Mitigating Artifacts in Real-World Video Super-resolution Models

Liangbin Xie[*1,2,3], **Xintao Wang**[3], **Shuwei Shi**[1,4], **Jinjin Gu**[5,6], **Chao Dong**[†1,6], **Ying Shan**[3]

[1] The Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology,
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2] University of Macau
[3] ARC Lab, Tencent PCG
[4] Shenzhen International Graduate School, Tsinghua University
[5] The University of Sydney
[6] Shanghai Artificial Intelligence Laboratory
{lb.xie, chao.dong}@siat.ac.cn, ssw20@mails.tsinghua.edu.cn, jinjin.gu@sydney.edu.au
{xintaowang, yingsshan}@tencent.com

Abstract

The recurrent structure is a prevalent framework for the task of video super-resolution, which models the temporal dependency between frames via hidden states. When applied to real-world scenarios with unknown and complex degradations, hidden states tend to contain unpleasant artifacts and propagate them to restored frames. In this circumstance, our analyses show that such artifacts can be largely alleviated when the hidden state is replaced with a cleaner counterpart. Based on the observations, we propose a Hidden State Attention (HSA) module to mitigate artifacts in real-world video super-resolution. Specifically, we first adopt various cheap filters to produce a hidden state pool. For example, Gaussian blur filters are for smoothing artifacts while sharpening filters are for enhancing details. To aggregate a new hidden state that contains fewer artifacts from the hidden state pool, we devise a Selective Cross Attention (SCA) module, in which the attention between input features and each hidden state is calculated. Equipped with HSA, our proposed method, namely FastRealVSR, is able to achieve $2\times$ speedup while obtaining better performance than Real-BasicVSR. Codes will be available at https://github.com/TencentARC/FastRealVSR.

Introduction

Video Super-Resolution (VSR) aims to recover high-resolution (HR) frame sequences from their low-resolution (LR) counterparts, where the utilization of complementary information across adjacent frames is the key factor to improve performance. VSR can be roughly classified into two categories – sliding-window-based methods (Caballero et al. 2017; Huang, Wang, and Wang 2017; Wang et al. 2019; Tian et al. 2020; Li et al. 2020; Isobe et al. 2020a) and recurrent methods (Chan et al. 2021, 2022a; Huang, Wang, and Wang 2015; Sajjadi, Vemulapalli, and Brown 2018; Haris, Shakhnarovich, and Ukita 2019; Xiang et al. 2020; Isobe et al. 2020a,b; Lin, Huang, and Wang 2021). Recurrent methods are the prevalent approaches since they can
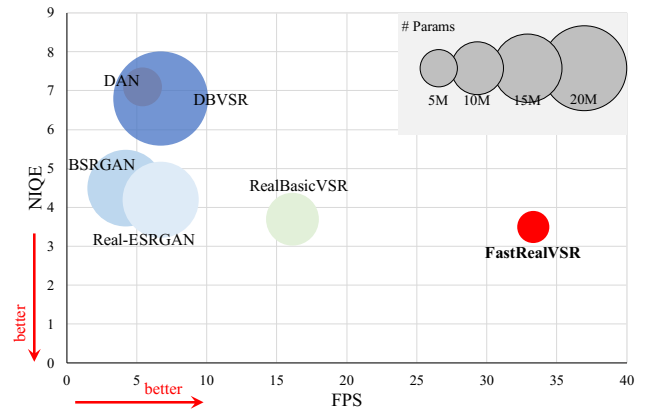

Figure 1: Our proposed FastRealVSR can ahieve the best trade-off between NIQE, speed and parameters in the real-world VideoLQ dataset.

model long-term dependencies with hidden states, and have won several championships in video restoration competitions (Yang 2021; Yang et al. 2022a).

When applied to real-world low-quality videos, existing recurrent methods (*e.g.*, BasicVSR (Chan et al. 2021)) tend to produce unpleasant artifacts, which will be amplified through propagation (Chan et al. 2022b). This problem becomes more severe when the degradations are more complicated with unknown blur, noise and compression.

Recent work Real-BasicVSR (Chan et al. 2022b) makes the first attempt to alleviate those artifacts. It adopts a separate image pre-cleaning module before the video recurrent restoration model to explicitly remove degradations for each video frame. However, while mitigating unpleasant artifacts, such a strategy can also lead to detail loss, due to the explicit denoising operation in the image level. Besides, the separate module brings extra computation cost, which slows down the inference speed and is not friendly to practical usages.

In this work, we carefully investigate the cause of the unpleasant artifacts in VSR. We draw two observations. 1) Once the output of the previous frame produces artifacts, the

Previous hidden state is aggregated when restoring current frame.

Previous hidden state is **discarded** when restoring current frame.

Previous hidden state is aggregated when restoring current frame.

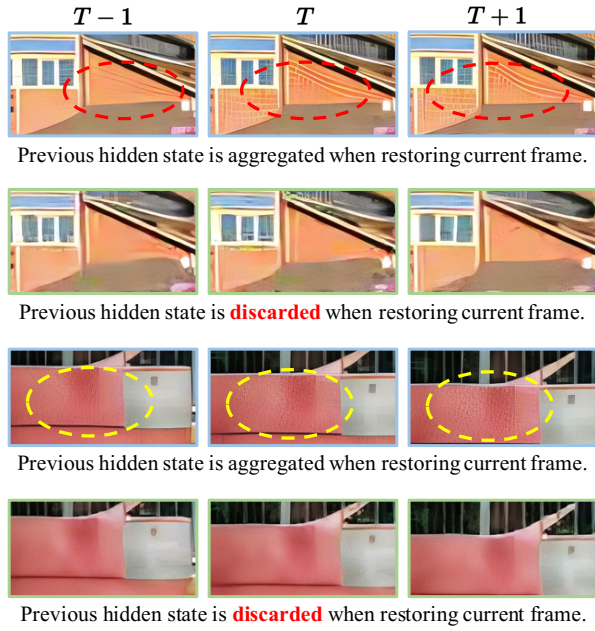Previous hidden state is **discarded** when restoring current frame.

Figure 2: Negative impact of hidden state in real-world scenario. The hidden state could contain artifacts that appears in the previous frames, and further propagate to consequent frames. The regions marked with the ellipse dotted box are the artifacts. (Zoom-in for best view)

hidden state will also contain such artifacts and further propagate to consequent frames. When we discard the hidden state and only use the current input frame, those artifacts disappear, but at the cost of detail loss. 2) When we replace the "problematic" hidden state with a cleaner one (*e.g.*, the corresponding one from the MSE model[1] or a Gaussian-blurred one), the artifacts are largely mitigated and do not further propagate to consequent frames.

Such observations motivate us to directly manipulate the hidden state instead of introducing extra cleaning modules. We propose a Hidden State Attention (HSA) module to mitigate artifacts in real-world video super-resolution. Specifically, we first generate a hidden state pool, where each new hidden state is produced by filtering the original hidden state with different filters, including Gaussian blur filters, box blur filters, sharpening filters, *etc*. Those operations will either blur hidden state to alleviate artifacts or sharpen edges to enhance details. We choose those classical filters as they only bring a little computation cost. After that, selective cross attention will calculate the similarity between the features of the current frame and each hidden state in the pool. As "problematic" hidden state usually has a large deviation from input features, such a design can effectively mitigate artifacts. A cleaned hidden state is then aggregated from the hidden state pool based on the calculated attention. As a result, this new hidden state not only contains fewer artifacts, but also maintains the details.

---

[1]Training with MSE loss tends to produce over-smooth results and thus the hidden state is also cleaner and more smooth.
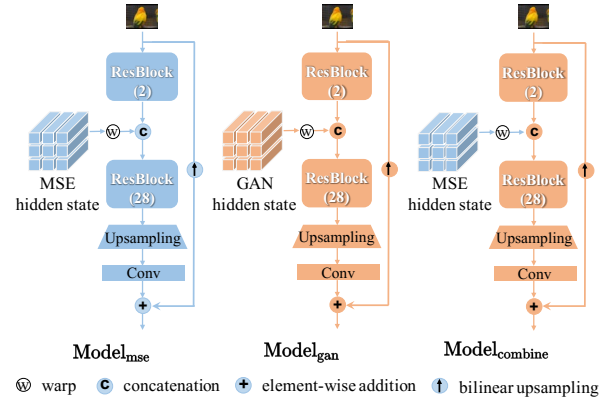

Figure 3: The diagram of general unidirectional recurrent network. $\text{Model}_{\text{combine}}$ is derived by replacing the GAN hidden state in $\text{Model}_{\text{gan}}$ with the MSE hidden state in $\text{Model}_{\text{mse}}$. We use a single color to represent a complete pre-trained model. (Zoom-in for best view)

To make our algorithm more practical in the real world, we also take inference speed into consideration. We adopt a compact unidirectional recurrent framework as the backbone of our method. Together with the efficient Hidden State Attention (HSA) module, our proposed method, namely FastRealVSR, achieves $2\times$ speed-up over Real-BasicVSR, while keeping comparable performance.

Our contributions can be summarized as follows. **1)** We analyze the hidden state in recurrent VSR methods and show that the hidden state is the main reason for unpleasant artifacts. **2)** An effective hidden state attention with cheap hidden states and selective cross attention is proposed to mitigate artifacts in real-world VSR models. **3)** FastRealVSR is capable of achieving a better speed-performance trade-off, surpassing previous methods.

Related Work

Video Super Resolution. Most existing VSR methods can be divided into two categories: sliding-window based methods (Caballero et al. 2017; Huang, Wang, and Wang 2017; Wang et al. 2019; Tian et al. 2020; Li et al. 2020) and recurrent methods (Chan et al. 2021, 2022a; Huang, Wang, and Wang 2015; Xiang et al. 2020; Isobe et al. 2020a,b). Sliding-window methods often take several adjacent frames as input and predict the center frame. In contrast to sliding-window based methods, recurrent methods adopt recurrent neural network (Mikolov et al. 2010) to process the input frame and generate the output progressively. Some recent methods (Chan et al. 2021, 2022a) have achieved excellent performance on synthetic degradations (*e.g.*, bicubic). Recently, several transformer-based (Cao et al. 2021; Liang et al. 2022) methods are proposed. However, when these methods (Chan et al. 2021) are applied to real-world scenarios, their performance deteriorates severely and obvious artifacts will appear since the degradations in real-world scene are unknown and complex.

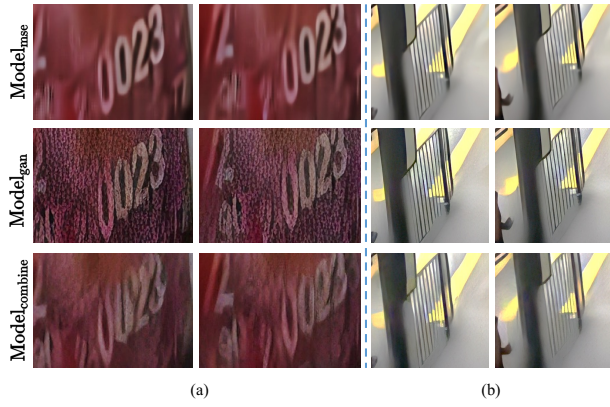**Real-World Video Super-Resolution.** Recent work Re-

Figure 4: Qualitative comparison between $\text{Model}_{\text{mse}}$, $\text{Model}_{\text{gan}}$ and $\text{Model}_{\text{combine}}$. As shown on the left side of the figure, after replacing GAN hidden state with MSE hidden state, the artifacts in the wall are largely mitigated. As shown on the right side of the figure, the output of $\text{Model}_{\text{combine}}$ also loses many necessary details in the railing. (Zoom-in for best view)
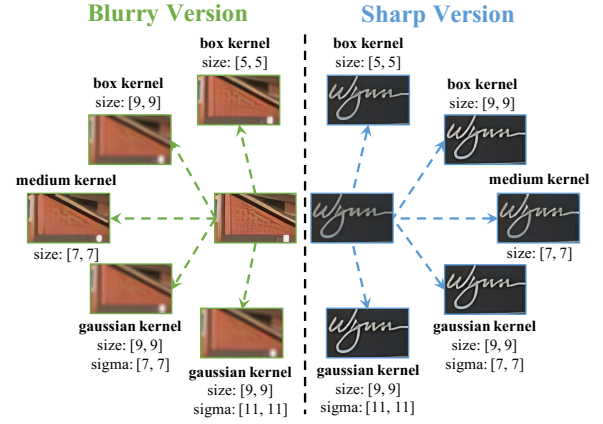

Figure 5: Effects of different kernels. For regions with artifacts, blur filters can mitigate artifacts. For regions with details, sharpening filters can further sharpen details.

alBasicVSR (Chan et al. 2022b) is proposed to investigate tradeoffs in real-world VSR. Similar to Real-ESRGAN (Wang et al. 2021), a second-order degradation model is adopted to simulate the degradations in the real-world. To alleviate the artifacts that are amplified in recurrent methods (*e.g.*BasicVSR), it uses a pre-cleaning module and a cleaning loss to "clean" the input sequence, which is then fed into a general VSR network. In this paper, to mitigate the artifacts, we do not introduce any cleaning modules. Instead, we manipulate the hidden state directly to mitigate the artifacts. Compared to RealBasicVSR, our proposed FastRealVSR achieves a better speed-performance trade-off.

Hidden State Manipulation. Hidden state is the key component in recurrent methods that adopted in video restoration task (Nah, Son, and Lee 2019; Zhong et al. 2020). To handle the misalignment between the hidden state and the referred current feature, some works (Nah, Son, and Lee 2019; Zhou et al. 2019) in video deblurring area are proposed to adapt the hidden states to the current frame by referring to the current feature. Different from them, we aggregate a new hidden state from a hidden state pool and aim to alleviate the mitigate contained in "problematic" hidden state.

Methodology

Analysis of Hidden State

Observation 1: *Hidden state in recurrent methods propagates the artifacts.*

In a unidirectional recurrent method, information in the hidden state serves two purposes. The first is to reconstruct the current frame with a reconstruction module, and the second is to propagate the previous information to the next frame. Therefore, once the output of the previous frame produces artifacts, the hidden state also contains such artifacts and further propagates to consequent frames. As shown in the 1st and 3rd rows in Fig. 2, the contents of ellipse dotted

box in T_{t-1}, T_t retain and amplify the artifacts that appears in T_{t-2}. If the hidden state is not used (setting to 0) and only the information of input image is used, the artifacts will be clearly alleviated. This phenomenon can be observed in the 2nd and 4th rows in Fig. 2. Directly discarding the hidden state, however, also lead to detail loss and unsharp edges (2nd and 4th rows in Fig. 2).

In a word, the hidden state not only enhances details but also propagates the artifacts, which is consistent with the observations in RealBasicVSR (Chan et al. 2022b).

Observation 2: *Replacing the hidden state in the GAN-based model with its corresponding hidden state in the MSE-based model can mitigate the artifacts.*

When the hidden state contains artifacts, although directly discarding the hidden state can alleviate the artifacts, it also leads to obvious details loss in other regions. Therefore, it is better to find a cleaner variant of the "problematic" hidden state which contains fewer artifacts and maintains its details. Typically, a GAN-based model is fine-tuned from an MSE-based model with a combination of MSE loss, perceptual loss (Johnson, Alahi, and Fei-Fei 2016) and GAN loss (Goodfellow et al. 2014). As shown in the 1st row of Fig. 4, the model trained with MSE loss tends to produce over-smooth results, thus the hidden state in the MSE-based model is cleaner and smooth. Consequently, we wonder whether the artifacts of restored outputs can be mitigated if we replace the 'problematic' hidden state with the corresponding one in the MSE model. For better clarification, the hidden state in the MSE-based model and the hidden state in GAN-based model are referred to as MSE hidden state and GAN hidden state, respectively.

Fig. 3 illustrates the simplified diagram of a unidirectional recurrent network that uses both hidden state and input frame as inputs. In Fig. 3, we use a single color to represent a complete pre-trained inference model. To replace the GAN hidden state with MSE hidden state, firstly, we need to use $\text{Model}_{\text{mse}}$ (pre-trained MSE model) to infer the input sequence and store all generated hidden states.
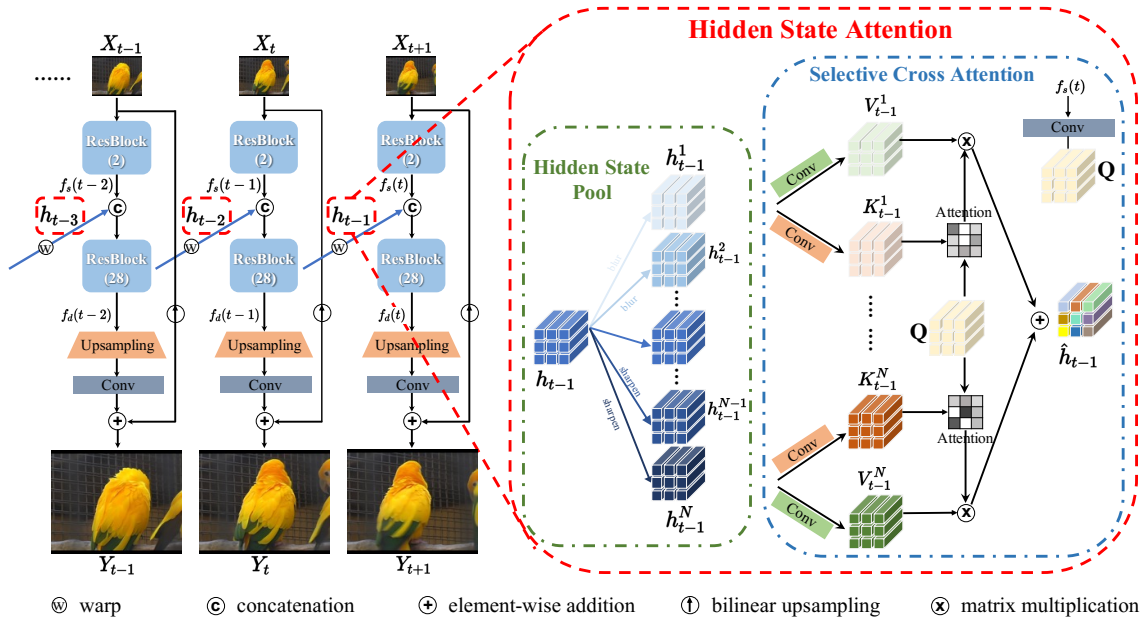
Figure 6: Framework overview of FastRealVSR. The left flowchart is a general unidirectional recurrent network. $f_s(t)$ and $f_d(t)$ represent the shallow and sharp feature of X_t, respectively. h_{t-1} is the hidden state that is propagated to restore X_t. By adopting hidden state attention (HSA), h_{t-1} is transformed into $\hat{h}_{t-1}$. HSA is composed of a hidden state pool and a selective cross attention module, as shown on the right side of the figure.

Then, we will infer the input sequence with $\text{Model}_{\text{combine}}$. $\text{Model}_{\text{combine}}$ is modified from $\text{Model}_{\text{gan}}$ and the difference is that the GAN hidden state is replaced with the previously stored MSE hidden state. The qualitative comparison between these three models is shown in Fig. 4. We can observe that **1)** from the comparison of the 2nd and 3rd rows (Fig. 4 (a)), the artifacts in $\text{Model}_{\text{gan}}$ are reduced in $\text{Model}_{\text{combine}}$. **2)** directly replacing the GAN hidden state with the MSE hidden state could sometimes lose the necessary details (Fig. 4 (b)), compared against $\text{Model}_{\text{gan}}$.

Although replacing the GAN hidden state with the MSE hidden state can mitigate artifacts, it is unpractical with two shortcomings. **1)** To obtain final results, it takes the double time since we need to run $\text{Model}_{\text{mse}}$ and $\text{Model}_{\text{combine}}$ in sequence. **2)** The output of $\text{Model}_{\text{combine}}$ also loses many necessary details compared against $\text{Model}_{\text{gan}}$. Therefore, it is preferable to directly manipulate the GAN hidden state, and construct a new hidden state that not only contains the smooth counterpart of the artifact region in the GAN hidden state, but also maintains the details of other regions.

Framework

Motivated by the above observations, we propose a hidden state attention (HSA) module to directly manipulate the hidden state and mitigate artifacts. An overview is illustrated in Fig. 6.
Formulation. We first briefly introduce the pipeline of the general unidirectional recurrent VSR method. Let X_t be the t-th image of the input sequence and Y_t be the t-th frame of the restored sequence. Several residual blocks RB_1 are used to extract the shallow feature of X_t:

$$f_s(t) = \text{RB}_1(X_t). \tag{1}$$

The hidden state from previous frame is first warped with the estimated optical flow. The warped hidden state is denoted as h_{t-1}. We use a pre-trained SPyNet (Ranjan and Black 2017) to generate the optical flow. Then h_{t-1} is concatenated with $f_s(t)$ and input to another set of residual blocks RB_2 to get the deep feature $f_d(t)$:

$$f_d(t) = \text{RB}_2(\text{Concat}(h_{t-1}, f_s(t))), \tag{2}$$

where $h_{t-1} \in \mathbb{R}^{H \times W \times C}$. $f_d(t)$ is then passed to an upsampling block UP, and added with the bilinear upsampling of X_t to generate Y_t:

$$Y_t = \text{UP}(f_d(t)) + \text{Bilinear}(X_t). \tag{3}$$

For our method, we replace h_{t-1} with $\hat{h}_{t-1}$ with HSA:

$$\hat{h}_{t-1} = \text{HSA}(h_{t-1}). \tag{4}$$

where HSA is composed of a hidden state pool module and a selective cross attention module. The specific details of these two modules are described below.
Hidden State Pool. Hidden state pool is a container that gathers the blurry and sharp versions of h_{t-1}. The blurry version is obtained by a blurring operation, while the sharp version is obtained by the simplified sharpening algorithm (*i.e.* unsharp masking (USM)). Specially, the sharp version is obtained by first subtracting the blurry version from the input, and then adding the residual to the input.

The above design is inspired by feature denoising (Xie et al. 2019) that is adopted to clean feature maps for improving the adversarial robustness of convolutional networks. The effects of different blur operations are shown in Fig. 5. Here, we use images to better illustrate the effects of blurry and sharpening filters. For the region with artifacts, we prefer its blurry version, while for the region with correct details, we prefer its sharp version.

In practical implementation, the operation of generating different versions of h_{t-1} can be easily achieved by a convolution layer with a fixed kernel. Such computational operations are very cheap, and the computational cost is negligible. The process of generating the i-th blurry version of h_{t-1} and the j-th sharp version of h_{t-1} are formulated as:

$$h_{t-1}^i = h_{t-1} \otimes \mathbf{k}^i, \qquad (5)$$

$$h_{t-1}^j = h_{t-1} + (h_{t-1} - h_{t-1} \otimes \mathbf{k}^j), \qquad (6)$$

where $\otimes$ denotes the convolving operation and $\mathbf{k}^i$, $\mathbf{k}^j$ are the blur kernels. The degree of blurring and sharpening can be controlled by adjusting the kernel size and variance (only specifically for the Gaussian filter). In our experiment, two mean filters and one Gaussian filter are adopted to generate blurry versions. We use two sharpening filters to generate the sharp versions. Detailed configurations of these filters are provided in the appendix.

Selective Cross Attention. After obtaining the hidden state pool $h_{t-1}^i \in \mathbb{R}^{H \times W \times C}$ ($1 \le i \le N$) (N denotes the number of hidden state included in the hidden state pool), it is vital to aggregate a new hidden state that contains fewer artifacts. From the 2nd and 4th rows of Fig. 2, we can observe that the artifacts in the output are largely mitigated if only the information of the input frame is utilized, indicating that the input feature contains fewer artifacts. Therefore, for the region with artifacts, the "problematic" hidden state has a large deviation from input features. Inspired by this, we propose a selective cross attention (SCA) module to aggregate a new hidden state $\hat{h}_{t-1}$ based on the attention between the hidden state pool $h_{t-1}^i \in \mathbb{R}^{H \times W \times C}$ ($1 \le i \le N$) and feature of input frame.

As shown in Fig. 6, we first process $f_s(t)$ with a convolutional layer H_{conv}, and get the query $Q \in \mathbb{R}^{H \times W \times C}$:

$$Q = H_{\text{conv}}(f_s(t)). \qquad (7)$$

For all h_{t-1}^i ($1 \le i \le N$), we adopt two 3×3 convolutional layers to generate the key K_{t-1}^i ($1 \le i \le N$) and value V_{t-1}^i ($1 \le i \le N$), respectively. Then we calculate the similarity between each K_{t-1}^i and Q by the attention mechanism (Vaswani et al. 2017; Liu et al. 2021). Based on the calculated attention maps, we aggregate $\hat{h}_{t-1}^i$ from V_{t-1}^i ($1 \le i \le N$). The attention matrix is computed as

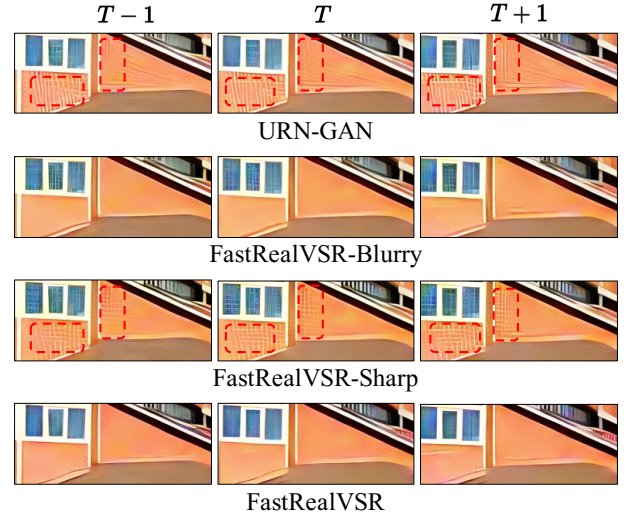$$\text{SoftMax}([Q(K^1)^T; ...; Q(K^N)^T]) \otimes (V^1; ...; V^N). \qquad (8)$$

Figure 7: Results of URN-GAN, FastRealVSR-Blurry, FastRealVSR-Sharp and FastRealVSR. The outputs of URN-GAN and FastRealVSR-Sharp contain unpleasant artifacts. (Zoom-in for best view)

Experiments

Setup

Training Settings. We train our FastRealVSR on the REDS (Nah et al. 2019) dataset. The following degradation model is adopted to synthesize training data:

$$\mathbf{X} = [(\mathbf{Y} \circledast \mathbf{k}_\sigma + \mathbf{n}_\delta) \downarrow_r]_{\text{FFMPEG}}, \qquad (9)$$

where $\mathbf{X}$ and $\mathbf{Y}$ are paired low-resolution and high-resolution sequences. The $\mathbf{k}$, $\mathbf{n}$, and r are blur kernel, additive noise, and downsampling factor, respectively. We use a constant rate factor (*crf*) to control the degree of FFMPEG compression. *crf* uses a specific quality by adjusting the bitrates automatically. The sampling range of σ, δ, and *crf* are $\{0.2 : 3\}$, $\{1 : 5\}$, and $\{18 : 35\}$, respectively.

Following RealBasicVSR, we load 15 frames as a sequence and flip them temporally in each iteration. The patch size of input LR frames is 64×64. The pre-trained SPyNet (Ranjan and Black 2017) is used as the flow network and its weights are fixed in training. We employ Adam optimizer (Kingma and Ba 2014). The training process is divided into two stages.

In the first stage, we adopt the Unidirectional Recurrent Network (URN) shown in Fig. 6, and train it for 300K iterations with the L_1 loss. The batch size and learning rate are set to 16 and 10^{-4}. In the second stage, we equip URN with the proposed HSA module to get the network FastRealVSR. We employ the pre-trained MSE model for initialization. Then we train FastRealVSR for 70K iterations with a combination of L_1 loss, perceptual loss (Johnson, Alahi, and Fei-Fei 2016) and GAN loss (Goodfellow et al. 2014), whose loss weights are set to $1, 1, 5 \times 10^{-2}$, respectively. The specific implementation details, network configuration and the evaluation metrics can be found in the appendix.
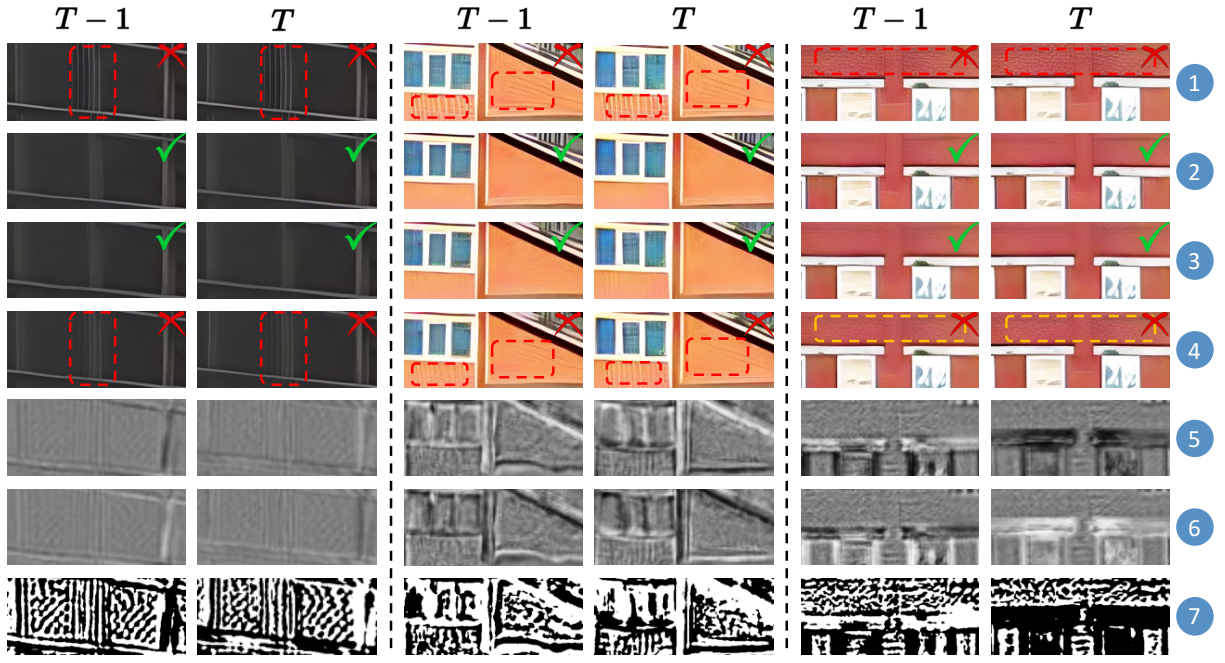
Figure 8: Analysis of hidden state pool and SCA. The outputs of URN-GAN contain obvious artifacts (①). The artifacts produced in URN-GAN are largely alleviated in FastRealVSR (②). When all hidden states in the hidden state pool are replaced with a single blurry hidden state, the artifacts are mitigated (③). While the artifacts are retained when all hidden states are replaced with a single sharp hidden state (④). The summed blurry/sharp attention maps are visualized in ⑤ and ⑥, respectively. The binary map (⑦) indicates the region where the value of ⑤ is larger than ⑥. The regions marked with ellipse dotted boxes are artifacts. (Zoom-in for best view)

Ablation Studies

Hidden State Pool. We use the GAN version of URN (URN-GAN) as the baseline model. Then we train FastRealVSR, FastRealVSR-Blurry and FastRealVSR-Sharp with the same settings. The only difference among them is the representation type of the hidden states in the hidden state pool. To be specific, FastRealVSR-Blur contains only blurry variants of hidden states, while FastRealVSR-Sharp contains only sharp ones. The quantitative comparison is listed in Tab. 1. The performance of FastRealVSR is significantly better than FastRealVSR-Blurry and FastRealVSR-Sharp, indicating that the information in the blurry and sharp representations of hidden states is complementary to each other. Fig. 7 shows the restored results of these four models. It can be observed that FastRealVSR and FastRealVSR-Blurry both produce outputs with fewer artifacts, whereas the outputs of URN-GAN and FastRealVSR-Blurry contain obvious artifacts. Besides, the edges of FastRealVSR are sharper than FastRealVSR-Blurry.

Selective Cross Attention. In SCA, each hidden state interacts with input features by attention. If the input feature is not used to guide the aggregation of hidden states and only a convolution layer is adopted to aggregate these hidden states, a performance drop is observed in Tab. 2. It indicates that the guidance of input features is necessary when aggregating a new hidden state from a hidden state pool.

Network	NIQE↓
URN-GAN	3.8435
FastRealVSR-Blurry	3.8544
FastRealVSR-Sharp	3.8210
FastRealVSR	3.7658

Table 1: Ablation studies of different representation types in hidden state pool. FastRealVSR achieves best performance.

Variant	w/o SCA	FastRealVSR (w/ SCA)
NIQE↓	3.8446	3.7658

Table 2: Ablation study of SCA. With SCA, FastRealVSR achieves better performance.

Analysis of Hidden State Attention

The proposed HSA module aims to alleviate artifacts in the hidden state. Hence, we further validate the effectiveness of HSA for the "problematic" hidden state. When the previous output contains artifacts, the hidden state generated in URN-GAN contains artifacts and is a "problematic" hidden state (see Fig. 7 and the 1st row in Fig. 8). To get the "problematic" hidden state, we run URN-GAN and store all generated hidden states. We use h_t ($1 \leq t \leq L-1$) to refer to the "problematic" hidden state that produced by URN-GAN during the process of restoring low-resolution frame X_t. To test whether HSA in RealFastVSR has the capacity

Figure 9: Qualitative comparison on the VideoLQ dataset for $\times 4$ video SR. (Zoom-in for best view)

	Bicubic	DAN	RealSR	BSRGAN	Real-ESRGAN	RealVSR	DBVSR	RealBasicVSR	**FastRealVSR**
Params (M)	-	4.3	16.7	16.7	16.7	**2.7**	25.5	6.3	3.8
Runtime (ms)	-	185	149	149	149	1082	239	63	**30**
NIQE ↓	8.0049	7.1230	4.1482	4.2460	4.2091	8.0606	6.7866	3.7662	**3.7658**
BRISQUE ↓	54.899	51.563	30.542	30.213	32.103	54.988	50.936	29.030	**29.374**

Table 3: Quantitative comparison on VideoLQ dataset. FastRealVSR is capable of achieving a better speed-performance trade-off. Runtime is computed with an output size of 720×1280, with an NVIDIA V100 GPU. Following RealBasicVSR, metrics are computed on the Y-channel of the first, middle, and last frames of each sequence.

to mitigate artifacts, we replace the original hidden state in RealFastVSR with the "problematic" hidden state h_{t-1} from URN-GAN when restoring $\boldsymbol{X}_t$. From the 2nd row in Fig. 8, we can observe that HSA has the ability to mitigate artifacts contained in the "problematic" hidden state.

Investigation of Different Filters in Hidden State Pool.
We then investigate the respective effectiveness of blurry and sharp filters utilized in the hidden state pool. Specifically, we replace all hidden states in the hidden state pool with a single variant of hidden state, *i.e.*, the hidden state pool contains identical hidden states after same filters. Note that the number of hidden states in the hidden state pool remains the same. The results of adopting only a single blurry or a single sharp filter in hidden state pool are shown in the 3rd and 4th rows of Fig. 8, respectively. We can observe that blurry filters can mitigate artifacts but with the cost of detail loss, whereas sharp filters enhance both artifacts and details. Therefore, *the combination of blurry filters and sharp filters can make a better trade-off between enhancing details and suppressing artifacts.*

Visualization of Attention Map in SCA. We also visualize the attention map in the selective cross attention (SCA) module to understand how the information in different representations of hidden state is selected based on the input feature. For better visualization, blurry and sharp attention maps are summed together, respectively, resulting in two maps. The visualization of these two maps is shown in the 5th and 6th rows in Fig. 8. We can observe that for different versions (*i.e.*, blurry or sharp) of h_{t-1}, SCA indeed learns to select the proper information with end-to-end training. Besides, we compare these two maps and get the binary map shown in the 7th row in Fig. 8. The binary map indicates the region where the value of blurry attention map is larger than the sharp one. It can be observed that *for the regions with artifacts, SCA prefers the blurry representations.*

Comparison with State-of-the-Art Methods

We compare our FastRealVSR method with seven algorithms: DAN (Huang et al. 2020), RealSR (Ji et al. 2020), BSRGAN (Zhang et al. 2021), Real-ESRGAN (Wang et al. 2021), RealVSR (Yang et al. 2021), DBVSR (Pan et al. 2021) and RealBasicVSR (Chan et al. 2022b) on the VideoLQ (Chan et al. 2022b) dataset. VideoLQ consists of 50 real-world low-quality videos that are collected from various video hosting sites such as Flickr and YouTube.

The quantitative results on VideoLQ are shown in Tab. 3. Compared with other methods, FastRealVSR achieves a better speed-performance trade-off. In particular, compared to RealBasicVSR, FastRealVSR achieves comparable performance with $2 \times$ faster speed and smaller model size. Qualitative results are presented in Fig. 9. Compared to previous methods, FastRealVSR can effectively remove annoying artifacts and preserve sharp details.

Conclusion

The hidden state in the recurrent VSR models not only enhances details but also propagates artifacts. Our analyses show that such artifacts can be largely mitigated if the hidden state is replaced with a cleaner counterpart. Therefore, we propose the Hidden State Attention (HSA) module that is composed of a hidden state pool and a selective cross attention module. Our FastRealVSR combines the unidirectional recurrent network and the proposed HSA, achieving a better speed-performance trade-off than previous methods.

Acknowledgments

References

Blau, Y.; Mechrez, R.; Timofte, R.; Michaeli, T.; and Zelnik-Manor, L. 2018. The 2018 PIRM challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 0–0.

Caballero, J.; Ledig, C.; Aitken, A.; Acosta, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4778–4787.

Cao, J.; Li, Y.; Zhang, K.; and Van Gool, L. 2021. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*.

Chan, K. C.; Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2021. BasicVSR: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4947–4956.

Chan, K. C.; Zhou, S.; Xu, X.; and Loy, C. C. 2022a. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5972–5981.

Chan, K. C.; Zhou, S.; Xu, X.; and Loy, C. C. 2022b. Investigating Tradeoffs in Real-World Video Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5962–5971.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Haris, M.; Shakhnarovich, G.; and Ukita, N. 2019. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3897–3906.

Huang, Y.; Li, S.; Wang, L.; Tan, T.; et al. 2020. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems*, 33: 5632–5643.

Huang, Y.; Wang, W.; and Wang, L. 2015. Bidirectional recurrent convolutional networks for multi-frame super-resolution. *Advances in neural information processing systems*, 28.

Huang, Y.; Wang, W.; and Wang, L. 2017. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 1015–1028.

Isobe, T.; Li, S.; Jia, X.; Yuan, S.; Slabaugh, G.; Xu, C.; Li, Y.-L.; Wang, S.; and Tian, Q. 2020a. Video super-resolution with temporal group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8008–8017.

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

Ji, X.; Cao, Y.; Tai, Y.; Wang, C.; Li, J.; and Huang, F. 2020. Real-world super-resolution via kernel estimation and noise injection. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 466–467.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.

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

Li, W.; Tao, X.; Guo, T.; Qi, L.; Lu, J.; and Jia, J. 2020. Mucan: Multi-correspondence aggregation network for video super-resolution. In *European conference on computer vision*, 335–351. Springer.

Liang, J.; Cao, J.; Fan, Y.; Zhang, K.; Ranjan, R.; Li, Y.; Timofte, R.; and Van Gool, L. 2022. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*.

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

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, 1045–1048. Makuhari.

Mittal, A.; Moorthy, A. K.; and Bovik, A. C. 2011. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, 723–727. IEEE.

Mittal, A.; Soundararajan, R.; and Bovik, A. C. 2012. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3): 209–212.

Nah, S.; Baik, S.; Hong, S.; Moon, G.; Son, S.; Timofte, R.; and Mu Lee, K. 2019. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Nah, S.; Son, S.; and Lee, K. M. 2019. Recurrent neural networks with intra-frame iterations for video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8102–8111.

Pan, J.; Bai, H.; Dong, J.; Zhang, J.; and Tang, J. 2021. Deep blind video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4811–4820.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Ranjan, A.; and Black, M. J. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4161–4170.

Sajjadi, M. S.; Vemulapalli, R.; and Brown, M. 2018. Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6626–6634.

Tian, Y.; Zhang, Y.; Fu, Y.; and Xu, C. 2020. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3360–3369.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, X.; Chan, K. C.; Yu, K.; Dong, C.; and Change Loy, C. 2019. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Wang, X.; Xie, L.; Dong, C.; and Shan, Y. 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1905–1914.

Xiang, X.; Tian, Y.; Zhang, Y.; Fu, Y.; Allebach, J. P.; and Xu, C. 2020. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3370–3379.

Xie, C.; Wu, Y.; Maaten, L. v. d.; Yuille, A. L.; and He, K. 2019. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 501–509.

Yang, R. 2021. NTIRE 2021 challenge on quality enhancement of compressed video: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 647–666.

Yang, R.; Timofte, R.; Zheng, M.; Xing, Q.; Qiao, M.; Xu, M.; Jiang, L.; Liu, H.; Chen, Y.; Ben, Y.; et al. 2022a. NTIRE 2022 challenge on super-resolution and quality enhancement of compressed video: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1221–1238.

Yang, S.; Wu, T.; Shi, S.; Lao, S.; Gong, Y.; Cao, M.; Wang, J.; and Yang, Y. 2022b. MANIQA: Multi-dimension Attention Network for No-Reference Image Quality Assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1191–1200.

Yang, X.; Xiang, W.; Zeng, H.; and Zhang, L. 2021. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4781–4790.

Yaz, Y.; Foo, C.-S.; Winkler, S.; Yap, K.-H.; Piliouras, G.; Chandrasekhar, V.; et al. 2018. The unusual effectiveness of averaging in GAN training. In *International Conference on Learning Representations*.

Zhang, K.; Liang, J.; Van Gool, L.; and Timofte, R. 2021. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4791–4800.

Zhong, Z.; Gao, Y.; Zheng, Y.; and Zheng, B. 2020. Efficient spatio-temporal recurrent neural network for video deblurring. In *European Conference on Computer Vision*, 191–207. Springer.

Zhou, S.; Zhang, J.; Pan, J.; Xie, H.; Zuo, W.; and Ren, J. 2019. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2482–2491.