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], and Difan Liu[1]

[1] Adobe Research
[2] University of Maryland, College Park
http://videogigagan.github.io

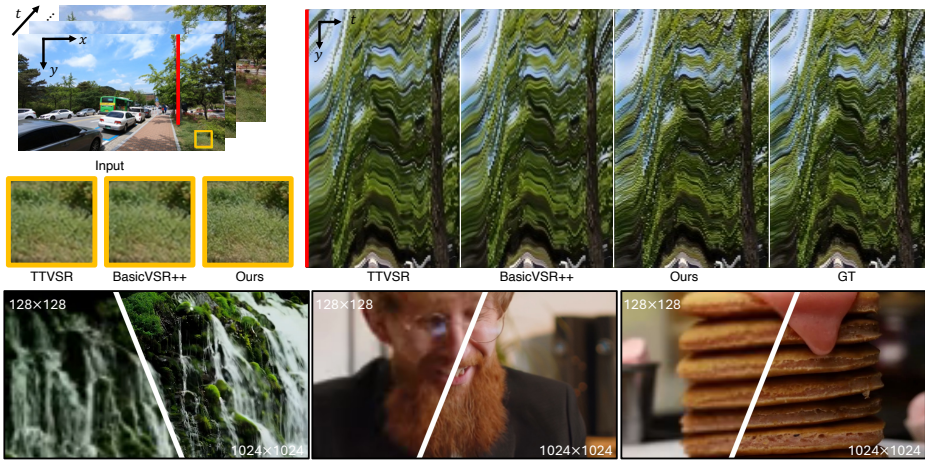

Fig. 1: We present **VideoGigaGAN**, a generative video super-resolution model that can upsample videos with high-frequency details while maintaining temporal consistency. *Top:* we show the comparison of our approach with TTVSR [33] and BasicVSR++ [7]. Our method produces temporally consistent videos with more fine-grained detailed than previous methods. *Bottom:* our model can produce high-quality videos with 8× super-resolution. Please see the video results on our project page.

Abstract. Video super-resolution (VSR) approaches have shown impressive temporal consistency in upsampled videos. However, these approaches tend to generate blurrier results than their image counterparts as they are limited in their generative capability. This raises a fundamental question: can we extend the success of a generative image upsampler to the VSR task while preserving the temporal consistency? We introduce VideoGigaGAN, a new generative VSR model that can produce videos with high-frequency details and temporal consistency. VideoGigaGAN builds upon a large-scale image upsampler – GigaGAN. Simply inflating GigaGAN to a video model by adding temporal modules produces severe temporal flickering. We identify several key issues and propose techniques that significantly improve the temporal consistency

of upsampled videos. Our experiments show that, unlike previous VSR methods, VideoGigaGAN generates temporally consistent videos with more fine-grained appearance details. 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× super-resolution.

1 Introduction

Video super-resolution (VSR) is a classical but challenging task in computer vision and graphics, aiming to recover high-resolution videos from their low-resolution counterparts. VSR has two main challenges. The first challenge is to maintain temporal consistency across output frames. The second challenge is to generate high-frequency details in the upsampled frames. Previous approaches [6–8, 20] focus on addressing the first challenge and have shown impressive temporal consistency in upsampled videos. However, these approaches often produce blurry results and fail to produce high-frequency appearance details or realistic textures (see Fig. 2). An effective VSR model needs to generate plausible new contents not present in the low-resolution input videos. Current VSR models, however, are limited in their generative capability and unable to hallucinate detailed appearances.

Generative Adversarial Networks (GANs) [13] have shown impressive generative capability on the task of image super-resolution [50, 51]. These methods can effectively model the distribution of high-resolution images and generate fine-grained details in upsampled images. GigaGAN [21] further increases the generative capability of image super-resolution models by training a large-scale GAN model on billions of images. GigaGAN can generate highly detailed textures even for 8× upsampling tasks. However, applying GigaGAN or other GAN-based image super-resolution models to each low-resolution video frame independently leads to severe temporal flickering and aliasing artifacts (see Fig. 2). In this work, we ask – is it possible to apply GigaGAN for video super-resolution while achieving temporal consistency in upsampled videos?

We first experiment with a baseline of inflating the GigaGAN by adding temporal convolutional and attention layers. These simple changes alleviate the temporal inconsistency, but the high-frequency details of the upsampled videos are still flickering over time. 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. Previous VSR approaches use regression-based networks to trade high-frequency details for better temporal consistency. 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 [16] 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

Fig. 2: Limitations of previous methods. Previous VSR approaches such as BasicVSR++ [7] suffer from lack of details, as seen from the car example. Image GigaGAN produces sharper results with richer details, but it generates videos with temporal flickering and artifacts like aliasing (see building). Our VideoGigaGAN can produce video results with both high-frequency details and temporal consistency while artifacts like aliasing are significantly mitigated.

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 mitigating aliasing or temporal flickering.

Contributions. We present VideoGigaGAN, the first large-scale GAN-based model for video super-resolution. We recognize the consistency-quality trade-off that has not been well discussed in previous VSR literature. We introduce the feature propagation module, anti-aliasing blocks and HF shuttle which significantly improve the temporal consistency when applying GigaGAN for VSR. We show that VideoGigaGAN can upsample videos with much more fine-grained details than state-of-the-art methods evaluated on multiple datasets. We also show that our model can produce detailed and temporally consistent videos even for challenging 8× upsampling tasks.

2 Related Work

Video Super-Resolution. Significant work has been invested in video super-resolution, using sliding-window approaches [5, 28, 45, 47, 48, 55] and recurrent networks [18–20, 27, 29, 30, 42, 43]. BasicVSR [6] summarizes the common VSR approaches into a unified pipeline. It proposes an effective baseline using optical flow for temporal alignment and bidirectional recurrent networks for feature propagation. BasicVSR++ [7] redesigns BasicVSR by introducing second-order

grid propagation and flow-guided deformable alignment. To improve the generalizability on real-world low-resolution videos, methods like RealBasicVSR [8] and FastRealVSR [54] use diverse degradations as data augmentation during training. While these approaches can produce temporally consistent upsampled videos, they are often trained with simple regression objectives and lack the generative capability, which leads to unrealistic textures and overly blurry results. Unlike previous VSR approaches, we propose a GAN-based VSR model to generate high-frequency details while maintaining temporal consistency in the upsampled videos.

GAN-based Image Super-Resolution. SRGAN [25] is a seminal image super-resolution work that uses a GAN framework to model the manifold of high-resolution images. ESRGAN [51] further enhances the visual quality of upsampled images by improving the architecture and loss of SRGAN. Real-ESRGAN [50] extends ESRGAN to restore general real-world low-resolution images. While these methods can produce impressive results, they are still limited in model capacity and unsuitable for large upsampling factors. To scale up the model capacity of GANs, GigaGAN [21] introduces filter bank and attention layers to StyleGAN2 [23] and trains the model on billions of images. Even for 8× image super-resolution tasks, GigaGAN can effectively generate new content not present in the low-resolution image and produce realistic textures and fine-grained details.

Generative Video Models. Many video generation works are based on the VAEs [1,26,56], GANs [10,44,60], and autoregressive models [52]. LongVideoGAN [4] 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 [2,3,11,12,17]. Imagen Video [16] proposes pixel diffusion models for video super-resolution. Concurrent work Upscale-A-Video [63] adds temporal modules to a latent diffusion image upsampler [39] and finetunes it as a video super-resolution model. 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 [21] (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

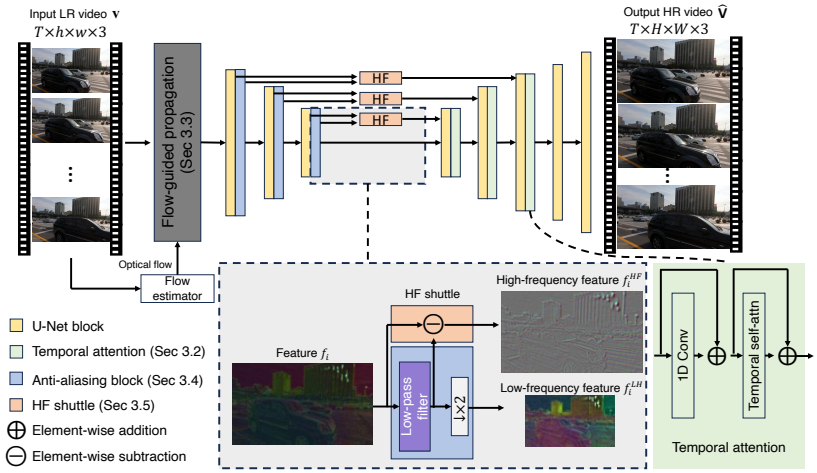

Fig. 3: Overview of our method for 4× upsampling. Our Video Super-Resolution (VSR) model is built upon the asymmetric U-Net architecture of the image GigaGAN upsampler [21]. To enforce temporal consistency, we first inflate the image upsampler into a video upsampler by adding temporal attention layers 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.

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 [21]. GigaGAN scales up the StyleGAN2 [23] 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 [3, 11, 16, 53, 57, 63]. 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 [16]. 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 [59] 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. 5).

To address these issues, we augment the input image with features aligned by optical flow. Specifically, we introduce a recurrent flow-guided feature propagation module (see Fig. 3) prior to the inflated GigaGAN, inspired by BasicVSR++ [7]. 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) [6, 7] 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 5.

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 chuck 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 [37]. 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 strided convolution layers in the upsampler encoder inspired by [61]. 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 [22]. The temporal flickering is mitigated, 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. 5, 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 [14], R1 regularization [34], LPIPS [62] and Charbonnier loss [9] 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, μ_{GAN}, μ_{R1}, μ_{LPIPS}, μ_{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 [6, 7, 33]: **REDS** [36] and **Vimeo-90K** [55]. 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

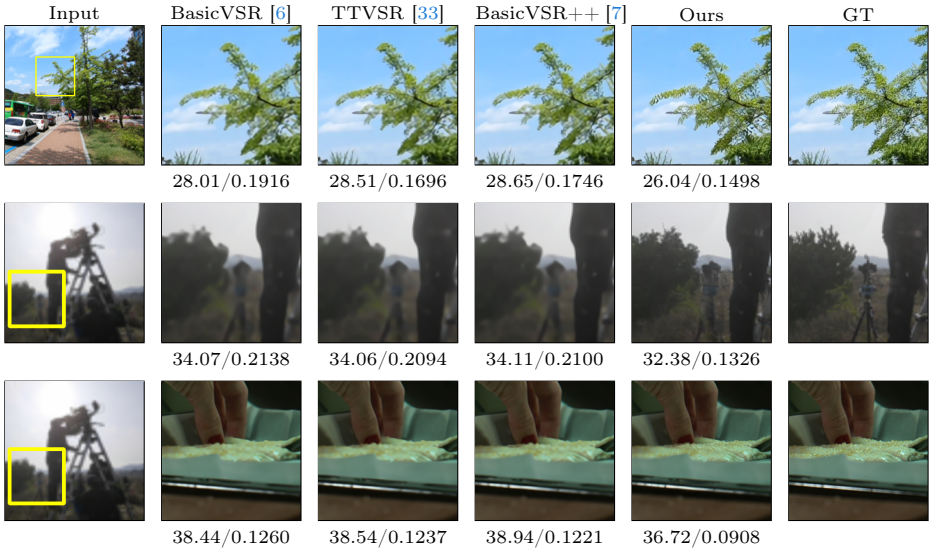

Fig. 4: Qualitative comparison with other baselines on public datasets (REDS4 [36], Vimeo-90K-T [55]. We show **PSNR/LPIPS** below each output frame. PSNR does not align well with human perception and favor blurry results. LPIPS is a preferred metric that aligns better with human perception. Compared to previous VSR approaches, our model can produce more realistic textures and more fine-grained details.

sequence contains seven frames with a resolution of 448×256. Following previous works [6, 7], 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 [32] and UDM10 [58], with different degradation algorithms (Bicubic Downsampling – BI and Blur Downsampling – BD). We follow MMagic [35] to perform degradation algorithms. All data are $4\times$ downsampled to generate LR frames following standard evaluation protocols [6, 7].

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** [62]. We report SSIM scores in the supplementary material. For temporal consistency, the warping error E_{warp} [24] 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 [46], and $M_t \in \{0, 1\}$ is a non-occlusion mask indicating non-occluded pixels [40]. However, as reported in Table 2, previous baselines such as BasicVSR++ or even simple bicubic upsampling achieve lower E_{warp} than ground truth high-resolution video

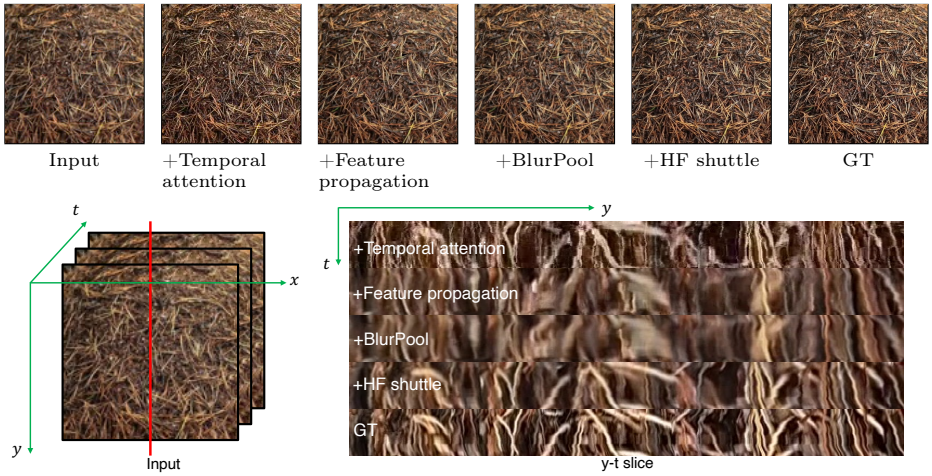

Fig. 5: Ablation study. Starting from the inflated GigaGAN (*+Temporal attention* in the figure), we progressively add components to demonstrate its effectiveness. With **temporal attention**, the local temporal consistency is improved compared to using image GigaGAN to upsample each frame independently. 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 project website). 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.

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** $E^{\text{ref}}_{\text{warp}}$. The referenced warping error between two frames is

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

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 **generated** frames $(\hat{\mathbf{X}}_t, \hat{\mathbf{X}}_{t+1})$ using RAFT [46].

Hyperparameters. We use a pretrained 4× 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 [38]. 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 $100,000$.

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 [36]. We report the quantitative results in Table 1. We also present a qualitative comparison in Fig. 5. We see that the **flow-guided feature propagation** brings a large LPIPS and E^{ref}_{warp} 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. 5). 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. 5). We strongly encourage the readers to watch the videos in the project website.

Model	LPIPS↓	$E^{ref}_{\text{warp}} \downarrow (\times 10^{-3})$
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>

Table 1: Ablation study. We use LPIPS to evaluate per-frame quality and $E^{ref}_{\text{warp}} \downarrow (\times 10^{-3})$ 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: <u>underline</u>.

4.3 Comparison with previous models

We conduct extensive experiments by comparing with 9 models including BasicVSR++ [7] and TTVSR [33]. At this point we cannot include Upscale-A-Video [63] since there is no available code. We report the quantitative comparison

Method	LPIPS↓	$E_{\text{warp}}\downarrow (\times 10^{-3})$	$E^{ref}_{\text{warp}}\downarrow (\times 10^{-3})$
Bicubic	0.3396	**1.161**	2.4232
EDVR [49]	0.2097	1.521	2.1429
MuCAN [28]	0.2162	1.562	2.1574
BasicVSR [6]	0.2023	1.371	2.1220
IconVSR [6]	0.1939	1.379	2.2119
TTVSR [33]	0.1836	1.390	**2.1178**
BasicVSR++ [7]	0.1786	1.401	2.1206
Ours	**0.1582**	2.313	2.1773
Ground truth	-	2.127	2.1272

Table 2: Comparison of VideoGigaGAN and previous VSR approaches in terms of 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 E^{ref}_{warp} for temporal consistency.

	BI degradation			BD degradation		
	REDS4 [36]	Vimeo-90K-T [55]	Vid4 [32]	UMD10 [58]	Vimeo-90K-T	Vid4
TOFlow [55]	-/27.98	-/33.08	-/25.89	-/36.26	-/34.62	-
RBPN [15]	-/30.09	-/37.07	-/27.12	-/38.66	-/37.20	-
PFNL [58]	-/29.63	-/36.14	-/26.73	-/38.74	-	-/27.16
EDVR [49]	0.2097/31.05	-/37.61	-/27.35	-/39.89	-/37.81	-/27.85
MuCAN [28]	0.2162/30.88	0.1523/37.32	-	-	-	-
BasicVSR [6]	0.2023/31.42	0.1616/37.18	0.2812/27.24	0.1148/39.96	0.1551/37.53	0.2555/27.96
IconVSR [6]	0.1939/31.67	0.1587/37.47	0.2739/27.39	0.1152/40.03	0.1531/37.84	0.2462/28.04
TTVSR [33]	0.1836/32.12	-	-	0.1112/40.41	0.1507/37.92	0.2381/28.40
BasicVSR++ [7]	0.1786/32.39	0.1506/37.79	0.2627/27.79	0.1131/40.72	0.1440/38.21	0.2390/29.04
RVRT [31]	0.1727/**32.74**	0.1502/**38.15**	0.2500/**27.99**	0.1100/**40.90**	0.1465/**38.59**	0.2219/**29.54**
Ours	**0.1582**/30.46	**0.1120**/35.97	**0.1925**/26.78	**0.1060**/36.57	**0.1129**/35.30	**0.1832**/27.04

Table 3: Quantitative comparisons of VideoGigaGAN and previous VSR approaches in terms of per-frame quality (LPIPS↓/PSNR↑) evaluated on multiple datasets. We also report SSIM scores in the supplementary material.

of the per-frame quality in Table 3. We show the comparison of temporal consistency for 6 of them in Table 2. Additionally, we provide qualitative comparisons in Fig. 4.

Per-frame quality. As shown in Table 3, our LPIPS outperforms all the other models by a large margin while showing a poorer performance of PSNR and SSIM (for SSIM, please refer to supplementary material). We observe that PSNR and SSIM do not align well with human perception and favor blurry results, as also reported in the literature [21, 39, 41]. Thus we consider LPIPS [62] as our core metric to evaluate per-frame quality as it is closer to the human perception. In Fig. 4, 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 [24], the widely used warping error metric favors a more blurry video. This is also illustrated in the Table 2. The simple bicubic upsampling method achieves the best performance for the commonly used warping error, which is much better than the GT warping

error. We proposed the referenced warping error (**RWE**) in Section 4.1 to address the issue of warping error favoring blurry results. In terms of the referenced warping error, our method is slightly worse than previous methods (0.05×10^{-3} compared to BasicVSR++ [7]). The newly proposed **RWE** is more suitable for evaluating the temporal consistency of upsampled videos. However, it is still biased towards more blurry results as seen in Table 2 (several methods, including BasicVSR, BasicVSR++, and TTVSR, are still better than the ground truth high-resolution videos). We leave a better metric of VSR temporal consistency for future works.

4.4 Analysis of the trade-off between temporal consistency and frame fidelity

To better understand the trade-off between the temporal consistency and per-frame quality, we include a visualization in Fig. 6. We can see that the previous VSR approaches focus on achieving better temporal consistency, but this comes with a sacrifice of per-frame quality (also see the qualitative comparisons in Fig. 4). Unlike previous VSR approaches, our final model - VideoGigaGAN, achieves a good balance between temporal consistency and per-frame quality. Compared to the base model GigaGAN, our proposed components significantly improve both the temporal consistency and per-frame quality by a large margin.

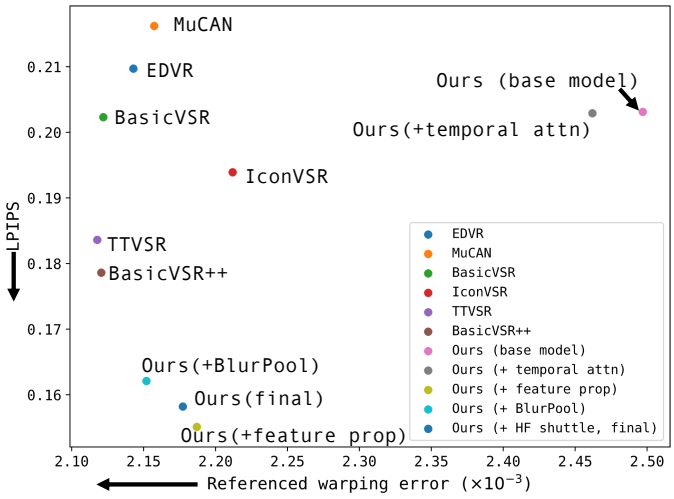

Fig. 6: Trade-off between per-frame quality (LPIPS↓) and temporal consistency (RWE↓). Our final model achieves a good balance between the temporal consistency and per-frame quality.

	#Params(M)	Runtime(ms)
RBPN [15]	12.2	1507
EDVR [49]	20.6	378
BasicVSR [6]	6.3	63
IconVSR [6]	8.7	70
BasicVSR++ [7]	7.3	77
Ours	369	295

Table 4: Comparison of model sizes and runtimes. We compute runtimes per frame on 320×180 to 1280×720 on REDS4 [36]. Our VideoGigaGAN model has a competitive runtime with a larger model size.

4.5 Model sizes and runtimes

We show the model sizes and runtimes for different models in Table 4. Our model has a large size for its generative capacity, and still has a competitive inference speed compared to previous feedforward VSR methods. Unlike diffusion-based video super-resolution models [16, 63] that require iterative denoising processes, our VideoGigaGAN can generate outputs in a *single feedforward pass* with much faster inference speed. We also experimented with scaling previous feed-forward models such as BasicVSR++ [7]. However, previous VSR models do not have good scalability and show unstable training when scaling up as also discussed in [21].

4.6 8× video upsampling

Our model is capable for 8x video upsampling with both good temporal consistency and per-frame quality with rich details. We encourage readers to visit our project website for more results.

5 Limitations

Our model encounters challenges when processing extremely long videos (e.g., 200 frames or more). This difficulty arises from misguided feature propagation caused by inaccurate optical flow in such extended video sequences. Additionally, our model does not perform well in handling small objects, such as text and characters, as the information pertaining to these objects is significantly lost in the LR video input. Examples of these failure cases are illustrated in Fig. 7.

6 Conclusions

We present a novel generative VSR model, VideoGigaGAN, that can upsample input low-resolution videos to high-resolution videos with both high-frequency

(a) Extremely long video (b) Small objects

Fig. 7: Limitations. Our approach has some limitations. (a) When the video is **extremely long**, the feature propagation becomes inaccurate, which may introduce undesired artifacts like incorrect propagated patterns. (b) Our model cannot handle well **small objects**, e.g., small characters.

details and temporal consistency. Previous VSR approaches often use regression-based networks and tend to generate blurry results. To this end, our VSR model built upon the powerful generative image upsampler – GigaGAN. We identify several issues when applying GigaGAN to video super-resolution tasks including temporal flickering and aliased artifacts. To address these issues, we introduce new components to the GigaGAN architecture that can effectively improve both the temporal consistency and per-frame quality. Our results demonstrate that VideoGigaGAN strike a balance in addressing the consistency-quality dilemma of VSR compared to previous methods.

References

1. Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R.H., Levine, S.: Stochastic variational video prediction. In: ICLR (2018) 4
2. 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) 4
3. 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) 4, 6
4. 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) 4

5. 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) 3
6. 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) 2, 3, 6, 8, 9, 12, 14
7. 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, 3, 6, 8, 9, 11, 12, 13, 14
8. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Investigating tradeoffs in real-world video super-resolution. In: CVPR (2022) 2, 4
9. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: ICIP (1994) 8
10. 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) 4
11. 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) 4, 6
12. 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) 4
13. 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
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NeurIPS (2017) 8
15. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. In: CVPR (2019) 12, 14
16. 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, 4, 6, 14
17. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: NeurIPS (2022) 4
18. Huang, Y., Wang, W., Wang, L.: Bidirectional recurrent convolutional networks for multi-frame super-resolution. NeurIPS (2015) 3
19. Huang, Y., Wang, W., Wang, L.: Video super-resolution via bidirectional recurrent convolutional networks. TPAMI **40**(4), 1015–1028 (2017) 3
20. Isobe, T., Jia, X., Gu, S., Li, S., Wang, S., Tian, Q.: Video super-resolution with recurrent structure-detail network. In: ECCV (2020) 2, 3
21. 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, 4, 5, 12, 14
22. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: NeurIPS (2021) 7
23. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020) 4, 5
24. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV (2018) 9, 12
25. 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) 4

26. Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic adversarial video prediction. arXiv preprint arXiv:1804.01523 (2018) 4
27. Li, F., Zhang, L., Liu, Z., Lei, J., Li, Z.: Multi-frequency representation enhancement with privilege information for video super-resolution. In: CVPR (2023) 3
28. Li, W., Tao, X., Guo, T., Qi, L., Lu, J., Jia, J.: Mucan: Multi-correspondence aggregation network for video super-resolution. In: ECCV (2020) 3, 12
29. Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., Van Gool, L.: Vrt: A video restoration transformer. TIP (2024) 3
30. 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) 3
31. 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) 12
32. Liu, C., Sun, D.: On bayesian adaptive video super resolution. TPAMI **36**(2), 346–360 (2013) 9, 12
33. Liu, C., Yang, H., Fu, J., Qian, X.: Learning trajectory-aware transformer for video super-resolution. In: CVPR (2022) 1, 8, 9, 11, 12
34. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: ICML (2018) 8
35. MMagic Contributors: MMagic: OpenMMLab multimodal advanced, generative, and intelligent creation toolbox. https://github.com/open-mmlab/mmagic (2023) 9
36. 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) 8, 9, 11, 12, 14
37. Nyquist, H.: Certain topics in telegraph transmission theory. Transactions of the American Institute of Electrical Engineers **47**(2), 617–644 (1928) 7
38. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR (2017) 11
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 4, 12
40. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos. In: GCPR (2016) 9
41. 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) 12
42. Sajjadi, M.S., Vemulapalli, R., Brown, M.: Frame-recurrent video super-resolution. In: CVPR (2018) 3
43. Shi, S., Gu, J., Xie, L., Wang, X., Yang, Y., Dong, C.: Rethinking alignment in video super-resolution transformers. In: NeurIPS (2022) 3
44. Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In: CVPR (2022) 4
45. Tao, X., Gao, H., Liao, R., Wang, J., Jia, J.: Detail-revealing deep video super-resolution. In: ICCV (2017) 3
46. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020) 9, 10
47. Tian, Y., Zhang, Y., Fu, Y., Xu, C.: Tdan: Temporally-deformable alignment network for video super-resolution. In: CVPR (2020) 3
48. 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) 3

49. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: CVPRW (2019) 12, 14
50. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: ICCVW (2021) 2, 4
51. 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, 4
52. Weissenborn, D., Täckström, O., Uszkoreit, J.: Scaling autoregressive video models. In: ICLR (2020) 4
53. 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) 6
54. Xie, L., Wang, X., Shi, S., Gu, J., Dong, C., Shan, Y.: Mitigating artifacts in real-world video super-resolution models. In: AAAI (2023) 4
55. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV **127**(8), 1106–1125 (2019) 3, 8, 9, 12
56. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. arXiv preprint arXiv:2104.10157 (2021) 4
57. 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) 6
58. 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) 9, 12
59. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023) 6
60. Zhang, Q., Yang, C., Shen, Y., Xu, Y., Zhou, B.: Towards smooth video composition. In: ICLR (2023) 4
61. Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019) 7
62. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 8, 9, 12
63. 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) 4, 6, 11, 14