Efficient Video Super-Resolution for Real-time Rendering with Decoupled G-buffer Guidance

Mingjun Zheng* Long Sun* Jiangxin Dong Jinshan Pan†
School of Computer Science and Engineering, Nanjing University of Science and Technology
{mingjunzheng, cs.longsun, jxdong, jspan}@njust.edu.cn

Abstract

*Latency is a key driver for real-time rendering applications, making super-resolution techniques increasingly popular to accelerate rendering processes. In contrast to existing methods that directly concatenate low-resolution frames and G-buffers as input without discrimination, we develop an asymmetric UNet-based super-resolution network with decoupled G-buffer guidance, dubbed **RDG**, to facilitate the spatial and temporal feature exploration for minimizing performance overheads and latency. We first propose a dynamic feature modulator (DFM) to selectively encode the spatial information to capture precise structural information. We then incorporate auxiliary G-buffer information to guide the decoder to generate detail-rich, temporally stable results. Specifically, we adopt a high-frequency feature booster (HFB) to adaptively transfer the high-frequency information from the normal and bidirectional reflectance distribution function (BRDF) components of the G-buffer, enhancing the details of the generated results. To further enhance the temporal stability, we design a cross-frame temporal refiner (CTR) with depth and motion vector constraints to aggregate the previous and current frames. Extensive experimental results reveal that our proposed method is capable of generating high-quality and temporally stable results in real-time rendering. The proposed RDG-s produces **1080P** rendering results on a RTX 3090 GPU with a speed of **126 FPS**. Our source codes and pre-trained models are available at: https://github.com/sunny2109/RDG.*

1. Introduction

Real-time rendering is widely used in a variety of applications, such as video games, virtual reality, and movies, to quickly produce high-quality and high-resolution images. With the breakthrough development of graphics hardware, real-time rendering technology has made remarkable improvements. However, it is still extremely challenging to render high-quality, high-resolution images in real-time.

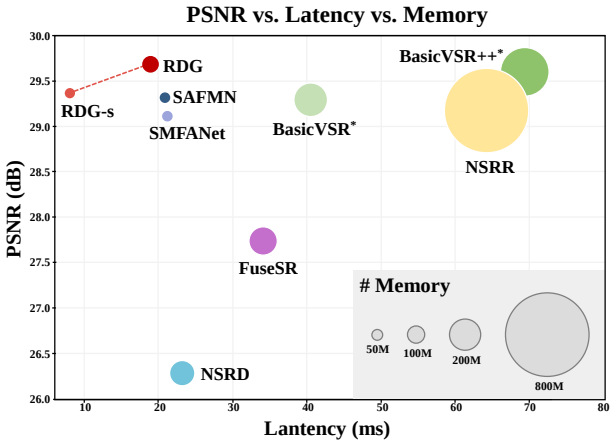


Figure 1. Comparisons of our proposed methods and state-of-the-art ones in terms of accuracy, latency, and GPU consumption. Circle sizes indicate the usage of GPU memory. The proposed RDG family obtain better trade-off between performance and efficiency.

To meet this challenge, deep learning super-sampling (DLSS) techniques [1, 3, 4] have been widely adopted in the industry, which perform super-resolution (SR) on the rendered low-resolution (LR) images to generate the corresponding high-resolution outputs in real time. Although these methods effectively decrease rendering time, it remains open to strike a balance between latency and quality.

A number of approaches have been developed for the above tasks, including video super resolution (VSR) [8, 9, 31, 40, 53] and efficient super resolution (ESR) [21, 34, 47, 48, 60]. While these methods have achieved impressive results, they cannot be directly used for real-time rendering. For video super-resolution, there are three main reasons that affect its application. First, most existing VSR methods rely on bidirectional feature propagation, which conflicts with the rendering process, as no future frame information can be obtained. Next, feature warping operation for temporal fusion requires time to estimate the optical flow, which increases considerably as the input resolution increases, thus significantly limiting the inference efficiency of the model.

*Co-first authorship
†Corresponding author

Lastly, recent advanced VSR approaches [31, 40, 45] use the Transformer framework to improve SR performance, which is computationally expensive and incompatible with real-time rendering tasks. As for the ESR task, since the input is a single-frame image, the reconstruction process cannot make use of the information from adjacent frames, resulting in temporally unstable reconstruction results. Furthermore, it is difficult to recover detail-rich results as the current frame has limited texture information.

Unlike solely relying on low-resolution video sequences, another branch [16, 28, 57, 58, 61] facilitates high-fidelity detail generation by incorporating auxiliary G-buffer information (i.e., depth, motion vector, normal, and BRDF). These methods directly concatenate G-buffers with historical frame information as input, which has limited ability to transfer useful information for latent frame recovery as demonstrated in [28]. To overcome this problem, [28, 61] introduce radiance demodulation. While it is feasible to inject high-resolution detail information in some scenes, when materials violate the partitioning and approximation assumptions, such as translucent and anisotropic materials, such an approach is not possible and leads to severe remodulation artifacts and errors, as shown in Figure 3.

In this paper, we propose an effective real-time video super-resolution network to distinctively explore G-buffers and spatial-temporal information from LR inputs for generating high-fidelity up-sampled results. Motivated by the success of u-shaped architectures [43, 59], we develop an asymmetric UNet-based network with decoupled G-buffer guidance to facilitate spatial and temporal feature exploration. First, the encoder part consists of several dynamic feature modulators (DFMs) used to selectively encode spatial information to capture precise structural information. Then we incorporate auxiliary G-buffer information to guide the decoder to generate detail-rich, temporally stable results. Specifically, we adopt a high-frequency feature booster (HFB) to adaptively transfer the high-frequency information from the normal and BRDF components of the G-buffer, enhancing the details of the generated results. To enhance temporal stability, we utilize the motion vector to warp the historical feature as an initial alignment reference. To avoid the alignment errors, we design a cross-frame temporal refiner (CTR) with depth map from G-buffer to accurately aggregate previous alignment features with the current frame. Extensive experimental results reveal that our proposed method is capable of generating high-quality and temporally stable results in real-time rendering.

Our contributions can be summarized as follows:

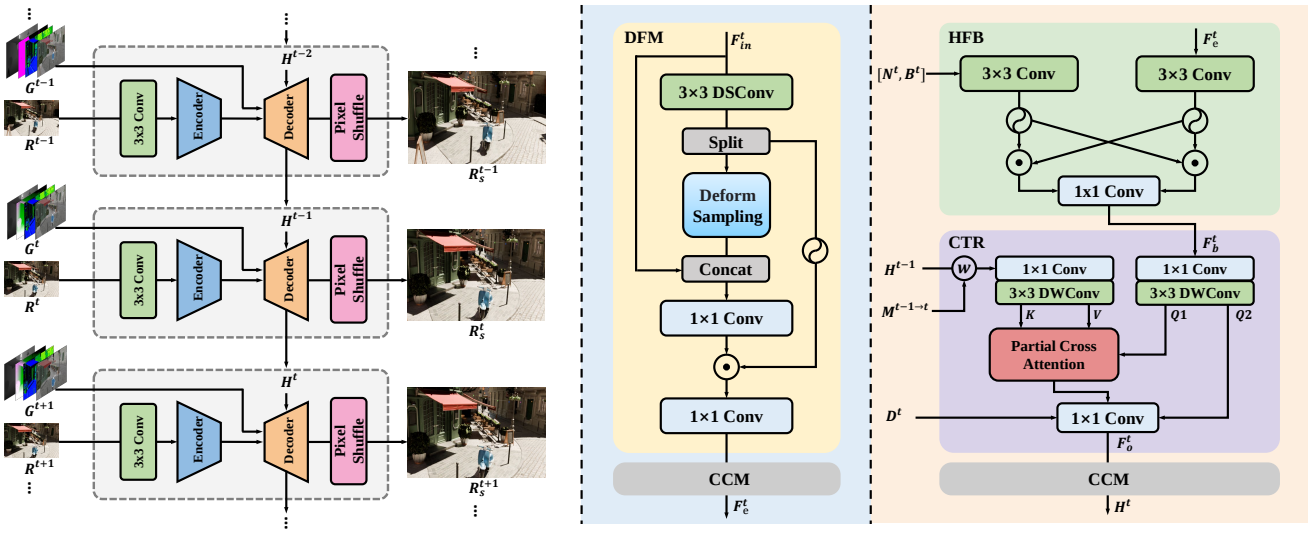
- We propose a lightweight dynamic feature modulator (DFM) to compose the encoder, which is used to selectively encode contextual information for acquiring accurate structural representations.
- We develop a simple yet effective high-frequency feature booster (HFB) to adaptively transfer the high-frequency information from the normal and BRDF for fine-grade frame reconstruction.
- We present a cross-frame temporal refiner (CTR) to efficiently propagate useful information from the previous frame and avoid error accumulation for better super-resolution.
- We conduct quantitative and qualitative evaluations for our proposed method on rendering test datasets, and the results demonstrate that our method achieves a favorable trade-off between model complexity and reconstruction performance.

2. Related Work

Video Super-Resolution. Compared to sliding window-based [6, 23, 29, 50, 53] video super-resolution (VSR), recurrent-based [8, 9, 15, 18, 20, 22] VSR methods have better temporal coherence. TTVSR [33] develops a trajectory-aware Transformer, which significantly reduces computational costs and enables long-range modeling in videos. BasicVSR [8] utilizes bidirectional propagation to reconstruct the current frame, which estimates bidirectional optical flows to align both historical and future frames. BasicVSR++ [9] improves BasicVSR and achieves excellent performance by incorporating second-order grid propagation and flow-guided deformable alignment. While these VSR methods yield excellent reconstruction results, bidirectional propagation incurs significant latency, making real-time video super-resolution infeasible.

Efficient Super-Resolution. Efficient super-resolution (ESR) methods [21, 30, 34, 47, 48, 60] aim to enhance visualization quality while reducing computational costs. IMDN [21] introduces the information multi-distillation block to refine the features, significantly reducing the complexity of the model. BSRN [30] employs lightweight blueprint convolution to reduce computational costs. SAFMN [48] learns multi-scale feature representations and aggregates these features for efficient image super-resolution. SMFANet [60] explores non-local and local information to enhance image reconstruction. Although these ESR methods can achieve real-time image super-resolution, it is difficult to maintain temporal consistency as they cannot exploit inter-frame information.

Real-time Rendering Super-Resolution. In the context of constrained hardware capabilities, early researchers use a computationally efficient supersampling algorithm to solve the problem of rendering image aliasing. MSAA [5] is a traditional supersampling method that processes one pixel at a time to avoid redundant subpixel sampling of the same polygon. FXAA [36] addresses the undersampling problem by attenuating subpixel features, improving temporal stability. MLAA [42] identifies edges by analyzing color discontinuities and blends the colors of pixels spanning these edges to



(a) Overall Architecture (b) Encoder Block (c) Decoder Block

Figure 2. **Network architecture of the proposed RDG.** RDG is an asymmetric UNet-based rendering network with decoupled G-buffer guidance. For the encoder module, we introduce multiple dynamic feature modulators (DFMs) to encode structural information progressively. For the decoder module, we incorporate auxiliary G-buffer information to enrich details and improve temporal stability. Within each decoder block, we propose a high-frequency feature booster (HFB) to transfer high-frequency details from the G-buffer and a cross-frame temporal refiner (CTR) to improve the robustness of the temporal propagation. Each DFM and CTR block is followed by a CCM [48] layer to perform channel interactions and enhance the local modeling capability.

enhance image boundaries. SMAA [24] improves MLAA by refining the boundary detection mechanism, achieving better visual quality while maintaining low performance overhead. TAA [25] mitigates image aliasing and flicker by utilizing temporal sampling, thereby enhancing the smoothness of visual rendering. However, these methods rely on a priori information and the processed results tend to be blurred in dynamic scenarios.

To alleviate this challenge, researchers utilize large-scale data to train deep learning-based super-sampling models to solve the real-time rendering problem. DLSS [1] and XeSS [4] leverage neural networks to upscale low-resolution frames and significantly improve the visual experience of real-time rendering. NSRR [58] employs a U-Net architecture to integrate zero-upsampling historical frame features. NSRD [28] and FuseSR [61] exploit the irradiance demodulation technique to improve the rendering quality, which processes the low-resolution irradiance through a neural network, and then re-modulates the enhanced irradiance maps into a high-resolution image. While these approaches improve the performance of super-sampling, they can lead to severe re-modulation artifacts and errors with translucent and anisotropic materials. Moreover, it remains a challenge to strike a balance between performance, computational overhead, and latency. In this paper, we propose an efficient neural model with auxiliary G-buffer information to generate high-quality and temporally stable results at a lower computational cost.

3. Proposed Method

Our proposed RDG aims to reconstruct detail-rich, time-stable high-resolution rendering results. In this section, we first introduce the overall architecture and then discuss the implementation details of our method.

3.1. Overall Architecture

Figure 2 (a) illustrates the overall architecture of our method. Unlike existing methods [28, 58, 61], which indiscriminately concatenate low-resolution frames with G-buffer as input, our approach decouples the G-buffer components based on their diverse attributes to facilitate spatial and temporal feature exploration.

Given a sequence of low-resolution rendered frames $\{..., R^{t-1}, R^t, R^{t+1}, ...\}$ and their corresponding G-buffers $\{..., G^{t-1}, G^t, G^{t+1}, ...\}$, our proposed RDG processes t-th frame R^t and G^t as follows: a 3×3 convolutional layer is first applied to extract the shallow features from the input R^t. The shallow features are then encoded under an encoder module to extract representative structural information, where the feature encoding is performed by our dynamic feature modulator (DFM). After the encoder module, the encoded features are passed into the decoder part to generate finer features by incorporating decoupled G-buffer information, where G^t includes normal N^t, BRDF B^t, depth D^t, and motion vector $M^{t-1 \to t}$. Within the decoder module, we transfer the high-frequency details contained in N^t and B^t by a high-frequency feature booster (HFB), and im-

prove temporal stability by exploiting $M^{t-1\to t}$ and D^t to form a cross-frame temporal refiner (CTR). Finally, the aggregated features are used to generate the high-resolution output R_s^t through a 3×3 convolutional layer and a PixelShuffle [46] layer.

3.2. Dynamic Feature Modulator

The feature modulation mechanism [17, 48, 54, 60, 62] demonstrates promising performance on the ESR task, but it suffers from deficiencies in acquiring non-local information modulated by the guidance map. Previous approaches rely on pooling operation [48, 60] or large kernel depth-wise convolution [54, 62] to capture non-local features, where the pooling operation tends to smooth the spatial information and large kernel depth-wise convolution lacks cross-channel interaction. To improve these limitations, we develop a deformable sampling strategy to model non-local feature interactions across channels.

Inspired by [10, 13], given an input feature $X \in \mathbb{R}^{H\times W\times C_{in}}$, where $H \times W$ denotes the spatial size and C_{in} is the number of channels, the deformable sampling procedure can be represented as:

$$\begin{aligned} \hat{X}_{[i,j,:]} &= \sum_{c=0}^{C_{in}} w_{[c,:]} \cdot X_{[i+\Delta_i(c),j+\Delta_j(c),c]} + b, \\ i &\in \{0,1,...,H-1\}, \quad j \in \{0,1,...,W-1\}, \end{aligned} \tag{1}$$

where $\hat{X}_{[i,j,:]}$ are the sampling values of all channels at the spatial position (i, j), $w \in \mathbb{R}^{C_{in}\times C_{out}}$ and $b \in \mathbb{R}^{c_{out}}$ are learnable parameters. $\Delta_i(c)$ and $\Delta_j(c)$ are the spatial offsets with a sampling size of (S_h, S_w) on the c-th channel, which are defined as:

$$\begin{aligned} \Delta_i(c) &= (c \bmod S_h) - 1, \\ \Delta_j(c) &= (\lfloor \frac{c}{S_h} \rfloor \bmod S_w) - 1. \end{aligned} \tag{2}$$

When the channel number is less than $S_h \times S_w$, this sampling manner will lead to incomplete feature interactions, thus we choose an asymmetric sampling way for our dynamic feature modulator to increase the sampling density.

Based on the above deformable sampling, we describe the proposed dynamic feature modulator as follows. Given the t-th frame shallow feature $F_{in}^t \in \mathbb{R}^{H\times W\times C}$, we first project F_{in}^t into two part features: $F_i \in \mathbb{R}^{H\times W\times C}$ and $F_s \in \mathbb{R}^{H\times W\times C}$. The asymmetric deformable sampling operators are then introduced in F_s to extract finer non-local information $\hat{F}_s$. After that, we obtain the modulated guidance map F_g by aggregating information from $\hat{F}_s$ and F_{in}^t. This process can be formulated as follows:

$$\begin{aligned} F_i, F_s &= \mathcal{S}(DSConv_{3\times3}(F_{in}^t)), \\ \hat{F}_s &= DS_{1\times7}(DS_{7\times1}(F_s)), \\ F_g &= Conv_{1\times1}(\mathcal{C}[\hat{F}_s, F_{in}^t]), \\ F_e^t &= Conv_{1\times1}(\phi(F_i) \odot F_g), \end{aligned} \tag{3}$$

where $DSConv_{3\times3}(\cdot)$ is formed by 3×3 separable convolution [12], $DS_{1\times7}(\cdot)$ and $DS_{7\times1}(\cdot)$ are deformable sampling operators with 1×7 and 7×1 sampling sizes, respectively, $\mathcal{S}(\cdot)$ denotes a channel splitting operation, $\mathcal{C}(\cdot)$ represents a concatenation operation, $\phi(\cdot)$ refers to the GELU activation function [19], $\odot$ represents the element-wise product operation, and $Conv_{1\times1}(\cdot)$ is only the 1×1 convolution, and $F_e^t \in \mathbb{R}^{H\times W\times C}$ is the output feature.

3.3. High-frequency Feature Booster

As the normal and BRDF components of the G-buffer store the geometry and material information in the scene, which contains rich high-frequency detail and texture information, we thus develop a parameter-efficient high-frequency feature booster (HFB) to transfer high-frequency features from normal N^t and BRDF B^t.

As illustrated in Figure 2 (c), the encoder output feature F_e^t and the G-buffer components (i.e. N^t and B^t) are projected into hidden features $\hat{F}_e^t$ and I^t by a 3×3 convolution, respectively. The cross-gating mechanism [51] is then applied to fuse $\hat{F}_e^t$ and I^t, generating detail-boosted results F_b^t. The process is described by the following equations:

$$\begin{aligned} \hat{F}_e^t &= Conv_{3\times3}(F_e^t), \\ I^t &= Conv_{3\times3}(\mathcal{C}[N^t, B^t]), \\ F_b^t &= Conv_{1\times1}(\mathcal{C}[\phi(\hat{F}_e^t) \odot I^t, \phi(I^t) \odot \hat{F}_e^t]), \end{aligned} \tag{4}$$

where $F_b^t \in \mathbb{R}^{H\times W\times C}$.

3.4. Cross-frame Temporal Refiner

Exploiting temporal information from adjacent frames can effectively improve the reconstruction performance and temporal consistency. Most of the existing methods [8, 9, 28, 28, 61] use flow fields or motion vectors to warp the propagated frames. However, in cases involving object occlusion or long-distance displacements, the warped frames may contain misalignment errors, such as blurred or deformed motion boundaries (see Figure 4). Propagation of these unreliably warped frames can lead to time-dragging or flicker artifacts.

To address this problem, we develop a lightweight and efficient cross-frame temporal refiner (CTR), which combines the motion vector with the depth map to provide accurate boundary information, thus ensuring temporal consistency of the fine details. The CTR block first warps the previous hidden feature H^{t-1} by the motion vector $M^{t-1\to t}$. We then introduce an efficient partial cross attention (PCA) to refine the warped feature $\hat{H}^{t-1}$. Specifically, we take the current frame feature F_b^t as the query Q, the warped feature $\hat{H}^{t-1}$ as the key K and value V, and perform a projection on the above features using a combination of 1×1 convolution and 3×3 depth-wise convolution. After that, we

Table 1. **Comparison with SOTA methods.** All PSNR/SSIM results are calculated on the **RGB**-channel. #Memory and #Latency are measured on an input frame of size 270×480. The best and second-best performances are highlighted by bolding and underlining, respectively. * indicates that we adjusted the parameter settings of the model for faster inference.

Methods	Efficiency Metrics			Testing Data (PSNR/SSIM/VMAF)								
	#Params	#Memory	#Latency	Bar	Bistro	Forest	NYC	Square	Squre-night	Tenshu	ZeroDay	Average
SAFMN [48]	0.240M	**49.46M**	20.88ms	30.85 0.8192 90.31	25.92 0.8104 79.33	25.77 0.6439 61.32	29.21 0.906 93.48	26.49 0.8332 90.64	33.01 0.9048 82.49	31.46 0.8913 73.78	31.91 0.901 81.63	29.33 0.8387 81.62
SMFANet [60]	**0.197M**	56.72M	21.23ms	30.78 0.8164 92.27	25.93 0.8052 81.50	25.74 0.6421 62.92	28.27 0.8848 91.04	26.39 0.8264 92.80	32.81 0.9001 87.70	31.26 0.8860 72.58	31.76 0.9025 82.75	29.12 0.8329 82.95
BasicVSR* [8]	1.760M	249.68M	40.34ms	30.84 0.8182 93.64	25.92 0.8106 79.62	25.62 0.6356 63.19	29.32 0.9064 93.82	26.50 0.8335 90.29	32.92 0.9025 86.54	31.35 0.8881 75.34	31.90 0.9041 84.64	29.30 0.8374 83.38
BasicVSR++* [9]	2.408M	404.60M	67.12ms	30.96 0.8220 93.84	26.10 0.8159 79.66	25.76 0.6462 64.31	29.87 0.9163 96.97	26.70 0.8399 91.04	33.17 0.9084 87.28	31.64 0.8933 76.36	32.07 0.9053 85.65	29.53 0.8716 84.39
NSRR [58]	0.535M	766.97M	66.86ms	30.86 0.8218 91.93	25.93 0.8125 77.80	25.79 0.6487 63.70	29.24 0.9075 92.36	26.48 0.8333 88.51	33.14 0.9058 85.64	31.36 0.8898 75.14	31.94 0.8985 83.76	29.34 0.8397 82.35
NSRD [28]	1.61M	166.50M	23.06ms	27.75 0.8287 71.29	22.33 0.7066 64.73	21.33 0.5276 54.66	28.75 0.8133 77.32	23.08 0.7658 69.78	31.50 0.8880 65.03	27.13 0.8881 63.99	28.42 0.8648 72.98	26.29 0.7854 67.47
FuseSR [61]	2.247M	198.25M	33.99ms	28.09 0.8228 79.59	24.33 0.7550 69.45	21.94 0.5705 54.58	29.44 0.9100 75.58	24.58 0.8161 77.23	32.88 0.8907 78.49	28.88 0.8830 68.34	31.79 0.8978 71.42	27.74 0.8182 71.83
RDG-s (Ours)	0.304M	61.45M	**7.93ms**	**31.03** 0.8274 96.29	26.18 0.8159 83.94	25.74 0.6597 70.14	28.93 0.8989 98.36	26.65 0.8326 95.09	33.19 0.9073 89.40	31.39 0.8891 78.75	31.83 0.9061 85.82	29.37 0.8421 87.22
RDG (Ours)	1.474M	101.76M	18.78ms	30.96 **0.8299** **98.62**	**26.32** **0.8231** **86.04**	**25.93** **0.6601** **71.52**	**30.20** **0.9204** **99.12**	**27.02** **0.8425** **96.65**	**33.21** **0.9112** **92.85**	**31.70** **0.8937** **79.84**	**32.16** **0.9141** **87.06**	**29.69** **0.8494** **88.96**

split feature Q into Q_1 and Q_2 equally across channels and use Q_1 with feature K, V for attention computation [59]. Finally, the output O_{att} of the PCA, Q_2 and the depth D^t are concatenated and fused by a 1×1 convolution to generate the refined temporal feature F_o^t . This process can be formulated as:

$$\begin{aligned} \hat{H}^{t-1} &= Warp(H^{t-1}, M^{t-1\to t}), \\ Q_1, Q_2 &= \mathcal{S}(DW_{3\times3}(Conv_{1\times1}(F_b^t))), \\ K &= DWConv_{3\times3}(Conv_{1\times1}(\hat{H}^{t-1})), \\ V &= DWConv_{3\times3}(Conv_{1\times1}(\hat{H}^{t-1})), \\ O_{att} &= Softmax(Q_1 \cdot K^T) \cdot V, \\ F_o^t &= Conv_{1\times1}(\mathcal{C}[O_{att}, Q_2, D^t]), \end{aligned} \tag{5}$$

where $DWConv_{3\times3}(\cdot)$ is a 3×3 depth-wise convolution, $\{Q_1, Q_2\} \in \mathbb{R}^{C/2\times(HW)}$, $\{K, V\} \in \mathbb{R}^{C\times(HW)}$, $O_{att} \in \mathbb{R}^{C/2\times(HW)}$ and $F_o^t \in \mathbb{R}^{H\times W\times C}$.

4. Experimental Results

4.1. Datasets and Implementation

Rendering datasets. Our dataset consists of 20 scenes rendered using the Cycles engine [2]. These scenes are rich in content, encompassing both realistic and stylized environments, and include a diverse array of objects and materials. To capture the dynamic changes in these scenes, we set up different moving cameras for each scene and acquire 180 video clips as the training dataset, each clip consisting of 100 frames. For each frame, we render the paired LR-HR data with a size of 270×480 pixels and 1080×1920 pixels, respectively. Additionally, we also save the corresponding G-buffer components, which include normal, BRDF, depth, and motion vector.

For the testing data, we use 8 representative scenes to verify the effectiveness of the model, including Bar [37], Bistro [37], NYC [7], Forest [49], Square [39], Square-night [39], Tenshu [38], and ZeroDay [56]. Each scene consists of 200 rendered frames. For further details on the datasets, please refer to the supplementary material.

Implementation details. Our proposed RDG is optimized by Adam [26] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The batch size is 8 and the patch size of input LR frames is 96×96. We set the initial learning rate to 1×10^{-3} and the minimum one to 1×10^{-5}, which is updated by the cosine annealing scheme [35]. The number of total iterations for all experiments is set to 300,000. All experiments are conducted with the PyTorch framework on an NVIDIA GeForce RTX 3090 GPU. We train two versions of RDG with different complexities. The standard RDG has 36 channels while the small version, RDG-s, has 16 channels. The employed loss function consists of three components: a charbonnier loss [27], a temporal consistency loss [14], and an FFT-based frequency loss [11, 47], with the weights $\{1, 1, 0.05\}$.

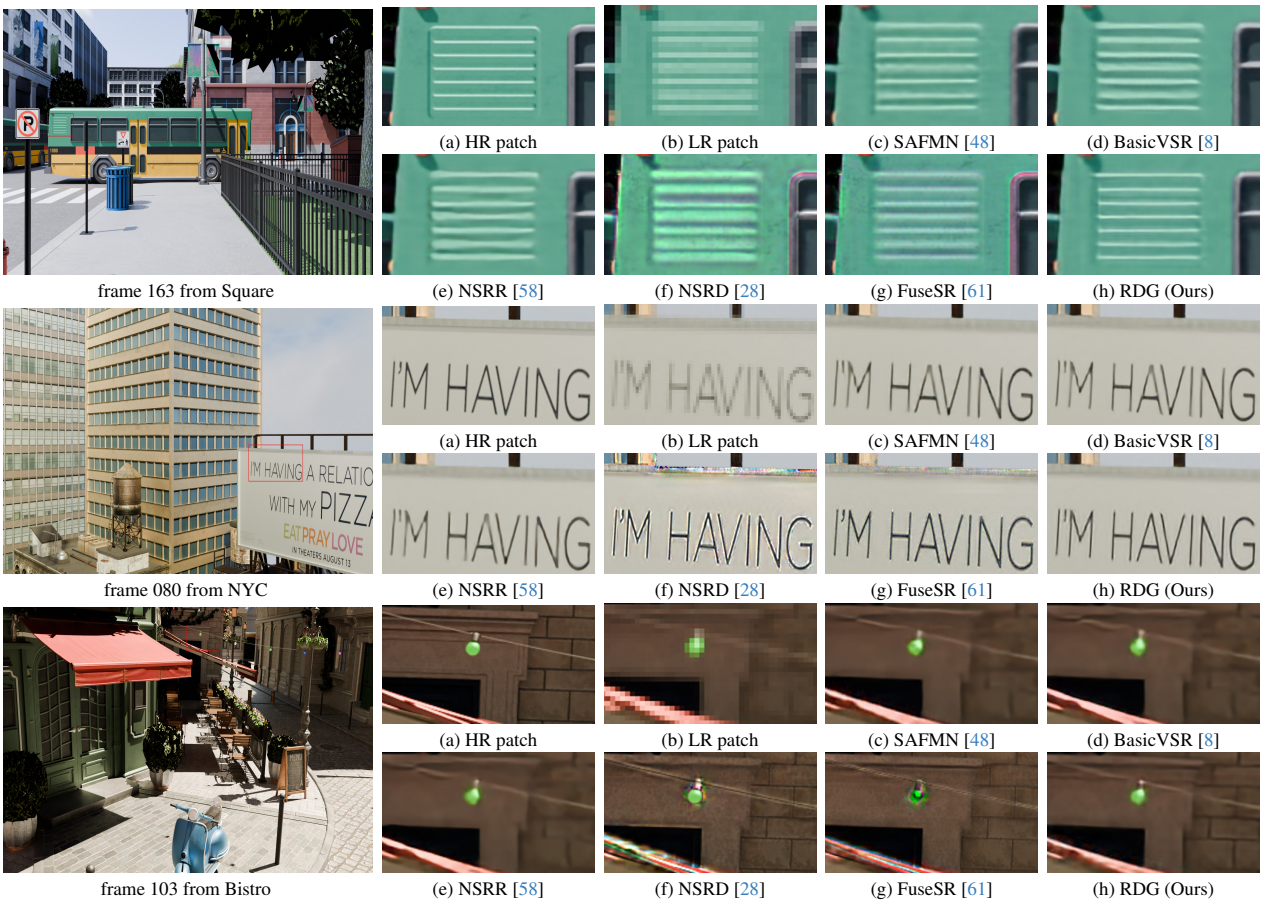


Figure 3. **Super-resolved results on the testing dataset.** The rendered results in (c)-(g) still contain significant blurring or artifacts. The proposed method generates a sharper image. For example, lines, characters and shapes are much clearer.

4.2. Comparisons with State-of-the-art Methods

To fully evaluate the performance of our method, we first compare our RDG family with state-of-the-art ESR methods and VSR methods, including SAFMN [48], SMFANet [60], BasicVSR [8], and BasicVSR++ [9], and further compare state-of-the-art real-time rendering super-resolution methods, including NSRR [58], NSRD [28], and FuseSR [61]. Note that BasicVSR and BasicVSR++ are computationally expensive and unable to achieve fast inference, in this work we reduce the channel number and the number of residual blocks in the propagation process to better meet the demands of real-time rendering. Moreover, there is no official implementation of NSRR, so we reproduce it based on the paper [58].

The quantitative comparisons on test datasets are reported in Table 1, where **all the comparison methods listed are re-trained in our training dataset**. In addition to PSNR, SSIM [55] and VMAF [41] metrics, we also list the number of parameters (#Params), peak GPU memory (#Memory) and per-frame inference time (#Latency). We calculate the #Memory and #Latency under a setting of super resolving 500 LR frames to 1080×1920 pixels.

Quantitative comparison. Our proposed RDG with auxiliary G-buffers can generate detail-rich, temporally stable high-resolution rendering video. Table 1 shows that our RDG achieves better performance on almost all test datasets. With the benefit of the decoupled G-buffer guidance, our approaches can produce high-quality results. Compared to NSRR [58], which directly concatenates image and G-buffers as input, our RDG outperforms it by 0.35dB on average, and even the RDG-s, which has much fewer parameters, is also competitive in performance with NSRR [58]. Compared with the radiance demodulation approaches [28, 61], our RDG family avoid unpleasant re-modulation errors on scenes containing translucent and anisotropic materials, and the proposed RDG has a significant PSNR gain, which is 3.4dB and 1.95dB on average higher than NSRD [28] and FuseSR [61], respectively. Furthermore, benefiting from our CTR which allows us to accurately aggregate the previous and current frames for better temporal consistency, RDG is 4.57 higher than BasicVSR++ [9] in terms of VMAF value.

Qualitative comparisons Figure 3 presents visual comparisons between our proposed methods and listed meth-

Table 2. Effectiveness of the DFM.

	#Params	#Latency	PSNR	VMAF
RDG-s	**0.303M**	**7.93ms**	**29.37**	**87.22**
DFM → ResBlock	0.301M	6.58ms	29.17	87.08
DS → Pooling+$DW_{3\times3}$	0.297M	7.57ms	29.21	87.10
DS → $DW_{15\times15}$	0.318M	8.13ms	29.25	87.13
Sampling size 7 → 9	0.303M	7.96ms	29.23	87.43
Sampling size 7 → 11	0.304M	8.14ms	29.25	87.13
DS → Symmetrical $DS_{7\times7}$	0.300M	7.96ms	29.29	87.13

Table 3. Effectiveness of the G-buffers.

	#Params	#Latency	PSNR	VMAF
RDG-s	**0.303M**	**7.93ms**	**29.37**	**87.22**
w/o All G-buffers	0.295M	6.89ms	28.90	85.91
w/o N^t	0.301M	7.84ms	29.27	87.06
w/o B^t	0.301M	7.89ms	29.11	86.86
w/o $M^{t-1\to t}$	0.303M	7.19ms	29.00	86.67
w/o D^t	0.303M	7.86ms	29.33	87.01

ods [8, 28, 48, 58, 61]. SAFMN [48] and BasicVSR [8] lack G-buffer guidance and tend to generate blurred lines and characters. Additionally, NSRD [28] and FuseSR [61] show unpleasant re-modulation artifacts on the edges, due to the misalignment between the reconstructed irradiance map and the high-resolution BRDF. Our proposed method exploits auxiliary G-buffer guidance to produce more accurate results, yielding sharper lines and clearer characters.

5. Analysis and Discussions

In this section, we conduct extensive ablation studies to analyze and evaluate the effect of each component in our proposed method. We implement all ablation experiments based on the RDG-s model and measure average PSNR and VMAF on all test datasets for comparisons.

Effectiveness of the DFM. The encoder utilizes the DFM to capture the spatial structural information. To demonstrate its effectiveness, we replace the DFM modules with commonly-used residual blocks for comparison with the RDG-s. Table 2 shows that a decrease of 0.2dB in PSNR and a decrease of 0.14 in VMAF.

In addition, as the DFM develops the deformable sampling (DS) to encode non-local spatial information, we conduct ablation studies to demonstrate its advantages. Replacing the DS with a combination of a pooling operation and a 3×3 depth-wise convolution results in almost unchanged latency, but leads to a 0.16dB drop in PSNR and a 0.12 drop in VMAF. As for the large kernel 15×15 depth-wise convolution, it increases the parameters and latency of the model and obtains an average PSNR of 29.25dB and an average VMAF of 87.13 on the test datasets.

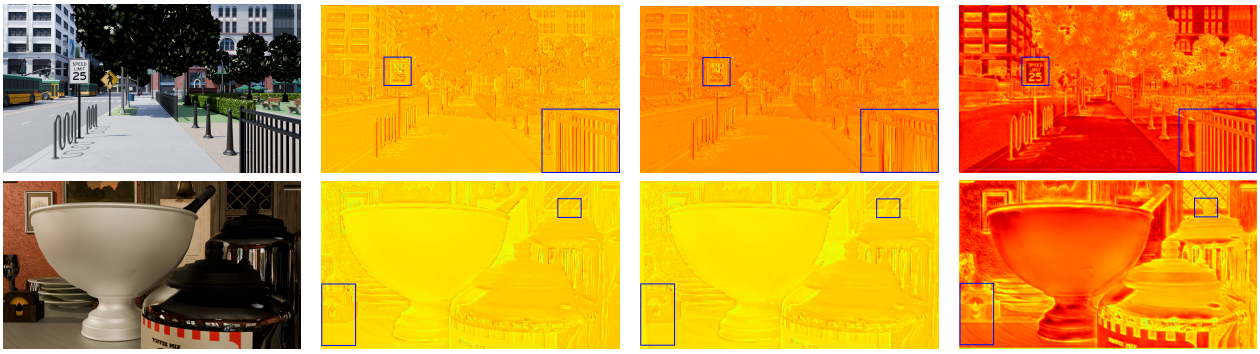
Table 2 demonstrates the impact of different sampling sizes in DFM. After the sampling size is greater than 7, increasing the sampling range will make the PSNR or VMAF decrease while increasing the latency. The sampling size of 7 can achieve a better balance between latency and reconstruction performance. Moreover, we replace the asymmetric sampling with a symmetric $DS_{7\times7}(\cdot)$. As mentioned in Section 3.2, symmetric sampling with a small channel number leads to incomplete feature extraction, resulting in a PSNR drop of 0.08dB and a VMAF drop of 0.09.

Effectiveness of the G-buffers. The RDG employs G-buffer components to facilitate the spatial and temporal feature exploration. To demonstrate the effectiveness and necessity of incorporating the G-buffer, we first conduct the experiment without any G-buffer, resulting in a significant deterioration in both image quality and temporal consistency. Specifically, the absence of the G-buffer leads to a PSNR decrease of 0.47dB and a VMAF decrease of 1.31.

Subsequently, we study the effect of specific components $\{N^t, B^t, D^t, M^{t-1\to t}\}$ of the G-buffer for reconstruction. The HFB captures high-frequency texture features from N^t and B^t to enhance the details of the generated results. Removing the N^t and B^t inputs, respectively, causes the absence of high-frequency guidance, resulting in a loss of detail in image reconstruction and a corresponding decrease in PSNR by 0.1dB and 0.26dB, respectively. The CTR module uses D^t and $M^{t-1\to t}$ to aggregate the previous and current frames and capture the spatial-temporal consistency. Without $M^{t-1\to t}$, directly merging the unaligned historical frame with the current frame leads to time flickering and artifacts. This results in a PSNR decrease of 0.37dB and a VMAF decrease of 0.55. And reducing the D^t inputs in the CTR causes the VMAF to decrease by 0.21 while the PSNR remains unchanged, suggesting that D^t contributes to the stabilization of the temporal consistency.

Effectiveness of the decoupled G-buffer guidance. The above analysis demonstrates the effectiveness of incorporating the G-buffer for real-time rendering, one may wonder whether directly using G-buffer components generates better results or not. To answer this question, we replace the decoupled G-buffer guidance with the previous way of concatenating the G-buffer [28, 58, 61], and then train this method using the same settings to compare our RDG-s. Table 4 shows that directly concatenating the G-buffer components does not generate favorable results. We further train this concatenated input using an NSRR-like UNet model and compare it to our RDG-s, and the results are significantly decreased in PSNR and VMAF values, suggesting that decoupling the G-buffer by our method yields better features for real-time rendering.

Next, we conduct various ablation experiments to illustrate the effectiveness of utilizing the decoupled G-buffer. The HFB layer employs the cross-gating mechanism to fuse $\hat{F}_e^t$ and I^t, generating detail-boosted feature maps. To verify the effectiveness of this approach, we replace the HFBs with residual blocks or SFT layers [52]. Table 4 shows that residual blocks cannot transfer high-frequency detail textures from normal and BRDF, with a drop of 0.32dB in PSNR and 0.12 in VMAF. Using the SFT layers, the PSNR



(a) Reference (b) The hidden feature H^{t-1} (c) The warped feature $\hat{H}^{t-1}$ (d) The output feature F_o^t

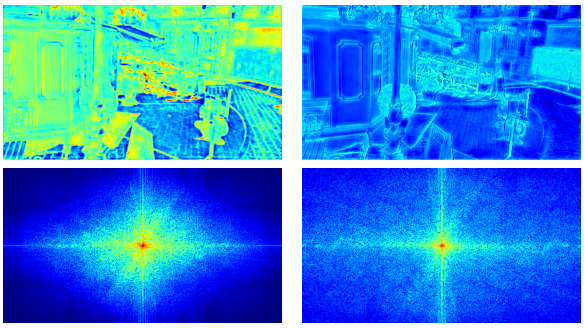
Figure 4. **Visualization of features H^{t-1}, $\hat{H}^{t-1}$ and F_o^t from CTR block**. The warp operation introduces unwanted errors, which can be efficiently corrected by the proposed CTR block and thus improves the temporal consistency.

Table 4. Effectiveness of the decoupled G-buffer guidance.

	#Params	#Latency	PSNR	VMAF
RDG-s	**0.303M**	**7.93ms**	**29.37**	**87.22**
RDG-s w/ G-buffer Concate	0.296M	7.67ms	29.29	86.68
Unet w/ G-buffer Concate	0.343M	4.39ms	28.80	83.02
HFB $\rightarrow$ ResBlock	0.300M	7.71ms	29.05	87.10
HFB $\rightarrow$ SFT Layer	0.311M	7.94ms	29.29	87.18
CTR $\rightarrow$ ResBlock	0.300M	5.31ms	29.23	85.27
CTR $\rightarrow$ ConvGRU	0.462M	5.78ms	29.34	86.39
PCA $\rightarrow$ Separable SA	0.294M	5.74 ms	29.15	85.09

values and VMAF drop by 0.08dB and 0.04, respectively. Figure 5 shows the power spectral density (PSD) maps of F_e^t and F_b^t from HFB. The input feature F_e^t contains more information about the spatial structure, and the energy is more concentrated in the low-frequency center. Whereas the output feature F_b^t enhances high-frequency information by fusing the detailed texture of the G-buffer through the HFB, the energy points of its PSD are spread in the high-frequency space. It further illustrates the effectiveness of HFB in boosting high-frequency details.

The CTR layer can efficiently propagate useful information from the previous frame and avoid errors accumulation for better reconstruction. To verify its effectiveness, we first present the visualization of the features from the CTR block. Figure 4 intuitively shows that our method is able to correct the unreliable regions (box labeled) caused by feature warping. We then provide quantitative results to objectively prove its validity. Replacing the CTR layer with residual blocks and concatenating the warped features $\hat{H}^{t-1}$ with the current features as input. Table 4 shows that using residual blocks makes the PSNR and VMAF drop to 29.23dB and 85.27, respectively. Replacing the CTR with ConvGRU [28, 32], it provides faster inference, but has a higher parameter count and is 0.83 lower on the VMAF metric. We further use separable self-attention [44] to replace the proposed PCA operation and find that the PSNR is reduced by 0.22dB and the VMAF is reduced by 2.13. These results show that our PCA effectively preserves the useful information of the current frame while avoiding errors accumulation during temporal propagation.



(a) Input feature F_e^t (b) Output feature F_b^t

Figure 5. **The power spectral density (PSD) visualizations of input feature F_e^t and output feature F_b^t from HFB**. We perform a periodic shift of the spectrum map such that the low-frequency component is moved to the center.

6. Conclusion

We have presented an efficient video super-resolution method for real-time rendering. We develop a lightweight dynamic feature modulator (DFM) to selectively encode contextual information for acquiring accurate structural representations. We further incorporate auxiliary G-buffer information to guide the decoder to generate detail-rich, temporally stable results. To boost high-frequency details, we adopt a high-frequency feature booster (HFB) to adaptively transfer the high-frequency information from the normal and BRDF components. To avoid errors introduced by the warping operation, we design a cross-frame temporal refiner (CTR) with depth map constraints to accurately aggregate the previous and current frames. By training the proposed approach in an end-to-end manner, we show that our methods perform favorably against the state-of-the-art models in terms of accuracy and efficiency.

Acknowledgments. This work has been supported in part by the National Natural Science Foundation of China (Nos. 62272233, U22B2049, and 62332010).

References

[1] Dlss: Three things you need to know. https://developer.nvidia.com/blog/dlss-three-things-you-need-to-know/, 2019. 1, 3

[2] Blender. https://www.blender.org/, 2023. 5

[3] Amd fidelityfx super resolution. https://www.amd.com/en/products/graphics/technologies/fidelityfx/super-resolution.html, 2023. 1

[4] Intel® arc™- xe super sampling. https://www.intel.com/content/www/us/en/products/docs/discrete-gpus/arc/technology/xess.html, 2023. 1, 3

[5] Kurt Akeley. Reality engine graphics. In *SIGGRAPH*, 1993. 2

[6] Jose Caballero, Christian Ledig, Andrew P. Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*, 2017. 2

[7] Cgtrader. Nyc block set, 2019. 5

[8] Kelvin C. K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7

[9] Kelvin C. K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 2022. 1, 2, 4, 5, 6

[10] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. In *ICLR*, 2022. 4

[11] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *ICCV*, 2021. 5

[12] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 4

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 4

[14] Peng Dai, Xin Yu, Lan Ma, Baoheng Zhang, Jia Li, Wenbo Li, Jiajun Shen, and Xiaojuan Qi. Video demoireing with relation-based temporal consistency. In *CVPR*, 2022. 5

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

[16] Jie Guo, Xihao Fu, Liqiang Lin, Hengjun Ma, Yanwen Guo, Shiqiu Liu, and Ling-Qi Yan. Extranet: Real-time extrapolated rendering for low-latency temporal supersampling. *TOG*, 2021. 2

[17] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *CVM*, 2023. 4

[18] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019. 2

[19] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units. *arXiv preprint arXiv:1606.08415*, 2016. 4

[20] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *NeurIPS*, 2015. 2

[21] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, 2019. 1, 2

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

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

[24] Jorge Jimenez, Jose I. Echevarria, Tiago Sousa, and Diego Gutierrez. SMAA: enhanced subpixel morphological antialiasing. *CGF*, 2012. 3

[25] Brian Karis. High quality temporal anti-aliasing. *TOG*, 2014. 3

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[27] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *TPAMI*, 2019. 5

[28] Jia Li, Ziling Chen, Xiaolong Wu, Lu Wang, Beibei Wang, and Lei Zhang. Neural super-resolution for real-time rendering with radiance demodulation. In *CVPR*, 2024. 2, 3, 4, 5, 6, 7, 8

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

[30] Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Jinjin Gu, Yu Qiao, and Chao Dong. Blueprint separable residual network for efficient image super-resolution. In *CVPR Workshops*, 2022. 2

[31] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention. In *NeurIPS*, 2022. 1, 2

[32] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *WACV*, 2022. 8

[33] Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian. Learning trajectory-aware transformer for video super-resolution. In *CVPR*, 2022. 2

[34] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *ECCV Workshops*, 2020. 1, 2

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[36] Timothy Lottes. FXAA. Technical report, NVIDIA, 2009. 2

[37] Amazon Lumberyard. Amazon lumberyard bistro, open research content archive (orca), 2017. 5

[38] Mihoyo. Tenshukaku, 2021. 5

[39] Kate Anderson Nicholas Hull and Nir Benty. Nvidia emerald square, open research content archive (orca), 2017. 5

[40] Zhongwei Qiu, Huan Yang, Jianlong Fu, Daochang Liu, Chang Xu, and Dongmei Fu. Learning degradation-robust spatiotemporal frequency-transformer for video super-resolution. *TPAMI*, 2023. 1, 2

[41] Reza Rassool. Vmaf reproducibility: Validating a perceptual practical video quality metric. In *BMSB*, 2017. 6

[42] Alexander Reshetov. Morphological antialiasing. In *HPG*, 2009. 2

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 8

[45] Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. In *NeurIPS*, 2022. 2

[46] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 4

[47] Long Sun, Jinshan Pan, and Jinhui Tang. ShuffleMixer: An efficient convnet for image super-resolution. In *NeurIPS*, 2022. 1, 2, 5

[48] Long Sun, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. Spatially-adaptive feature modulation for efficient image super-resolution. In *ICCV*, 2023. 1, 2, 3, 4, 5, 6, 7

[49] Blend Swap. Happy little pond, 2021. 5

[50] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. TDAN: temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020. 2

[51] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. MAXIM: Multi-axis MLP for image processing. In *CVPR*, 2022. 4

[52] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 7

[53] Xintao Wang, Kelvin C. K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: video restoration with enhanced deformable convolutional networks. In *CVPR Workshops*, 2019. 1, 2

[54] Yan Wang, Yusen Li, Gang Wang, and Xiaoguang Liu. Multi-scale attention network for single image super-resolution. In *CVPR Workshops*, 2024. 4

[55] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6

[56] Mike Winkelmann. Zero-day, open research content archive (orca), 2019. 5

[57] Songyin Wu, Sungye Kim, Zheng Zeng, Deepak Vembar, Sangeeta Jha, Anton Kaplanyan, and Ling-Qi Yan. Extrass: A framework for joint spatial super sampling and frame extrapolation. In *SIGGRAPH Asia*, 2023. 2

[58] Lei Xiao, Salah Nouri, Matthew Chapman, Alexander Fix, Douglas Lanman, and Anton Kaplanyan. Neural supersampling for real-time rendering. *TOG*, 2020. 2, 3, 5, 6, 7

[59] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 2, 5

[60] Mingjun Zheng, Long Sun, Jiangxin Dong, and Jinshan Pan. Smfanet: A lightweight self-modulation feature aggregation network for efficient image super-resolution. In *ECCV*, 2024. 1, 2, 4, 5, 6

[61] Zhihua Zhong, Jingsen Zhu, Yuxin Dai, Chuankun Zheng, Guanlin Chen, Yuchi Huo, Hujun Bao, and Rui Wang. Fusesr: Super resolution for real-time rendering through efficient multi-resolution fusion. In *SIGGRAPH Asia*, 2023. 2, 3, 4, 5, 6, 7

[62] Lin Zhou, Haoming Cai, Jinjin Gu, Zheyuan Li, Yingqi Liu, Xiangyu Chen, Yu Qiao, and Chao Dong. Efficient image super-resolution using vast-receptive-field attention. In *ECCV Workshops*, 2022. 4