VideoGigaGAN: Towards Detail-rich Video Super-Resolution

Yiran Xu[1,2] Taesung Park[1] Richard Zhang[1] Yang Zhou[1] Eli Shechtman[1]
Feng Liu[1] Jia-Bin Huang[2] Difan Liu[1]
[1]Adobe Research [2]University of Maryland, College Park
https://videogigagan.github.io/

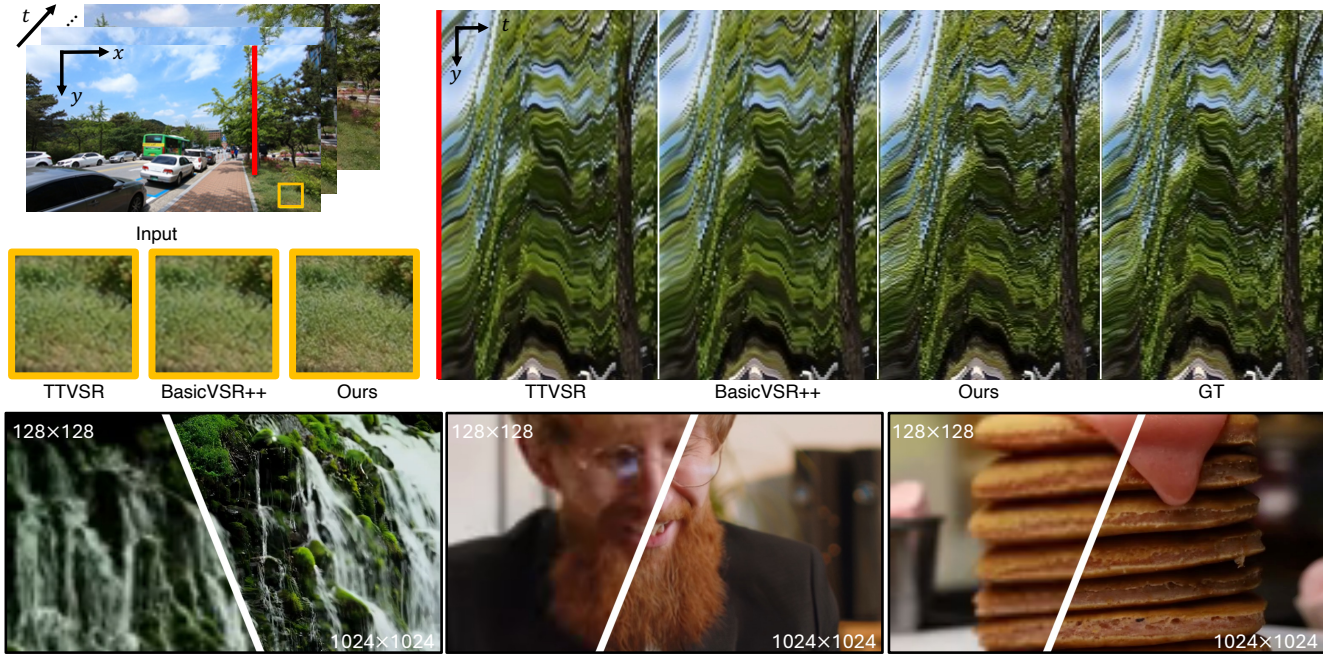

Figure 1. We present **VideoGigaGAN**, a generative video super-resolution model that upscales videos with high-frequency details while preserving temporal consistency. **Top:** We compare our approach with TTVSR [39] and BasicVSR++ [8]. Our method produces better temporal consistency and finer details than previous methods. **Bottom:** Our model produces high-quality videos with 8× super-resolution.

Abstract

Video super-resolution (VSR) models achieve temporal consistency but often produce blurrier results than their image-based counterparts due to limited generative capacity. This prompts the question: can we adapt a generative image upsampler for VSR while preserving temporal consistency? We introduce VideoGigaGAN, a new generative VSR model that combines high-frequency detail with temporal stability, building on the large-scale GigaGAN image upsampler. Simple adaptations of GigaGAN for VSR led to flickering issues, so we propose techniques to enhance temporal consistency. We validate the effectiveness of VideoGigaGAN by comparing it with state-of-the-art VSR models on public datasets and showcasing video results with 8× upsampling.

1. Introduction

Challenges. Video super-resolution (VSR) aims to reconstruct high-resolution videos from low-resolution inputs, a task challenged by the need for *temporal consistency* and *high-frequency detail generation*. VSR shows wide applications in generated videos [2, 20], face videos [13], satellite videos [66], and Animes [65]. Existing methods [7–9, 25] focus on consistency but often result in blurry outputs that lack high-frequency appearance details or realistic textures (see Fig. 2). Effective VSR requires generating plausible new content not present in the low-resolution inputs, a capability that these models struggle with. Recent diffusion-based methods [20, 47, 73, 80] enjoy higher per-frame quality but suffer from temporal flickering and slow inference.

ISR v.s. VSR. Generative models (*e.g.*, diffusion mod-

Figure 2. **Limitations of previous methods.** Previous VSR approaches such as VRT [35] suffer from lack of details, as seen from the building example. Generative models, image GigaGAN [27] and StableVSR [47] produce sharper results with richer details, but it generates videos with temporal flickering or artifacts like aliasing (see red arrows). Our VideoGigaGAN can produce video results with both high-frequency details and temporal consistency, while artifacts like aliasing are significantly mitigated. Please refer to our supplementary material for a visual comparison.

els [45, 59], VAEs [11], and GANs [18, 61, 62]) have advanced Image Super-Resolution (ISR) by modeling high-resolution image distributions, producing highly detailed textures. GigaGAN [27] further increases the generative capability of image super-resolution models by training a large-scale GAN model on billions of images. However, applying a generative model such as GigaGAN independently to video frames results in severe temporal artifacts (see Fig. 2). This raises the question: can GigaGAN's capabilities be harnessed for temporally consistent VSR?

Consistency-quality dilemma. We first experiment with an adapted GigaGAN baseline using temporal convolution and attention layers, which helps but fails to fully address the flickering of high-frequency details, brought by the strong hallucinations. Previous VSR approaches use regression-based networks to trade high-frequency details for better temporal consistency. As blurrier upsampled videos inherently exhibit better temporal consistency, the capability of GANs to hallucinate high-frequency details contradicts the goal of VSR in producing temporally consistent frames. We refer to this as the *consistency-quality dilemma* in VSR.

Our work. In this work, we identify several key issues of applying GigaGAN for VSR and propose techniques to achieve detailed and temporally consistent video super-resolution. Naively inflating GigaGAN with temporal modules [21] is not sufficient to produce temporally consistent results with high-quality frames. To address this issue, we employ a *recurrent flow-guided feature propagation module* to encourage information aggregation across different frames. We also apply *anti-aliasing blocks* in GigaGAN to address the temporal flickering caused by the aliased downsampling operations. Furthermore, we introduce an effective method for injecting high-frequency features into the GigaGAN decoder, called *high-frequency (HF) shuttle*. The proposed high-frequency shuttle can effectively add fine-grained details to the upsampled videos while maintaining the temporal consistency.

Our contributions are as follows:

- We present VideoGigaGAN, the first large-scale GAN-based model for VSR, addressing the overlooked consistency-quality trade-off.
- We introduce anti-aliasing blocks and HF shuttle, which significantly improve the temporal consistency.
- We show that VideoGigaGAN can upsample videos with much more fine-grained details than state-of-the-art methods evaluated on multiple datasets.
- VideoGigaGAN is capable for challenging $8\times$ VSR.

2. Related Work

Video Super-Resolution. Considerable research on video super-resolution (VSR) has explored sliding-window methods [6, 34, 55, 57, 58, 69] and recurrent networks [23–25, 33, 35, 36, 50–52]. BasicVSR [7] established a unified VSR pipeline using optical flow for alignment and bidirectional recurrent networks, which was later enhanced in BasicVSR++ [8] with flow-guided deformable alignment for better performance. Methods like RealBasicVSR [9], MGLD-VSR [72], and FastRealVSR [67] employ diverse degradations to improve generalization on real-world videos, but their reliance on regression objectives results in overly smooth, less realistic outputs. Diffusion-based VSR models [20, 47, 73, 80] offer finer detail but suffer from temporal flickering, fidelity loss and slow inference. In contrast, we propose a GAN-based VSR model that achieves high-frequency detail and temporal consistency in upsampled videos, with faster performance than diffusion-based approaches.

GAN-based Image Super-Resolution. SRGAN [31] pioneered using GANs for image super-resolution, modeling the high-resolution image manifold. ESRGAN [62] improved upon SRGAN with better architecture and loss functions, and Real-ESRGAN [61] extended it to handle real-world low-resolution images. However, these models are

limited in capacity and struggle with large upsampling factors. GigaGAN [27] addresses this by incorporating filter banks and attention layers into StyleGAN2 [29], scaling up to billions of images. Capable of handling $8\times$ upsampling, GigaGAN generates highly detailed, realistic textures, even creating new content beyond the low-resolution input.

Generative video models for VSR. Many video generation works are based on the VAEs [1, 32, 70], GANs [15, 54, 76], and autoregressive models [63]. LongVideoGAN [5] introduces a sliding-window approach for video super-resolution, but it is restricted to datasets with limited diversity. Recently, diffusion models have shown diverse and high-quality results in video generation tasks [3, 4, 16, 17, 22]. Imagen Video [21] proposes pixel diffusion models for video super-resolution. Lumiere [2] apply chunked SR with a MultiDiffusion for VSR. Unlike diffusion-based video super-resolution models that require iterative denoising processes, our VideoGigaGAN can generate outputs in a *single feedforward pass* with faster inference speed.

3. Method

Our VSR model $\mathcal{G}$ upsamples a low-resolution (LR) video $\mathbf{v} \in \mathbb{R}^{T\times h\times w\times 3}$ to a high-resolution (HR) video $\mathbf{V} = \mathcal{G}(\mathbf{v})$, where $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$, with an upsampling scale factor α such that $H = \alpha h$, $W = \alpha w$. We aim to generate HR videos with both high-frequency appearance details and temporal consistency.

We present the overview of our VSR model, **VideoGigaGAN**, in Fig. 3. We start with the large-scale GAN-based image upsampler – GigaGAN [27] (Section 3.1). We first inflate the 2D image GigaGAN upsampler to a 3D video GigaGAN upsampler by adding temporal convolutional and attention layers (Section 3.2). However, as shown in our experiments, the inflated GigaGAN still produces results with severe temporal flickering and artifacts, likely due to the limited spatial window size of the temporal attention. To this end, we introduce flow-guided feature propagation (Section 3.3) to the inflated GigaGAN to better align the features of different frames based on flow information. We also pay special attention to anti-aliasing (Section 3.4) to further mitigate the temporal flickering caused by the downsampling blocks in the GigaGAN encoder, while maintaining the high-frequency details by directly shuttling the HF features to the decoder blocks (Section 3.5). Our experimental results validate the importance of these model design choices.

3.1. Preliminaries: Image GigaGAN upsampler

Our VideoGigaGAN builds upon the GigaGAN image upsampler [27]. GigaGAN scales up the StyleGAN2 [29] architecture using several key components, including adaptive kernel selection for convolutions and self-attention layers. The GigaGAN image upsampler has an asymmetric U-Net architecture consisting of 3 downsampling blocks $\{E_i\}$ and $3 + k$ upsampling decoder blocks $\{D_i\}$.

$$\begin{aligned}\mathbf{X} &= \mathcal{G}(\mathbf{x}, \mathbf{z}) = D(E(\mathbf{x}, \mathbf{z}), \mathbf{z}) \\ &= \underbrace{D_{k+2} \circ \cdots D_3}_{\uparrow\times 2^k} \circ \underbrace{D_2 \circ D_1 \circ D_0}_{\uparrow\times 8} \circ \underbrace{E_2 \circ E_1 \circ E_0(\mathbf{x}, \mathbf{z})}_{\downarrow\times 8} .\end{aligned} \tag{1}$$

This GigaGAN upsampler is able to upsample an input image by 2^k. Both encoder E and decoder D blocks utilize random spatial noise $\mathbf{z}$ as a source of stochasticity. The decoder D contains spatial self-attention layers. The encoder and decoder block at same resolution are connected by skip connections.

3.2. Inflation with temporal modules

To adapt a pretrained 2D image model for video tasks, a common approach is to inflate 2D spatial modules into 3D temporal ones [4, 16, 21, 64, 71, 80]. To reduce the memory cost, instead of directly using 3D convolutional layers in each block, our temporal module uses a 1D temporal convolution layer that only operates on the temporal dimension of kernel size 3, followed by a temporal self-attention layer with no spatial receptive field. Both 1D temporal convolution and temporal self-attention are inserted after the spatial self-attention with residual connection [21]. In summary, at each block D_i, we first process the features of individual video frames using the spatial self-attention layer and then jointly processed by our temporal module. Through our experiment, we find adding temporal modules to the decoder D of the generator $\mathcal{G}$ is sufficient to improve video consistency. We also inflate the discriminator $\mathcal{D}$ with comparable temporal modules.

We follow [75] to initialize both temporal convolutions and temporal self-attention layers with zero weights, such that $\mathcal{G}$ and $\mathcal{D}$ still perform the same as an image upsampler at the beginning of the training, leading to a smoother transition to a video upsampler.

3.3. Flow-guided feature propagation

The temporal modules alone are insufficient to ensure temporal consistency, mainly due to the high memory cost of the 3D layers. For input videos with long sequences of frames, one could partition the video into small, non-overlapping chunks and apply temporal attention. However, this leads to temporal flickering between different chunks. Even within each chunk, the spatial window size of the temporal attention is limited, meaning a large motion (i.e., exceeding the receptive field) cannot be modeled by the attention module (see Fig. 4).

To address these issues, we augment the input image with features aligned by optical flow. Specifically, we in-

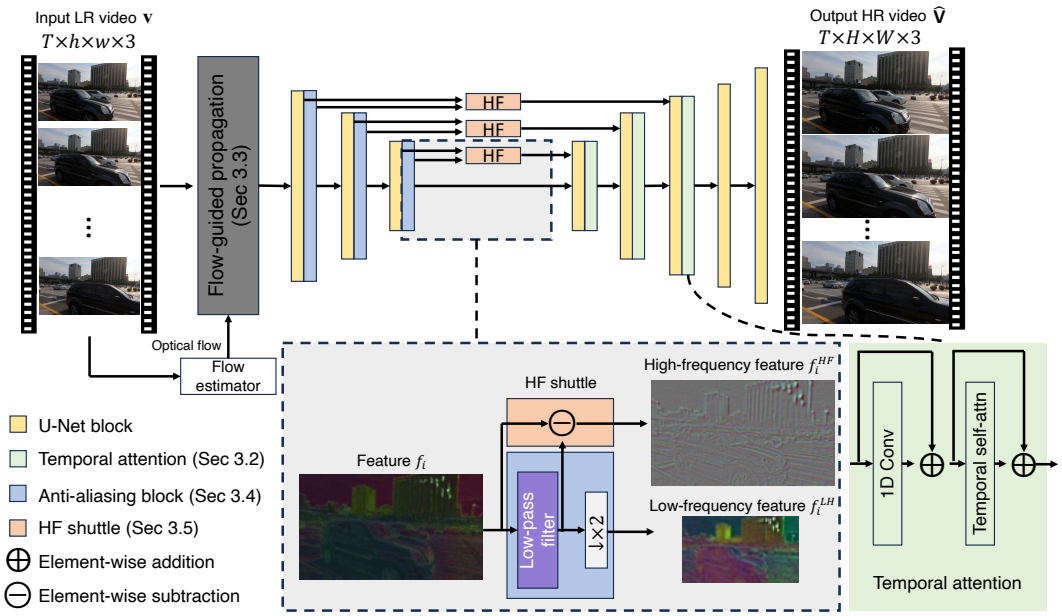

Figure 3. **Overview of VideoGigaGAN** for 4× upsampling. Our VideoGigaGAN model is built upon the asymmetric U-Net architecture of the image GigaGAN upsampler [27]. To enforce temporal consistency, we first inflate the image upsampler into a video upsampler by adding temporal attention layers into the decoder blocks. We also enhance consistency by incorporating the features from the flow-guided propagation module. To suppress aliasing artifacts, we use Anti-aliasing block in the downsampling layers of the encoder. Lastly, we directly shuttle the high frequency features via skip connection to the decoder layers to compensate for the loss of details in the BlurPool process.

troduce a recurrent flow-guided feature propagation module (see Fig. 3) prior to the inflated GigaGAN, inspired by BasicVSR++ [8]. Instead of directly using the LR video as input to the inflated GigaGAN, we use the temporal-aware features produced by the flow-guided propagation module. It comprises a bi-directional recurrent neural network (RNN) [7, 8] and an image backward warping layer. We initially employ the optical flow estimator to predict bi-directional optical flow maps from the input LR video. Subsequently, these maps and the original frame pixels are fed into the RNN to learn temporal-aware features. Finally, these features are explicitly warped using the backward warping layer, guided by the pre-computed optical flows, before being fed into the later inflated GigaGAN blocks. The flow-guided propagation module can effectively handle large motion and produce better temporal consistency in output videos, as demonstrated in Fig 4.

During training, we jointly train the flow-guided feature propagation module and the inflated GigaGAN model. At inference time, given an input LR video with an arbitrary number of frames, we first generate frame features using the flow-guided propagation module. We then partition the frame features into non-overlapping chunks and independently apply the inflated GigaGAN on each chunk. Since the features inside each chunk are *aware* of the other chunks, thanks to the flow-guided propagation module, the temporal consistency between consecutive chunks is preserved well.

3.4. Anti-aliasing blocks

With both temporal and feature propagation modules enabled, our VSR model can process longer videos and produce results with better temporal consistency. However, the high-resolution frames remain flickering in areas with high-frequency details (for example, the windows in the building in Fig. 2). We identify that the downsampling operations in the GigaGAN encoder contribute to the flickering of those regions. The high-frequency components in the input can easily alias into lower frequencies due to the downsampling rate not meeting the classical sampling criterion [43]. The aliasing of pixels manifests as temporal flickering in video super-resolution. Previous VSR approaches often use regression-based objectives, which tend to remove high-frequency details. Consequently, these methods produce output videos free of aliasing. However, in our GAN-based VSR framework, the GAN training objectives favor the hallucination of high-frequency details, making aliasing a more severe problem.

In the GigaGAN upsampler, the downsampling operation in the encoder is achieved by strided convolutions with a stride of 2. To address the aliasing issue in our output video, we apply BlurPool layers to replace all the

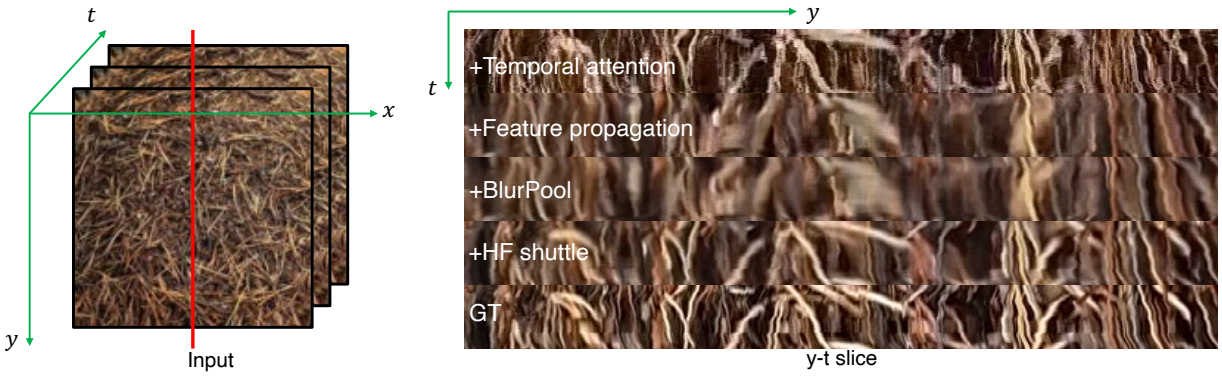

Figure 4. **Ablation study.** Starting from the inflated GigaGAN (+*Temporal attention* in the figure), we progressively add components to demonstrate its effectiveness. With **temporal attention**, the local temporal consistency is improved compared to using image GigaGAN to upsample each frame independently. The global temporal consistency improves with **feature propagation**, but aliasing still exists in the areas with high-frequency details (please refer to the videos in the supplementary material). Also, the video results become more blurry. By using the anti-aliasing blocks – **BlurPool**, the aliasing issue is much better, but the video results become even more blurry. Finally, with **HF shuttle**, we can bring the per-frame quality and high-frequency details back while preserving good temporal consistency.

strided convolution layers in the upsampler encoder inspired by [77]. More specifically, during downsampling, instead of simply using a strided convolution, we use convolution with a stride of 1, followed by a low-pass filter and a subsampling operation. We show the anti-aliasing blocks in Fig. 3. Our experiments show that the anti-aliasing downsampling blocks perform significantly better than naive strided convolutions in preserving temporal consistency for high-frequency details. We also experimented with StyleGAN3 blocks for anti-aliasing upsampling [28], but we observed a notable drop in frame quality.

3.5. High-frequency shuttle

With the newly introduced components, the temporal flicker in our results is significantly suppressed. However, as shown in Fig. 4, adding the flow-guided propagation module (Section 3.3) leads to a blurrier output. Anti-aliasing blocks (Section 3.4) make the results even blurrier. We still need the high-frequency information in the GigaGAN features to compensate for the loss of high-frequency details. However, as discussed in Section 3.4, the traditional flow of high-frequency information in GigaGAN leads to aliased output.

We present a simple yet effective approach to address the conflict of high-frequency details and temporal consistency, called *high-frequency shuttle* (HF shuttle). To guide where the high-frequency details should be inserted, the HF shuttle leverages the skip connections in the U-Net and uses a pyramid-like representation for the feature maps in the encoder. More specifically, at the feature resolution level i, we decompose the feature map f_i into low-frequency (LF) feature and high-frequency (HF) components. The LF feature map f_i^{LF} is obtained via the low-pass filter mentioned in Section 3.4, while the HF feature map is computed from the residual as $f_i^{HF} = f_i - f_i^{LF}$. The HF feature map f_i^{HF} containing high-frequency details are injected through the skip connection to the decoder (Fig. 3). Our experiments show that the high-frequency shuttle can effectively add fine-grained details to the upsampled videos while mitigating issues such as aliasing or temporal flickering.

3.6. Loss functions

We use stardard, non-saturating GAN loss [19], R1 regularization [40], LPIPS [78] and Charbonnier loss [10] during the training.

$$\begin{aligned}\mathcal{L}(\mathbf{X}_t, \mathbf{x}_t) = \mu_{GAN}\mathcal{L}_{GAN}(\mathcal{G}(\mathbf{x}_t), \mathcal{D}(\mathcal{G}(\mathbf{x}_t))) + \mu_{R1}\mathcal{L}_{R1}(\mathcal{D}(\mathbf{X}_t)) \\ + \mu_{LPIPS}\mathcal{L}_{LPIPS}(\mathbf{X}_t, \mathbf{x}_t) + \mu_{Char}\mathcal{L}_{Char}(\mathbf{X}_t, \mathbf{x}_t)\,,\end{aligned} \tag{2}$$

where Charbonnier loss is a smoothed version of pixelwise ℓ_1 loss, $\mu_{GAN}, \mu_{R1}, \mu_{LPIPS}, \mu_{Char}$ are the scales of different loss functions. $\mathbf{x}_t$ is one of the LR input frames, $\mathbf{X}_t$ is the corresponding ground-truth HR frame. We average the loss over all the frames in a video clip during the training.

4. Experimental Results

4.1. Setup

Datasets. We strictly follow two widely used training sets from previous VSR works [7, 8, 39]: **REDS** [42] and **Vimeo-90K** [69]. The REDS dataset contains 300 video sequences. Each sequence consists of 100 frames with a resolution of 1280×720. We use REDS4 as our test set and REDSval4 as our validation set; the rest of the sequences are used for training. The Vimeo-90K contains $64,612$ sequences for training and $7,824$ for testing (known as Vimeo-90K-T). Each sequence contains seven frames with a resolution of 448×256. Following previous works [7, 8], we compute the metrics only on the center frame of each sequence. In addition to the official test set Vimeo-90K-T, we

also evaluate the model on Vid4 [38] and UDM10 [74], with different degradation algorithms (Bicubic Downsampling – BI and Blur Downsampling – BD). We follow MMagic [41] to perform degradation algorithms. All data are $4\times$ downsampled to generate LR frames following standard evaluation protocols [7, 8].

Evaluation metrics. We are interested in two aspects of our evaluation: *per-frame quality* and *temporal consistency*. For per-frame quality, we use **PSNR, SSIM,** and **LPIPS** [78]. For temporal consistency, the warping error E_{warp} [30] is commonly used.

$$E_{\text{warp}}(\hat{\mathbf{X}}_t, \hat{\mathbf{X}}_{t+1}) = \frac{1}{\sum M_t^i} \sum M_t^i ||\hat{\mathbf{X}}_t^i, W(\hat{\mathbf{X}}_{t+1}^i, \mathcal{F}_{t\to t+1})||_2^2 , \tag{3}$$

where $(\hat{\mathbf{X}}_t, \hat{\mathbf{X}}_{t+1})$ are **generated** frames at time t and $t+1$, i is the index of the i-th pixel, and $W(\cdot)$ is the warping function, $\mathcal{F}_{t\to t+1}$ is the forward flow estimated from the generated frames $(\hat{\mathbf{X}}_t, \hat{\mathbf{X}}_{t+1})$ using RAFT [56], and $M_t \in \{0, 1\}$ is a non-occlusion mask indicating non-occluded pixels [48]. However, as reported in Fig. 6, previous baselines or even simple bicubic upsampling achieve lower E_{warp} than ground truth high-resolution video since E_{warp} favors over-smoothed results. Consider an extreme algorithm where all the generated frames are entirely black. E_{warp} computes the warping errors by warping the generated frames. The warping error for this algorithm is **0** since the generated frames are over-smoothed (in this extreme case, all black). Therefore, instead of warping the generated frames, we propose to warp the ground-truth frames using the flow computed on the generated frames. We refer to this new warping error as **referenced warping error (RWE)** $E_{\text{warp}}^{\text{ref}}$. The referenced warping error between two frames is

$$E_{\text{warp}}^{\text{ref}}(\mathbf{X}_t, \mathbf{X}_{t+1}) = \frac{1}{\sum M_t^i} \sum M_t^i ||\mathbf{X}_t^i, W(\mathbf{X}_{t+1}^i, \mathcal{F}_{t\to t+1}))||_2^2 , \tag{4}$$

where $(\mathbf{X}_t, \mathbf{X}_{t+1})$ are ground-truth frames at time t and $t+1$, $\mathcal{F}_{t\to t+1}$ is the forward flow estimated from the **output** frames $(\hat{\mathbf{X}}_t, \hat{\mathbf{X}}_{t+1})$ using RAFT [56].

Hyperparameters. We use a pretrained $4\times$ GigaGAN image upsampler as our base model. It contains three downsampling blocks in the encoder and five upsampling blocks in the decoder. The spatial self-attention layers are only used in the first block of the decoder for memory efficiency. For the flow network, we use a lightweight SpyNet [44]. For the low-pass filters, we use a kernel of $\frac{1}{16}[1, 4, 6, 4, 1]$ before the downsampling. We set $\mu_{GAN} = 0.05$, $\mu_{R1} = 0.2048$, $\mu_{LPIPS} = 5$, $\mu_{Char} = 10$ in Eqn. 2. During training, we randomly crop a 64×64 patch from each LR input frame at the same location. We use 10 frames of each video and a batch size of 32 for training. The batch is distributed into 32 NVIDIA A100 GPUs. We use a fixed learning rate of 5×10^{-5} for both generator and discriminator. The total number of training iterations is $100,000$.

Table 1. **Ablation study.** We use LPIPS to evaluate per-frame quality and $E_{\text{warp}}^{\text{ref}} \downarrow (\times 10^{-3})$ for temporal consistency. Starting from the image GigaGAN (upsampling each frame independently with the image upsampler), we progressively add components to demonstrate its effectiveness. The best number: **bold**. The second best number: underline.

Model	LPIPS↓	$E_{\text{warp}}^{\text{ref}} \downarrow (\times 10^{-3})$
GigaGAN (base upsampler)	0.2031	2.497
+ Temporal attention	0.2029	2.462
+ Flow-guided propagation	**0.1551**	2.187
+ BlurPool	0.1621	**2.152**
+ High-freq shuttle	<u>0.1582</u>	<u>2.177</u>

4.2. Ablation study

To demonstrate the effect of each proposed component, we progressively add them one by one and evaluate them on the REDS4 dataset [42]. We report the quantitative results in Table 1. We also present a qualitative comparison in Fig. 4. We see that the **flow-guided feature propagation** brings a large LPIPS and $E_{\text{warp}}^{\text{ref}}$ improvement compared to the **temporal attention**. This demonstrates the effectiveness of the feature propagation contributing to the temporal consistency. By further introducing BlurPool as the **anti-aliasing** block, the model has a warping error drop but an LPIPS loss increase (also shown in Fig. 4). Finally, by using **HF shuttle**, we can bring the LPIPS back with a slight loss of temporal consistency. Though it is not reflected on the number clearly, we observed that the sharpness of the frame improves significantly with the HF shuttle (see in the x-t slice plot in Fig. 4 and also supplementary material).

4.3. Comparison with previous models

We conduct extensive experiments and report the quantitative comparison of the per-frame quality in Table 2. We compare temporal consistency in Fig. 6. Additionally, we provide qualitative comparisons in Fig. 5.

Per-frame quality. As shown in Table 2, our LPIPS outperforms all the other models while showing a poorer performance of PSNR and SSIM. We observe that PSNR and SSIM do not align well with human perception and favor blurry results, as also reported in many literatures [12, 27, 46, 47, 49]. For the same reason, StableVSR [47] in Table 2 also shows bad performance on PSNR/SSIM. Thus we consider LPIPS [78] as our core metric to evaluate per-frame quality as it is closer to the human perception. In Fig. 5, it is noticeable that our model produces results with the most fine-grained details. Previous approaches tend to predict blurry results with a critical loss of details.

Temporal consistency. As observed in previous works [30], the widely used warping error metric favors a more blurry video. This is also illustrated in Fig. 6. The simple bicubic upsampling method achieves the best per-

Table 2. **Quantitative comparison in terms of per-frame quality** (LPIPS↓/PSNR↑ /SSIM↑) evaluated on multiple datasets. We separate models into regression-based models and generative models (StableVSR [47] and ours). We exclude LPIPS evaluation on Vimeo-90K-T from EvTexture [26] due to the lack of released preprocessed data. For StableVSR [47], we omit Vimeo-90K-T evaluation due to its *significantly long* runtime (Table 3). **We highlight LPIPS** as PSNR/SSIM often misaligns with human perception and favors blurrier results, as noted in many studies [12, 27, 46, 47, 49]. Our VideoGigaGAN aligns the best with human perception.

	BI degradation (LPIPS↓/PSNR↑/SSIM↑)			BD degradation (LPIPS↓/PSNR↑/SSIM↑)		
	REDS4 [42]	Vimeo-90K-T [69]	Vid4 [38]	UDM10 [74]	Vimeo-90K-T [69]	Vid4 [38]
EDVR [60]	0.2097/31.05/0.8793	-/37.61/0.9489	-/27.35/0.8264	-/39.89/0.9686	-/37.81/0.9523	-/27.85/0.8503
MuCAN [34]	0.2162/30.88/0.8750	0.1523/37.32/0.9465	-	-	-	-
BasicVSR [7]	0.2023/31.42/0.8909	0.1616/37.18/0.9450	0.2812/27.24/0.8251	0.1148/39.96/0.9694	0.1551/37.53/0.9498	0.2555/27.96/0.8553
IconVSR [7]	0.1939/31.67/0.8948	0.1587/37.47/0.9476	0.2739/27.39/0.8279	0.1152/40.03/0.9694	0.1531/37.84/0.9524	0.2462/28.04/0.8570
TTVSR [39]	0.1836/32.12/0.9021	-	-	0.1112/40.41/0.9712	0.1507/37.92/0.9526	0.2381/28.40/0.8643
BasicVSR++ [8]	0.1786/32.39/0.9069	0.1506/37.79/0.9500	0.2627/27.79/0.8400	0.1131/40.72/0.9722	0.1440/38.21/ 0.9550	0.2390/29.04/0.8753
RVRT [37]	0.1727/32.74/0.9113	0.1502/38.15/0.9527	0.2500/27.99/0.8464	0.1100/40.90/0.9729	0.1465/38.59/ 0.9576	0.2219/29.54/0.8811
PSRT-recurrent [53]	0.1676/32.72/0.9106	0.1509/**38.27**/0.9536	0.2448/28.07/0.8485	-	-	-
MIA-VSR [81]	0.1659/**32.79**/0.9115	0.1428/38.22/0.9532	0.2474/28.20/0.8507	-	-	-
IA-RT [68]	0.1629/32.89/0.9138	0.1498/38.14/0.9528	0.2501/28.26/0.8517	0.1129/**41.15/0.9750**	0.1435/38.62/0.9579	0.2201/**29.68/0.8884**
VRT [35]	0.1818/32.19/0.9005	0.1461/38.20/0.9530	0.2478/27.93/0.8425	0.1097/41.05/0.9737	0.1421/**38.72/0.9584**	0.2214/29.42/0.8795
EvTexture [26]	0.1684/**32.79/0.9173**	-/38.23/**0.9544**	0.2188/**29.51/0.8909**	-	-	-
StableVSR [47]	0.1934/27.98/0.7952	-	0.2803/24.48/0.6989	-	-	-
Ours	**0.1582**/30.46/0.8718	**0.1120**/35.97/0.9238	**0.1925**/26.78/0.8029	**0.1060**/36.57/0.9521	**0.1129**/35.30/0.9317	**0.1832**/27.04/0.8365

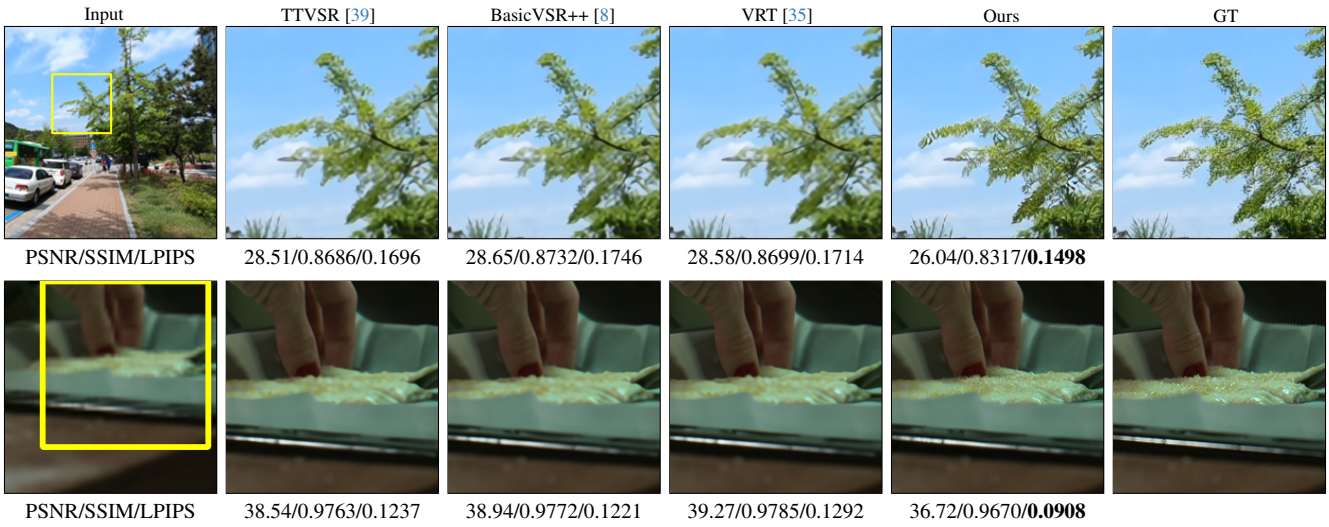

Figure 5. **Qualitative comparison with other baselines on public datasets (REDS4 [42], Vimeo-90K-T [69]).** We show **PSNR/SSIM/LPIPS** below each output frame. PSNR and SSIM do not align well with human perception and favor blurry results [12, 27, 46, 47, 49]. LPIPS is a preferred metric that aligns better with human perception. Compared to previous VSR approaches, our model can produce more fine-grained details. **Zoom in for clear comparison**. We encourage readers to view the videos in our **supplementary material**.

formance for the commonly used warping error, which is much better than the GT warping error. We proposed the referenced warping error (**RWE**) in Eqn. 4 to address the issue of warping error favoring blurry results. In terms of the referenced warping error, our method is slightly worse than previous regression-based methods (0.05×10^{-3} compared to BasicVSR++ [8]) but with much better frame quality. The newly proposed **RWE** is more suitable for evaluating the temporal consistency of videos. However, it is still biased towards more blurry results as seen in Fig. 6 (**RWE** of many methods are better than ground truth). We leave a better metric of VSR temporal consistency for future work.

4.4. Trade-off analysis: Temporal consistency vs. frame quality

To analyze the balance between temporal consistency and per-frame quality, we provide a visualization in Fig. 6. Regression-based models prioritize temporal consistency but at the cost of sharpness, reflected in higher LPIPS losses (see Fig. 5 for qualitative comparisons).

In Fig. 6, we also compare with diffusion-based VSR (UAV [80], DiffIR2VR-Zero [73], and VEnhancer [20]) that are trained on larger datasets. Despite large-scale training, they exhibit severe temporal inconsistency and low fidelity (significantly higher LPIPS) to the ground truth due

Method	LPIPS↓	$E_{\text{warp}} \downarrow (\times 10^{-3})$	$E^{\text{ref}}_{\text{warp}} \downarrow (\times 10^{-3})$
Bicubic	0.3396	**1.161**	2.4232
RealViformer [79]	0.2298	3.128	2.3183
TTVSR [39]	0.1836	1.390	2.1178
BasicVSR++ [8]	0.1786	1.401	2.1206
RVRT [37]	0.1727	1.438	2.1217
MIA-VSR [81]	0.1659	1.439	**2.1172**
VRT [35]	0.1818	1.398	2.1184
EvTexture [26]	0.1684	1.488	2.1320
StableVSR [47]	0.1934	3.957	2.2123
UAV [80]	0.4157	12.881	7.5241
DiffIR2VR-Zero [73]	0.3265	6.665	3.0942
VEnhancer [20]	0.4744	14.270	2.7383
Ours	**0.1582**	2.313	2.1773
Ground truth	-	2.127	2.1272

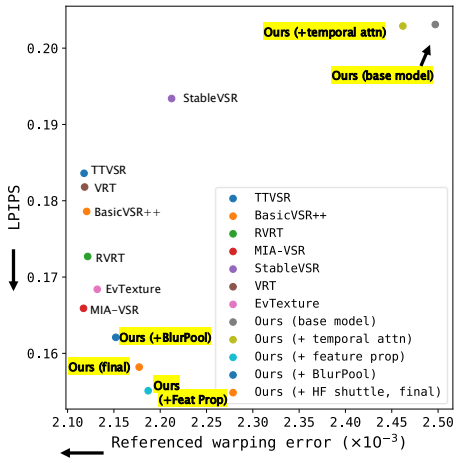

Figure 6. **Trade-off between per-frame quality (LPIPS↓) and temporal consistency (RWE $E^{\text{ref}}_{\text{warp}} \downarrow$).** Our final model achieves a good balance between temporal consistency and per-frame quality. The commonly used E_{warp} for temporal consistency favors more blurry results. The naive BICUBIC upsampling method achieves the lowest E_{warp}. To address this issue, we propose to use the referenced warping error (RWE) $E^{\text{ref}}_{\text{warp}}$ (Eqn. 4) for temporal consistency.

to model hallucination.

Our final model, VideoGigaGAN, achieves a balanced trade-off, significantly enhancing both temporal consistency and per-frame quality over the base GigaGAN model with our proposed improvements.

Table 3. **More comparison.** Per-frame **runtimes** on $320 \times 180 \rightarrow 1280 \times 720$ are evaluated on REDS4 [42]. VideoGigaGAN demonstrates competitive runtimes compared to (a) regression-based models [35, 37, 68] and is substantially faster than (b) diffusion models [20, 47, 73, 80]. (c) **Scaling up** BasicVSR++ does not yield performance improvements. (d) **Adding LPIPS** to the training loss improves performance on LPIPS, but it introduces lower PSNR/SSIM and makes the training unstable.

	Model	#Params (M)	Runtime (ms)	LPIPS↓	PSNR↑
Regress.	IA-RT [68]	13.4	1895	0.1629	32.89
	VRT [35]	35.6	219	0.1818	32.19
	RVRT [37]	10.8	169	0.1727	32.74
Diffusion	UAV [80]	746	7153	0.4157	23.03
	VEnhancer [20]	2496	5168	0.4744	19.92
	DiffIR2VR-Zero [73]	166	12212	0.3265	24.95
	StableVSR [47]	712	9242	0.1934	27.98
Scaling	BasicVSR++ (small) [8]	7.3	77	0.1786	32.39
	BasicVSR++ (medium)	166	85	0.1834	32.09
	BasicVSR++ (large)	368	92	0.1941	31.74
+LPIPS	BasicVSR++ (small) + LPIPS	7.3	77	0.1646	31.42
	RVRT + LPIPS	10.8	169	diverged	diverged
	VideoGigaGAN (ours)	369	295	0.1582	30.46

4.5. Additional results

Model sizes and runtimes. Table 3 compares model sizes and runtimes across VSR methods. Despite its larger size due to generative capacity, our model maintains competitive speed. Unlike slower diffusion-based models [20, 47, 73, 80] requiring iterative denoising, VideoGigaGAN achieves fast results in a *single feedforward pass*.

Scaling-up experiments. For fair comparison, we scale up BasicVSR++[8] with additional layers and channels, evaluating on REDS4[42]. Consistent with findings in [27], scaling up BasicVSR++ alone does not enhance performance. Despite a similar model size, training BasicVSR++ (large) becomes unstable past 40K iterations. We report its performance at 35K in Table 3, which is worse than our model.

Adding LPIPS to the training loss. We also add LPIPS to the training loss of BasicVSR++ and RVRT [37]. We report the results in Table 3. The performance of BasicVSR++ (small) on LPIPS metric improves, but with a drop in PSNR and SSIM. Training RVRT with LPIPS is unstable and finally diverges. Moreover, training BasicVSR++ with LPIPS produces severe checkerboard artifacts in all results, similar to previous works. We show some qualitative results in our supplementary material.

More perceptual metrics. We mainly use LPIPS as the metric for per-frame quality but acknowledge its limitations in capturing higher-level structures [14]. To address this, we also evaluate FID and DISTS in the supplementary material.

5. Conclusions

We present VideoGigaGAN, a novel generative VSR model that enhances high-frequency details and temporal consistency in upscaled videos. Unlike previous regression-based VSR methods that produce blurrier outputs, VideoGigaGAN leverages the powerful GigaGAN image upsampler. We address key challenges like temporal flickering and aliasing by introducing new components that improve both consistency and per-frame quality. Our results show that VideoGigaGAN effectively balances the consistency-quality trade-off in VSR, outperforming previous methods.

References

[1] Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R.H., Levine, S.: Stochastic variational video prediction. In: ICLR (2018) 3

[2] Bar-Tal, O., Chefer, H., Tov, O., Herrmann, C., Paiss, R., Zada, S., Ephrat, A., Hur, J., Liu, G., Raj, A., et al.: Lumiere: A space-time diffusion model for video generation. arXiv preprint arXiv:2401.12945 (2024) 1, 3

[3] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) 3

[4] Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR (2023) 3

[5] Brooks, T., Hellsten, J., Aittala, M., Wang, T.C., Aila, T., Lehtinen, J., Liu, M.Y., Efros, A., Karras, T.: Generating long videos of dynamic scenes. In: NeurIPS (2022) 3

[6] Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z., Shi, W.: Real-time video super-resolution with spatio-temporal networks and motion compensation. In: CVPR (2017) 2

[7] Chan, K.C., Wang, X., Yu, K., Dong, C., Loy, C.C.: Basicvsr: The search for essential components in video super-resolution and beyond. In: CVPR (2021) 1, 2, 4, 5, 6, 7

[8] Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In: CVPR (2022) 1, 2, 4, 5, 6, 7, 8

[9] Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Investigating tradeoffs in real-world video super-resolution. In: CVPR (2022) 1, 2

[10] Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: ICIP (1994) 5

[11] Chen, C., Shi, X., Qin, Y., Li, X., Han, X., Yang, T., Guo, S.: Real-world blind super-resolution via feature matching with implicit high-resolution priors. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 1329–1338 (2022) 2

[12] Chu, M., Xie, Y., Mayer, J., Leal-Taixé, L., Thuerey, N.: Learning temporal coherence via self-supervision for gan-based video generation. ACM Transactions on Graphics (TOG) **39**(4), 75–1 (2020) 6, 7

[13] Feng, R., Li, C., Loy, C.C.: Kalman-inspired feature propagation for video face super-resolution. In: ECCV (2024) 1

[14] Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T., Isola, P.: Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In: NeurIPS (2023) 8

[15] Ge, S., Hayes, T., Yang, H., Yin, X., Pang, G., Jacobs, D., Huang, J.B., Parikh, D.: Long video generation with time-agnostic vqgan and time-sensitive transformer. In: ECCV (2022) 3

[16] Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.B., Liu, M.Y., Balaji, Y.: Preserve your own correlation: A noise prior for video diffusion models. In: ICCV (2023) 3

[17] Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S.S., Shah, A., Yin, X., Parikh, D., Misra, I.: Emu video: Factorizing text-to-video generation by explicit image conditioning. arXiv preprint arXiv:2311.10709 (2023) 3

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 2

[19] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NeurIPS (2017) 5

[20] He, J., Xue, T., Liu, D., Lin, X., Gao, P., Lin, D., Qiao, Y., Ouyang, W., Liu, Z.: Venhancer: Generative space-time enhancement for video generation. arXiv preprint arXiv:2407.07667 (2024) 1, 2, 7, 8

[21] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022) 2, 3

[22] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: NeurIPS (2022) 3

[23] Huang, Y., Wang, W., Wang, L.: Bidirectional recurrent convolutional networks for multi-frame super-resolution. NeurIPS (2015) 2

[24] Huang, Y., Wang, W., Wang, L.: Video super-resolution via bidirectional recurrent convolutional networks. TPAMI **40**(4), 1015–1028 (2017)

[25] Isobe, T., Jia, X., Gu, S., Li, S., Wang, S., Tian, Q.: Video super-resolution with recurrent structure-detail network. In: ECCV (2020) 1, 2

[26] Kai, D., Lu, J., Zhang, Y., Sun, X.: EvTexture: Event-driven Texture Enhancement for Video Super-Resolution. In: Proceedings of the 41st International Conference on Machine Learning. vol. 235, pp. 22817–22839. PMLR (2024) 7, 8

[27] Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: CVPR (2023) 2, 3, 4, 6, 7, 8

[28] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: NeurIPS (2021) 5

[29] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020) 3

[30] Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV (2018) 6

[31] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017) 2

[32] Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic adversarial video prediction. arXiv preprint arXiv:1804.01523 (2018) 3

[33] Li, F., Zhang, L., Liu, Z., Lei, J., Li, Z.: Multi-frequency representation enhancement with privilege information for video super-resolution. In: CVPR (2023) 2

[34] Li, W., Tao, X., Guo, T., Qi, L., Lu, J., Jia, J.: Mucan: Multi-correspondence aggregation network for video super-resolution. In: ECCV (2020) 2, 7

[35] Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., Van Gool, L.: Vrt: A video restoration transformer. IEEE Transactions on Image Processing (2024) 2, 7, 8

[36] Liang, J., Fan, Y., Xiang, X., Ranjan, R., Ilg, E., Green, S., Cao, J., Zhang, K., Timofte, R., Gool, L.V.: Recurrent video restoration transformer with guided deformable attention. In: NeurIPS (2022) 2

[37] Liang, J., Fan, Y., Xiang, X., Ranjan, R., Ilg, E., Green, S., Cao, J., Zhang, K., Timofte, R., Gool, L.V.: Recurrent video restoration transformer with guided deformable attention. In: NeurIPS (2022) 7, 8

[38] Liu, C., Sun, D.: On bayesian adaptive video super resolution. TPAMI **36**(2), 346–360 (2013) 6, 7

[39] Liu, C., Yang, H., Fu, J., Qian, X.: Learning trajectory-aware transformer for video super-resolution. In: CVPR (2022) 1, 5, 7, 8

[40] Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: ICML (2018) 5

[41] MMagic Contributors: MMagic: OpenMMLab multimodal advanced, generative, and intelligent creation toolbox. https://github.com/open-mmlab/mmagic (2023) 6

[42] Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: CVPRW (2019) 5, 6, 7, 8

[43] Nyquist, H.: Certain topics in telegraph transmission theory. Transactions of the American Institute of Electrical Engineers **47**(2), 617–644 (1928) 4

[44] Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR (2017) 6

[45] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 2

[46] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 6, 7

[47] Rota, C., Buzzelli, M., van de Weijer, J.: Enhancing perceptual quality in video super-resolution through temporally-consistent detail synthesis using diffusion models. In: ECCV (2024) 1, 2, 6, 7, 8

[48] Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos. In: GCPR (2016) 6

[49] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. TPAMI **45**(4), 4713–4726 (2022) 6, 7

[50] Sajjadi, M.S., Vemulapalli, R., Brown, M.: Frame-recurrent video super-resolution. In: CVPR (2018) 2

[51] Shang, W., Ren, D., Zhang, W., Fang, Y., Zuo, W., Ma, K.: Arbitrary-scale video super-resolution with structural and textural priors. In: ECCV (2024)

[52] Shi, S., Gu, J., Xie, L., Wang, X., Yang, Y., Dong, C.: Rethinking alignment in video super-resolution transformers. In: NeurIPS (2022) 2

[53] Shi, S., Gu, J., Xie, L., Wang, X., Yang, Y., Dong, C.: Rethinking alignment in video super-resolution transformers. In: NeurIPS (2022) 7

[54] Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In: CVPR (2022) 3

[55] Tao, X., Gao, H., Liao, R., Wang, J., Jia, J.: Detail-revealing deep video super-resolution. In: ICCV (2017) 2

[56] Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020) 6

[57] Tian, Y., Zhang, Y., Fu, Y., Xu, C.: Tdan: Temporally-deformable alignment network for video super-resolution. In: CVPR (2020) 2

[58] Wang, H., Su, D., Liu, C., Jin, L., Sun, X., Peng, X.: Deformable non-local network for video super-resolution. IEEE Access **7**, 177734–177744 (2019) 2

[59] Wang, J., Yue, Z., Zhou, S., Chan, K.C., Loy, C.C.: Exploiting diffusion prior for real-world image super-resolution. International Journal of Computer Vision (2024) 2

[60] Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: CVPRW (2019) 7

[61] Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: ICCVW (2021) 2
[62] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: ECCVW (2018) 2
[63] Weissenborn, D., Täckström, O., Uszkoreit, J.: Scaling autoregressive video models. In: ICLR (2020) 3
[64] Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: ICCV (2023) 3
[65] Wu, Y., Wang, X., Li, G., Shan, Y.: Animesr: Learning real-world super-resolution models for animation videos. In: NeurIPS (2022) 1
[66] Xiao, Y., Su, X., Yuan, Q., Liu, D., Shen, H., Zhang, L.: Satellite video super-resolution via multiscale deformable convolution alignment and temporal grouping projection. IEEE Transactions on Geoscience and Remote Sensing **60** (2021) 1
[67] Xie, L., Wang, X., Shi, S., Gu, J., Dong, C., Shan, Y.: Mitigating artifacts in real-world video super-resolution models. In: AAAI (2023) 2
[68] Xu, K., Yu, Z., Wang, X., Mi, M.B., Yao, A.: Enhancing video super-resolution via implicit resampling-based alignment. In: CVPR (2024) 7, 8
[69] Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV **127**(8), 1106–1125 (2019) 2, 5, 7
[70] Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. arXiv preprint arXiv:2104.10157 (2021) 3
[71] Yang, S., Zhou, Y., Liu, Z., , Loy, C.C.: Rerender a video: Zero-shot text-guided video-to-video translation. In: ACM SIGGRAPH Asia (2023) 3
[72] Yang, X., He, C., Ma, J., Zhang, L.: Motion-guided latent diffusion for temporally consistent real-world video super-resolution. In: ECCV (2024) 2
[73] Yeh, C.H., Lin, C.Y., Wang, Z., Hsiao, C.W., Chen, T.H., Liu, Y.L.: Diffir2vr-zero: Zero-shot video restoration with diffusion-based image restoration models. arXiv preprint arXiv:2407.01519 (2024) 1, 2, 7, 8
[74] Yi, P., Wang, Z., Jiang, K., Jiang, J., Ma, J.: Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In: ICCV (2019) 6, 7
[75] Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023) 3
[76] Zhang, Q., Yang, C., Shen, Y., Xu, Y., Zhou, B.: Towards smooth video composition. In: ICLR (2023) 3
[77] Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019) 5
[78] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 5, 6
[79] Zhang, Y., Yao, A.: Realviformer: Investigating attention for real-world video super-resolution. In: ECCV (2024) 8
[80] Zhou, S., Yang, P., Wang, J., Luo, Y., Loy, C.C.: Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In: CVPR (2024) 1, 2, 3, 7, 8
[81] Zhou, X., Zhang, L., Zhao, X., Wang, K., Li, L., Gu, S.: Video super-resolution transformer with masked inter&intra-frame attention. In: CVPR (2024) 7, 8