{C}^f$ is a 3×3 convolution and $\odot$ denotes element-wise multiplication. In this way, the current-frame features h^L_t serve as a stable baseline, while temporal features from previous frames contribute only in regions where they are reliable. By preserving trustworthy temporal information and down-weighting misaligned regions, RMGF improves the overall reconstruction quality.

3.4. Frame-Type-Aware Reconstruction (FTAR)

Videos are typically encoded as a mixture of intra-coded frames (I-frames) and predictive frames (P-frames). As we target online streaming, B-frames (requiring future frames) are not considered. I-frames contain full spatial information and serve as key references for subsequent decoding, whereas P-frames mainly store incremental updates with respect to previously decoded frames and occur much more frequently. In online VSR, reconstructing all frames with the same computational budget is inefficient: allocating excessive computation to P-frames leads to redundant cost, while insufficient modeling of I-frames may degrade the quality of the entire sequence.

To balance accuracy and efficiency, we propose a frame-type-aware reconstruction (FTAR) strategy that allocates computation according to the frame type. For I-frames, we employ a fine-grained reconstruction branch $\mathcal{R}_I$ with higher capacity to preserve global structures and maximize visual fidelity:

$$I^{SR}_t = \mathcal{R}_I(h^L_t), \quad \text{if } FT_t = \text{I}, \tag{11}$$

where h^L_t denotes the features of the current LR frame. For P-frames, we adopt a fast reconstruction branch $\mathcal{R}_P$ to accelerate inference while maintaining sufficient detail:

$$I^{SR}_t = \mathcal{R}_P(h^f_t), \quad \text{if } FT_t = \text{P}, \tag{12}$$

where h^f_t is the fused feature obtained from RMGF.

Concretely, we use the depth of residual blocks as a proxy for computational complexity and instantiate a fine-grained branch with m blocks for I-frames and a lightweight branch with n $(n < m)$ blocks for P-frames. This adaptive reconstruction strategy ensures that I-frames deliver high-quality details that benefit subsequent temporal propagation, while avoiding redundant computation on the more frequent P-frames.

3.5. Loss Function

Similar to existing works [45, 53, 57], we adopt the widely used Charbonnier loss [17]. Given the predicted high-resolution frame I^{SR}_t and the corresponding ground-truth frame I^{GT}_t, the Charbonnier loss is defined as:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\left(I^{SR}_t - I^{GT}_t\right)^2 + \epsilon^2}, \tag{13}$$

where T is the number of input LR frames and ϵ is a hyper-parameter.

4. Experiments

4.1. Experimental Settings

Datasets. The REDS dataset [28] is employed for training. This dataset contains 720p video sequences with diverse scenes and large inter-frame motion, which makes it particularly suitable for video super-resolution. For evaluation, the commonly used REDS4 dataset [28] is adopted. High-resolution frames are downsampled by $\times 4$ to form LR inputs and then encoded with H.264 codec [42] in capped constant rate factor (CRF) mode (CRF 18/23/28) using FFmpeg. This setting limits excessive bitrate and better reflects online video conditions. During decoding, we parse the bitstream to extract motion vectors, residual maps, and frame types as auxiliary information. To further evaluate the robustness of the proposed CDA-VSR across different resolutions, four sequences are randomly selected from the Inter4K dataset [33], each containing 300 frames with resolutions of 2K, 1080p, and 720p. Following the same processing and compression settings as REDS4, the sequences are encoded using Capped-CRF with CRF=23.

Implementation Details. We adopt 3 residual blocks (RBs) [23] for feature extraction. For reconstruction, 24 RBs are employed for I-frames, while 12 RBs are used for P-frames. The number of channels in the convolutional layer is 64. The model is trained using the Adam [16] optimizer with

Table 1. Comparison with state-of-the-art online VSR methods on the REDS4 dataset. Runtime is measured on a single NVIDIA RTX 3090 with LR inputs of 320×180. Following practical usage, methods with FPS ≥24 are regarded as film real-time, and FPS ≥60 as gaming real-time. The best and the second best results are colored with red and blue.

Method	Film R.T. (FPS≥24)	Game R.T. (FPS≥60)	Runtime ↓ (ms)	FPS ↑ (1/s)	Params ↓ (M)	MACs ↓ (G)	CRF18 (PSNR(dB)↑ / SSIM↑ / LPIPS↓) clip000	clip011	clip015	clip020	Avg	CRF23 Avg	CRF28 Avg
BasicVSR* [2]	✓	✗	34.5	29	4.0	254	25.72 0.7040 0.3435	27.91 0.7739 0.3371	30.26 0.8432 0.3239	26.62 0.7730 0.3343	27.63 0.7735 0.3347	26.54 0.7326 0.3809	25.13 0.6795 0.4335
RRN [13]	✓	✗	16.9	59	3.4	193	25.21 0.6748 0.3645	27.32 0.7546 0.3518	29.76 0.8223 0.3311	26.11 0.7553 0.3480	27.10 0.7518 0.3489	26.22 0.7192 0.3915	24.96 0.6718 0.4410
RSDN [12]	✓	✗	37.0	27	6.2	356	25.27 0.6782 0.3634	27.31 0.7546 0.3534	29.76 0.8325 0.3305	26.11 0.7550 0.3492	27.11 0.7551 0.3491	26.22 0.7198 0.3922	24.96 0.6721 0.4414
SSL-uni [44]	✓	✗	20.4	49	2.2	92	25.64 0.7013 0.3489	27.84 0.7712 0.3430	30.17 0.8420 0.3281	26.49 0.7684 0.3434	27.54 0.7707 0.3409	26.48 0.7302 0.3872	25.09 0.6781 0.4395
KSNet-uni [14]	✓	✗	29.4	34	3.0	148	25.71 0.7035 0.3376	27.97 0.7724 0.3304	29.93 0.8302 0.3312	26.72 0.7717 0.3307	27.58 0.7695 0.3325	26.57 0.7327 0.3824	25.12 0.6749 0.4327
MMVSR [34]	✓	✗	23.2	43	2.3	122	25.55 0.6911 0.3677	27.72 0.7666 0.3498	30.08 0.8396 0.3346	26.44 0.7648 0.3501	27.45 0.7655 0.3506	26.42 0.7257 0.3961	25.04 0.6740 0.4478
TMP [53]	✓	✗	22.2	45	3.1	176	25.70 0.7036 0.3371	27.99 0.7750 0.3303	30.31 0.8437 0.3181	26.73 0.7764 0.3235	27.68 0.7747 0.3273	26.58 0.7336 0.3748	25.17 0.6805 0.4288
CDA-VSR (Ours)	✓	✓	10.8	93	3.3	78	25.81 0.7085 0.3324	28.11 0.7788 0.3285	30.32 0.8455 0.3171	26.79 0.7788 0.3223	27.76 0.7779 0.3251	26.70 0.7384 0.3705	25.30 0.6869 0.4230

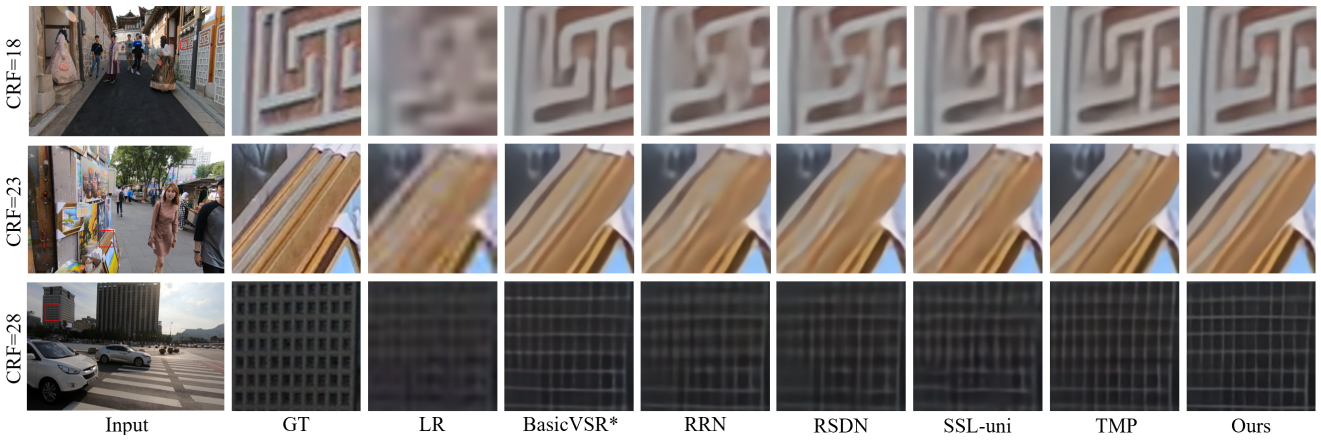

Figure 4. Qualitative comparison of different online VSR methods on the REDS4 dataset.

$\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rate is initialized to 2×10^{-4} and gradually decayed with the Cosine Annealing scheme [26]. We train on 15-frame clips with 64×64 random crops and random horizontal flips/rotations, for 300k iterations with batch size 8. The model is implemented in PyTorch and trained on an NVIDIA RTX 3090.

4.2. Experiment on the REDS4 Dataset

We evaluate the proposed CDA-VSR against several open-source state-of-the-art online VSR methods, including RRN [13], RSDN [12], SSL-uni [44], KSNet-uni [14], MMVSR [34] and TMP [53]. Additionally, we construct a variant *BasicVSR** by eliminating the backward propagation branch of BasicVSR [2] in order to satisfy the causality and latency requirements of online VSR applications. All methods are trained and evaluated under the same experimental settings as our method to ensure a fair comparison.

Quantitative results. Table 1 shows the quantitative comparison results under three compression levels (CRF18/23/28). As shown, our CDA-VSR achieves the best PSNR/SSIM across all compression levels while attaining the lowest runtime and MACs. At CRF28, our CDA-VSR

Table 2. Comparison with state-of-the-art online VSR methods on the Inter4K dataset at different resolutions. The best and the second best results are colored with red and blue.

Method	PSNR (dB)↑			SSIM↑			FPS (1/s)↑		
	720p	1080p	2K	720p	1080p	2K	720p	1080p	2K
BasicVSR* [2]	26.88	28.36	29.73	0.7941	0.8454	0.8816	28.6	13.6	7.7
RRN [13]	26.61	28.07	29.33	0.7821	0.8353	0.8707	58.3	26.5	15.2
RSDN [12]	26.65	28.07	29.29	0.7839	0.8361	0.8709	26.2	11.9	6.7
SSL-uni [44]	26.83	28.15	29.58	0.7916	0.8429	0.8783	48.1	27.1	16.9
KSNet-uni [14]	26.85	28.28	29.61	0.7905	0.8405	0.8768	34.0	17.8	10.6
MMVSR [34]	26.72	28.20	29.51	0.7852	0.8388	0.8748	43.1	21.1	12.4
TMP [53]	26.95	28.45	29.76	0.7959	0.8473	0.8822	45.7	20.8	11.4
CDA-VSR (Ours)	27.13	28.64	29.98	0.8022	0.8525	0.8868	92.6	44.2	25.1

outperforms TMP by +0.13 dB and exceeds its speed by over 2×. The number of parameters in CDA-VSR is higher because the reconstruction module has two branches. During inference, only one of these branches is activated for each frame depending on its type, so the effective runtime and computational cost remain minimal.

Qualitative results. We conduct visual comparisons with several representative methods on the REDS4 dataset, as shown in Figure 4. Implicit alignment methods (e.g., RRN and RSDN) tend to produce blurry edges and lose fine structures. BasicVSR*, SSL-uni, and TMP preserve sharper boundaries than implicit alignment methods, but still fail to recover fine textures. In contrast, our CDA-VSR reconstructs clearer edges and more detailed textures.

4.3. Experiments on the Inter4K Dataset at Different Resolutions

To further evaluate the robustness of our method under higher resolutions, we conduct experiments on the Inter4K dataset, which contains sequences at 720p, 1080p, and 2K resolutions. Table 2 reports PSNR, SSIM, and FPS. Our method consistently achieves the best PSNR and SSIM at all resolutions. At 2K, our CDA-VSR achieves 29.98 dB, outperforming TMP by +0.22 dB. In terms of efficiency, the advantage grows with resolution: at 1080p we maintain film real-time with a clear margin, and at 2K we still exceed the 24 FPS threshold while all other methods fall well below it. Overall, these results confirm that CDA-VSR achieves a superior trade-off between reconstruction quality and runtime efficiency.

4.4. Ablation Studies

In this section, we conduct ablation experiments on three key components of CDA-VSR: the MV-guided deformable alignment (MVGDA), the residual map gated fusion (RMGF), and the frame-type-aware reconstruction (FTAR). All models are trained on the REDS dataset and evaluated on REDS4.

Effectiveness of MVGDA. To validate the effectiveness of the MV-guided deformable alignment (MVGDA), we design three variants: *OnlyMV* with motion-vector warping only, *OnlyDCN* with deformable convolution only, and *OnlyGL* with optical flow warping only. All are retrained under the same settings, and the results are given in Table 3. *OnlyMV* is fastest but drops 0.17 dB at CRF18. *OnlyDCN* drops 0.41 dB at CRF18. *OnlyGL* attains competitive quality, but requires about 1.4× the runtime of our method. MVGDA achieves the best quality with low runtime, striking a favorable balance between accuracy and efficiency. To further explain these results, we visualize the feature maps before and after alignment, as illustrated in Figure 5. *OnlyDCN* struggles with large motions, leading to blurred and misaligned structures in dynamic regions. *OnlyMV* provides efficient global motion alignment but produces block-level discontinuities at object boundaries, as indicated by the red arrows. *OnlyGL* provides accurate motion compensation and preserves fine structures; however, residual misalignments persist in the regions indicated by the green arrows. In contrast, our MVGDA produces cleaner and more consistent features across frames, effectively reducing misalignment errors.

Table 3. Ablation study on the MV-guided deformable alignment (MVGDA) using the REDS4 dataset. The best results are colored with red.

Method	Runtime↓ (ms)	Params↓ (M)	PSNR(dB)↑ / SSIM↑		
			CRF18	CRF23	CRF28
OnlyMV	10.2	3.17	27.59/0.7724	26.56/0.7332	25.19/0.6825
OnlyDCN	10.6	3.28	27.35/0.7638	26.38/0.7263	25.06/0.6769
OnlyGL	15.5	4.61	27.73/0.7761	26.66/0.7364	25.25/0.6844
MVGDA (Ours)	10.8	3.28	27.76/0.7779	26.70/0.7384	25.30/0.6869

Table 4. Ablation study on the residual map gated fusion (RMGF) using the REDS4 dataset. The best results are colored with red.

Method	Runtime↓ (ms)	Params↓ (M)	PSNR(dB)↑ / SSIM↑		
			CRF18	CRF23	CRF28
NoGate	10.8	3.26	27.63/0.7739	26.60/0.7347	25.22/0.6838
RMGF (Ours)	10.8	3.28	27.76/0.7779	26.70/0.7384	25.30/0.6869

Effectiveness of RMGF. To assess the residual map gated fusion (RMGF), we build a no-gating variant (named *NoGate*). As shown in Table 4, RMGF outperforms NoGate at all compression levels. At CRF18, NoGate attains 27.63 dB, which is 0.13 dB lower than our full model. The gain comes from gating that regulates temporal fusion: simple concatenation propagates misaligned details, whereas RMGF upweights consistent regions and suppresses unreliable ones. To further corroborate this, we visualize the gate weights as heatmaps. As shown in Figure 6, the gate assigns higher weights to previous-frame features in well-aligned, temporally stable regions (e.g., the car body) while suppressing misaligned areas such as the rotating wheels indicated by the green arrows. These results confirm that residual maps provide a valuable cue for selective temporal fusion, improving reconstruction quality.

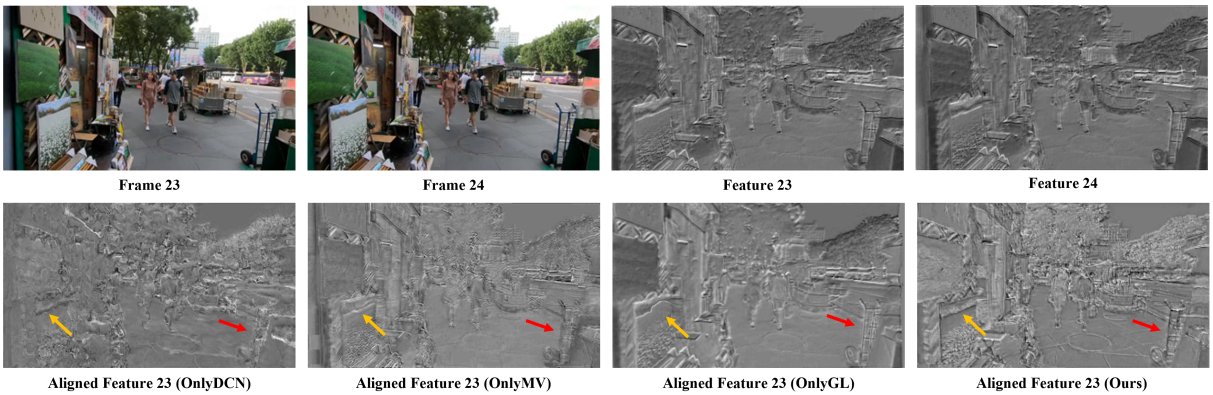

Figure 5. Feature map visualization of different alignment methods. Three variants (*OnlyMV*, *OnlyDCN*, *OnlyGL*) and our MV-guided deformable alignment (MVGDA) are compared by visualizing the feature maps before and after alignment. For clarity, only the first channel of each feature map is shown.

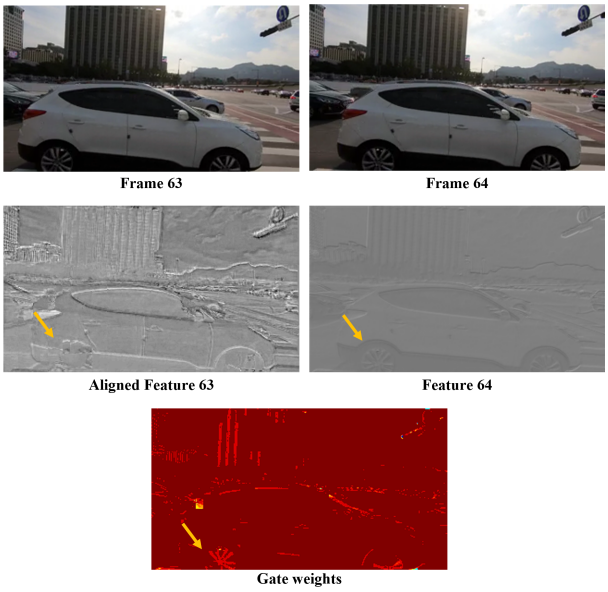

Figure 6. Heatmap visualization of gated fusion weights. Warmer colors (red) indicate larger gating values. The gate performs spatially selective fusion by assigning strong responses to stable regions while suppressing misaligned areas (green arrows).

Table 5. Ablation study on the frame-type-aware reconstruction (FTAR) using the REDS4 dataset.

Method	Runtime↓ (ms)	PSNR(dB)↑ / SSIM↑		
		CRF18	CRF23	CRF28
I=12,P=12	10.7	27.60/0.7724	26.57/0.7334	25.19/0.6824
I=24,P=24	16.8	27.80/0.7786	26.74/0.7390	25.34/0.6875
I=24,P=12 (Ours)	10.8	27.76/0.7779	26.70/0.7384	25.30/0.6869

Effectiveness of FTAR. We evaluate frame-type-aware reconstruction (FTAR) on the REDS4 dataset. As shown in Table 5, deepening both branches (I=24, P=24) improves PSNR and SSIM over the shallow baseline (I=12, P=12) at all CRFs, but increases latency by 57% (10.7 ms to 16.8 ms). In contrast, the FTAR setting (I=24, P=12) preserves most of the accuracy gains with almost no extra cost: relative to (I=12, P=12), it adds +0.16/+0.13/+0.11 dB PSNR and +0.0055/+0.0050/+0.0045 SSIM, with only +0.1 ms overhead. The consistent trend across CRFs indicates that allocating more capacity to I-frames captures most of the quality benefit, while keeping the P-frame branch lightweight avoids redundant computation and yields a trade-off between accuracy and efficiency for online VSR.

5. Conclusion

In this paper, we propose a compressed-domain-aware framework for online VSR, termed CDA-VSR. Unlike existing methods that operate only on LR frames, CDA-VSR leverages compressed-domain information through three dedicated modules. We develop a motion-vector-guided deformable alignment module that uses motion vectors for coarse warping and learns only local residual offsets, which greatly reduces the cost of offset estimation while maintaining accuracy. We further present a residual map gated fusion module that predicts spatial weights to suppress mismatched regions and emphasize reliable details. We also introduce a frame-type-aware reconstruction scheme that allocates higher capacity to I-frames while keeping a lightweight branch for P-frames to better balance accuracy and efficiency. Extensive experiments demonstrate that CDA-VSR achieves higher super-resolution quality and faster inference speed than current state-of-the-art online VSR methods, delivering more than 2× the FPS. In future work, we plan to extend the use of compressed-domain information to broader video restoration and enhancement tasks, such as artifact removal and temporal interpolation.

References

[1] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4778–4787, 2017. 2

[2] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4947–4956, 2021. 2, 6, 7

[3] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5972–5981, 2022. 2

[4] Liuhong Chen, Heming Sun, Jiro Katto, Xiaoyang Zeng, and Yibo Fan. Fast object detection in hevc intra compressed domain. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 756–760. IEEE, 2021. 3

[5] Peilin Chen, Wenhan Yang, Meng Wang, Long Sun, Kangkang Hu, and Shiqi Wang. Compressed domain deep video super-resolution. *IEEE Transactions on Image Processing*, 30:7156–7169, 2021. 3

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 2

[7] Zhetao Dong, Shujuan Hou, Hai Li, Yuhang Wang, and Ruixue Gao. Lightweight real-world image super-resolution via channel redundancy for edge iot devices. *IEEE Internet of Things Journal*, 2025. 2

[8] Shian Du, Menghan Xia, Chang Liu, Xintao Wang, Jing Wang, Pengfei Wan, Di Zhang, and Xiangyang Ji. Patchvsr: Breaking video diffusion resolution limits with patch-wise video super-resolution. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17799–17809. IEEE, 2025. 2

[9] Xiang Fang, Daizong Liu, Pan Zhou, and Guoshun Nan. You can ground earlier than see: An effective and efficient pipeline for temporal sentence grounding in compressed videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2448–2460, 2023. 3

[10] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3476–3485. IEEE, 2019. 1, 2

[11] Dario Fuoli, Martin Danelljan, Radu Timofte, and Luc Van Gool. Fast online video super-resolution with deformable attention pyramid. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1735–1744, 2023. 1, 2, 4

[12] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *European conference on computer vision*, pages 645–660. Springer, 2020. 1, 2, 6, 7

[13] Takashi Isobe, Fang Zhu, and Shengjin Wang. Revisiting temporal modeling for video super-resolution. In *BMVC*, 2020. 1, 2, 6, 7

[14] Shuo Jin, Meiqin Liu, Chao Yao, Chunyu Lin, and Yao Zhao. Kernel dimension matters: To activate available kernels for real-time video super-resolution. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8617–8625, 2023. 1, 2, 6, 7

[15] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE transactions on computational imaging*, 2(2):109–122, 2016. 2

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[17] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 5

[18] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *European conference on computer vision*, pages 335–351. Springer, 2020. 2

[19] Zekun Li, Hongying Liu, Fanhua Shang, Yuanyuan Liu, Liang Wan, and Wei Feng. Savsr: arbitrary-scale video super-resolution via a learned scale-adaptive network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3288–3296, 2024. 2

[20] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2

[21] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. *Advances in Neural Information Processing Systems*, 35:378–393, 2022. 2

[22] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *IEEE Transactions on Image Processing*, 33:2171–2182, 2024. 2

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 5

[24] Hao Liu, Lijun He, Miao Zhang, and Fan Li. Vadiffusion: Compressed domain information guided conditional diffusion for video anomaly detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(9):8398–8411, 2024. 3

[25] Xin Liu, Jie Liu, Jie Tang, and Gangshan Wu. Catanet: Efficient content-aware token aggregation for lightweight image

super-resolution. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17902–17912. IEEE, 2025. 2

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

[27] Vignesh V Menon, Prajit T Rajendran, Amritha Premkumar, Benjamin Bross, and Detlev Marpe. Video super-resolution for optimized bitrate and green online streaming. *arXiv preprint arXiv:2402.03513*, 2024. 1

[28] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1996–2005. IEEE, 2019. 5

[29] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6626–6634, 2018. 1, 2

[30] Wei Shang, Dongwei Ren, Wanying Zhang, Yuming Fang, Wangmeng Zuo, and Kede Ma. Arbitrary-scale video super-resolution with structural and textural priors. In *European Conference on Computer Vision*, pages 73–90. Springer, 2024. 2

[31] Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. *Advances in Neural Information Processing Systems*, 35:36081–36093, 2022. 2

[32] Shijun Shi, Jing Xu, Lijing Lu, Zhihang Li, and Kai Hu. Self-supervised controlnet with spatio-temporal mamba for real-world video super-resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7385–7395, 2025. 2

[33] Alexandros Stergiou and Ronald Poppe. Adapool: Exponential adaptive pooling for information-retaining downsampling. *IEEE Transactions on Image Processing*, 32:251–266, 2022. 5

[34] Guozhi Tang, Hongwei Ge, Yong Luo, Bo Li, and Chunguo Wu. Multi-memory streams: A paradigm for online video super-resolution in complex exposure scenes. *IEEE Transactions on Multimedia*, 2025. 1, 2, 3, 6, 7

[35] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3360–3369, 2020. 2

[36] Kavitha Viswanathan, Amit Sethi, Shashwat Pathak, Piyush Bharambe, and Harsh Choudhary. Low-resource video super-resolution using memory, wavelets, and deformable convolutions. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3444–3453. IEEE, 2025. 2, 5

[37] Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22387, 2023. 2

[38] Longguang Wang, Yulan Guo, Li Liu, Zaiping Lin, Xinpu Deng, and Wei An. Deep video super-resolution using hr optical flow estimation. *IEEE Transactions on Image Processing*, 29:4323–4336, 2020. 2

[39] Shiyao Wang, Hongchao Lu, and Zhidong Deng. Fast object detection in compressed video. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7104–7113, 2019. 3

[40] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1954–1963. IEEE, 2019. 2

[41] Yingwei Wang, Takashi Isobe, Xu Jia, Xin Tao, Huchuan Lu, and Yu-Wing Tai. Compression-aware video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2012–2021, 2023. 3

[42] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003. 5

[43] Xinyi Wu, Santiago López-Tapia, Xijun Wang, Rafael Molina, and Aggelos K Katsaggelos. Real-time lightweight video super-resolution with rred-based perceptual constraint. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10):10310–10325, 2024. 2, 5

[44] Bin Xia, Jingwen He, Yulun Zhang, Yitong Wang, Yapeng Tian, Wenming Yang, and Luc Van Gool. Structured sparsity learning for efficient video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22638–22647, 2023. 1, 6, 7

[45] Jun Xiao, Xinyang Jiang, Ningxin Zheng, Huan Yang, Yifan Yang, Yuqing Yang, Dongsheng Li, and Kin-Man Lam. Online video super-resolution with convolutional kernel bypass grafts. *IEEE Transactions on Multimedia*, 25:8972–8987, 2023. 1, 2, 5

[46] Rui Xie, Yinhong Liu, Penghao Zhou, Chen Zhao, Jun Zhou, Kai Zhang, Zhenyu Zhang, Jian Yang, Zhenheng Yang, and Ying Tai. Star: Spatial-temporal augmentation with text-to-video models for real-world video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17108–17118, 2025. 2

[47] Yiran Xu, Taesung Park, Richard Zhang, Yang Zhou, Eli Shechtman, Feng Liu, Jia-Bin Huang, and Difan Liu. Videogigagan: Towards detail-rich video super-resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2139–2149, 2025. 2

[48] 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

[49] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3106–3115, 2019. 2

[50] Guanghao Yin, Zefan Qu, Xinyang Jiang, Shan Jiang, Zhenhua Han, Ningxin Zheng, Huan Yang, Xiaohong Liu, Yuqing Yang, Dongsheng Li, et al. Online streaming video super-resolution with convolutional look-up table. *IEEE Transactions on Image Processing*, 33:2305–2317, 2024. 1

[51] 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 *European Conference on Computer Vision*, pages 220–235. Springer, 2022. 3

[52] Yuehan Zhang and Angela Yao. Realviformer: Investigating attention for real-world video super-resolution. In *European Conference on Computer Vision*, pages 412–428. Springer, 2024. 2

[53] Zhengqiang Zhang, Ruihuang Li, Shi Guo, Yang Cao, and Lei Zhang. Tmp: Temporal motion propagation for online video super-resolution. *IEEE Transactions on Image Processing*, 2024. 1, 2, 4, 5, 6, 7

[54] Mingjun Zheng, Long Sun, Jiangxin Dong, and Jinshan Pan. Smfanet: A lightweight self-modulation feature aggregation network for efficient image super-resolution. In *European conference on computer vision*, pages 359–375. Springer, 2024. 2

[55] Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2535–2545, 2024. 2

[56] Xingyu Zhou, Leheng Zhang, Xiaorui Zhao, Keze Wang, Leida Li, and Shuhang Gu. Video super-resolution transformer with masked inter&intra-frame attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25399–25408, 2024. 2

[57] Qiang Zhu, Xiandong Meng, Yuxian Jiang, Fan Zhang, David Bull, Shuyuan Zhu, and Bing Zeng. Trajectory-aware shifted state space models for online video super-resolution. *arXiv preprint arXiv:2508.10453*, 2025. 1, 2, 5