Investigating Tradeoffs in Real-World Video Super-Resolution

Kelvin C.K. Chan Shangchen Zhou Xiangyu Xu Chen Change Loy
S-Lab, Nanyang Technological University
{chan0899, s200094, xiangyu.xu, ccloy}@ntu.edu.sg

Figure 1. **Results on a Real-World Video.** In this work, we investigate various tradeoffs caused by the complex and diverse degradations in real-world VSR. Such tradeoffs are largely neglected in the literature. We propose simple yet effective solutions to the tradeoffs, and the resulting model *RealBasicVSR* acts as a strong baseline for real-world VSR. **(Zoom-in for best view)**

Abstract

The diversity and complexity of degradations in real-world video super-resolution (VSR) pose non-trivial challenges in inference and training. First, while long-term propagation leads to improved performance in cases of mild degradations, severe in-the-wild degradations could be exaggerated through propagation, impairing output quality. To balance the tradeoff between detail synthesis and artifact suppression, we found an image pre-cleaning stage indispensable to reduce noises and artifacts prior to propagation. Equipped with a carefully designed cleaning module, our RealBasicVSR outperforms existing methods in both quality and efficiency (Fig. 1). Second, real-world VSR models are often trained with diverse degradations to improve generalizability, requiring increased batch size to produce a stable gradient. Inevitably, the increased computational burden results in various problems, including 1) speed-performance tradeoff and 2) batch-length tradeoff. To alleviate the first tradeoff, we propose a stochastic degradation scheme that reduces up to 40% of training time without sacrificing performance. We then analyze different training settings and suggest that employing longer sequences rather than larger batches during training allows more effective uses of temporal information, leading to more stable performance during inference. To facilitate fair comparisons, we propose the new VideoLQ dataset, which contains a large variety of real-world low-quality video sequences containing rich textures and patterns. Our dataset can serve as a common ground for benchmarking. Code, models, and the dataset will be made publicly available at https://github.com/ckkelvinchan/RealBasicVSR.

1. Introduction

In real-world video super-resolution (VSR), we aim at increasing the resolution of videos containing unknown degradations. The diversity of degradations in this task poses significant challenges in designing benchmarks and training settings, and hence earlier works assume either synthetic [4, 6, 36] or camera-specific [42] degradations and focus on network designs. Although these works achieve remarkable success in restricted settings, the designs for these over-simplified scenarios cannot generalize well to the complex degradations in the wild. In addition, the complexity and diversity of degradations in real-world VSR introduce extra obstacles in both inference and training, including artifact amplification and increased computational budgets. This paper dives into the problems and tradeoffs in real-world VSR to share useful experiences in addressing the task.

It is shown by Chan *et al*. [4] that long-term information is beneficial to restoration. However, in real-world VSR, such information could also result in exaggerated artifacts, owing to error accumulation during propagation. This phenomenon leads to a tradeoff between *enhancing details* and *suppressing artifacts*, since the synthesizing power of a network comes at the cost of amplifying noises and artifacts. In this work, we show that a simple solution can sufficiently remedy this tradeoff. In particular, we place an *image cleaning* module prior to propagation for removing degradations in the input images. The resulting model *RealBasicVSR* avoids amplification of artifacts and achieves improved output quality while maintaining simplicity. We further develop a dynamic refinement scheme that repeatedly applies the cleaning module to remove excessive degradations in the inputs. Our scheme allows a flexible tradeoff between *smoothness* and *detailedness*, which can be adjusted based on a pre-defined threshold or user preference. A systematic analysis of different combinations of losses and architectures is conducted to demonstrate the significance of our designs.

Real-world VSR models are generally trained with diverse degradations to improve generalizability, and hence they are often trained with increased batch size to ensure stable gradient. As a result, real-world VSR usually requires a longer training time and more immense computational resources than the non-blind counterpart. This work inspects two tradeoffs in real-world VSR to improve training efficiency, hence shortening research cycles.

First, with increased batch size, training with long sequences is prohibitive owing to the I/O bottleneck induced by hardware limitations. The bottleneck is often alleviated by reducing either the batch size or sequence length, which results in degraded performance. To ameliorate the problem, we propose a *stochastic degradation* scheme that effectively reduces the I/O bottleneck without sacrificing the output quality. Notably, our degradation scheme yields up to *40%* reduction of training time in comparison to the conventional training scheme.

Second, with a fixed computational budget, the increased batch size in real-world VSR inevitably decreases sequence length during training. We are interested in the tradeoff between them with an aim to search for a more effective setting. To this end, we compare models trained with different combinations of batch sizes and sequence lengths. We conclude that networks trained with longer sequences rather than larger batches could more effectively employ the long-term information in the input sequence, improving stability.

In addition to the studies above, we introduce a new benchmark for real-world VSR. Most existing benchmarks [26, 33, 41, 43] are constructed by contaminating the high-resolution (HR) videos with pre-defined degradations. The most recent RealVSR dataset [42] exploits the dual-camera system in iPhone to capture paired data. Yet, the RealVSR dataset consists of only degradations specifically for the iPhone camera. With only pre-defined degradations, the benchmarks mentioned above cannot accurately reflect the generalizability of the models on real-world videos. In this work, we propose *VideoLQ*, a real-world video dataset consisting of diverse LR videos to cover various contents, resolutions, and degradations. Our dataset could serve as a common benchmark for future methods. We test existing methods on our datasets. Their quantitative and qualitative results and our dataset will be released for ease of future research.

2. Related Work

Video Super-Resolution. Most existing VSR methods [2, 4–6, 15–17, 19, 36, 40–42] are trained with pre-defined degradations (*e.g*., either synthetic [26, 33, 41, 43] or camera-specific [42]), and they deteriorate significantly when handling unknown degradations in reality. However, extending from non-blind VSR to real-world VSR is non-trivial due to various problems induced by the complex degradations in the wild. For example, artifact amplification during long-term propagation limits the performance of existing VSR methods, and increased computational costs lengthen research cycles. In this work, we investigate the challenges in both inference and training, and provide respective solutions to the challenges.

Real-World Super-Resolution. Extended from synthetic settings [3, 8–10, 38, 46], *blind* super-resolution [12, 14, 18, 24, 25, 28, 40] assumes the inputs are degraded by a known process with unknown parameters. The networks are trained with a pre-defined set of degradations with the parameters chosen at random. While the trained networks are able to restore images/videos with a range of degradations, the variation of degradations is often limited, and the generalizability to real-world degradations is in doubt.

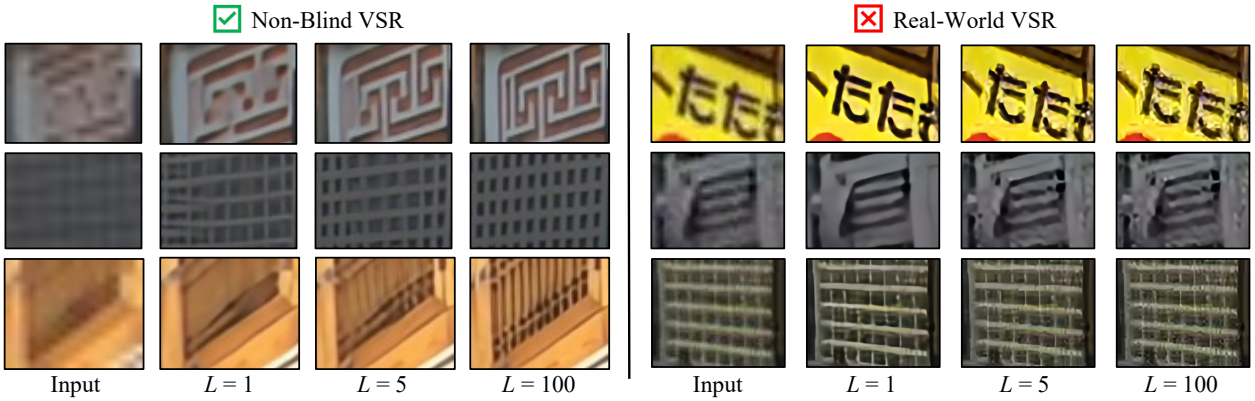


Figure 2. **Effects of Long-Term Propagation.** While employing long-term information leads to improved performance in non-blind VSR, propagation in real-world scenarios could lead to undesirable artifacts. L denotes the sequence length. **(Zoom-in for best view)**

Two recent studies [37, 45] propose to employ more diverse degradations for data augmentation during training. By using ESRGAN [39] with no changes in architecture, these two methods demonstrate promising performance in real-world images. However, we find that such a direct extension at the data augmentation level is not feasible in real-world VSR as the network tends to amplify the noise and artifacts. In this work, we investigate the cause and propose a simple yet effective *image cleaning* module to remedy the problem. Equipped with the cleaning module, *RealBasicVSR* outperforms existing works, including [37, 45], in both quality and efficiency.

Input Pre-Processing. In this study, we find that a seemingly trivial image cleaning module is critical to remove degradations prior to propagation and suppress artifacts in the outputs. The merit is even more profound in the existence of long-term propagation. In SISR, similar notions [21, 27, 30, 35, 44] have been discussed for unsupervised settings. Despite the success in the unsupervised paradigm, input pre-processing in supervised settings and in VSR are not explored. In contrast to the works above, we focus on an entirely different supervised VSR setting to remove degradations that are amplified during long-term propagation. In addition, we devise a dynamic refinement scheme, which has not been explored in previous works, to remove excessive degradations by repeatedly applying the cleaning module during inference. We also conduct systematic analysis on our image cleaning module and refinement scheme to verify its effectiveness and provide insights for future studies.

3. Tradeoff in Inference

3.1. Motivation

VSR networks boost details and improve perceptual quality through aggregating information from multiple frames. But in the case of unseen degradations, the network may fail to distinguish unwanted artifacts from favorable details. Therefore, such artifacts and noises are enhanced through temporal propagation. To verify our hypothesis, we retrain BasicVSR [4] for real-world VSR. BasicVSR accepts sequences with arbitrary lengths, allowing us to explore the effects of temporal propagation by adjusting the sequence length. We train BasicVSR with the degradation scheme and settings of Real-ESRGAN [37], which are shown effective in real-world SISR.

As shown in Fig. 2 (left), in non-blind settings, when the sequence length L increases, BasicVSR is able to aggregate useful information through long-term propagation, generating more details in the outputs. In contrast, in real-world VSR, while propagation helps enhance details in cases of mild degradations, it is observed in Fig. 2 (right) that propagating through a longer sequence could amplify noises and artifacts. For instance, when restoring the sequence using only one frame, BasicVSR is able to remove the noises in the inputs and produce smooth outputs, but propagating across the entire sequence leads to outputs with severe artifacts.

In real-world VSR, temporal propagation is a double-edged sword. While employing long-term information helps synthesize fine details, it can also introduce unpleasant artifacts. Clearly, there is a tradeoff between *enhancing details* and *suppressing artifacts*.

3.2. Input Pre-Cleaning for Real-World VSR

Motivated by the above, we propose a simple plugin to suppress degradations prior to temporal propagation. The high-level idea is to "clean" the input sequence so that the degradations in the inputs have a weaker effect on the subsequent VSR network. Despite being conceptually simple, the designs of the module require special care. More analysis of our cleaning module can be found in Sec. 3.3.

Formulation. An overview is shown in Fig. 3. The image cleaning module is used prior to BasicVSR [4]. The input

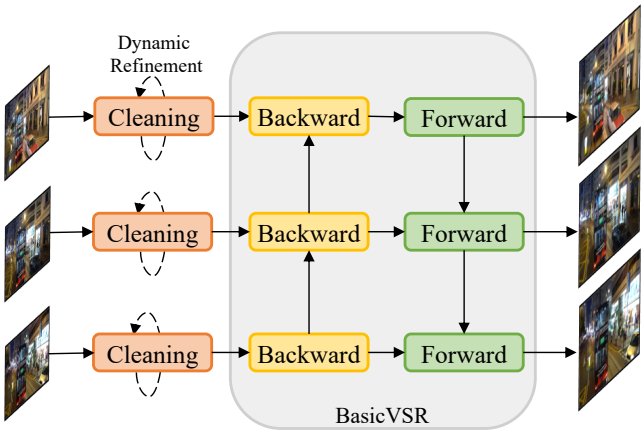


Figure 3. **Overview of RealBasicVSR.** The input images are first independently passed to our image cleaning module. The clean sequence is then passed to the VSR network (*i.e.*, BasicVSR [4]). Note that the whole network is trained end-to-end.

images are first independently passed to the cleaning module for degradation removal. Let x_i be the i-th image of the input sequence, and C be our image cleaning module, we have

$$\tilde{x}_i = C\left(x_i\right). \tag{1}$$

The clean sequence is then passed to the VSR network S for super-resolution:

$$\{y_i\} = S\left(\{\tilde{x}_i\}\right). \tag{2}$$

We adopt BasicVSR [4] in this work because of its promising performance in non-blind VSR through long-term propagation, and its simplicity in architecture.

To guide the image cleaning module, we constrain the outputs of the cleaning module with a low-resolution ground-truth:

$$\mathcal{L}_{clean} = \sum_i \rho\left(\tilde{x}_i - d(z_i)\right), \tag{3}$$

where z_i is the ground-truth high-resolution image, and d is a downsampling operator[1]. Here ρ represents the Charbonnier loss [7]. In addition to the cleaning loss, we also use the output fidelity loss to guide the cleaning module.

$$\mathcal{L}_{out} = \sum_i \rho\left(y_i - z_i\right). \tag{4}$$

Note that the cleaning module is detached from the perceptual loss [20] and adversarial loss [11] when we finetune the network with these two losses.

Dynamic Refinement. A single pass of input to the cleaning module cannot effectively remove the excessive degradations in many challenging cases. A simple yet effective method is to suppress the degradations further with another pass to the cleaning module. Formally, we design a refinement scheme that dynamically removes the degradations in test time:

$$\begin{cases} \tilde{x}_i^{j+1} = C(\tilde{x}_i^j) & \text{if } mean\left(|\tilde{x}_i^j - \tilde{x}_i^{j-1}|\right) \geq \theta, \\ \tilde{x}_i = \tilde{x}_i^j & \text{otherwise,} \end{cases} \tag{5}$$

where $\tilde{x}_i^0 = x_i$, and θ is a pre-determined stopping criteria. We find that θ=1.5 for non GAN-based models and θ=5 for GAN-based models are reasonable settings.

Architecture. In this work, we simply use a stack of residual blocks [13] as the cleaning module. It is noteworthy that while our cleaning module is conceptually straightforward, it cannot take arbitrary designs, as we will discuss in Sec. 3.3. In our design, the role of artifact suppression of VSR network is shared by the cleaning module, and hence a lighter VSR network can be adopted. In our experiments, we reduce the residual blocks in BasicVSR from 60 to 40 to maintain a comparable complexity.

[1]The `area` mode in PyTorch.

Figure 4. **Analysis of the Cleaning Module.** The proposed cleaning loss plays an important role in removing the artifacts. The design of the cleaning module requires special care. An alternative model that uses a recurrent structure fails to remove the artifacts. **(Zoom-in for best view)**

3.3. Analysis of Input Pre-Cleaning

Designs. We study the effects of the proposed image cleaning loss and the architecture of the cleaning module. Examples are shown in Fig. 4.

First, we train RealBasicVSR with the image cleaning loss (Eqn. (3)) removed. When the loss is removed, RealBasicVSR can be regarded as a single-stage network as BasicVSR. The network exaggerates the noises and artifacts,

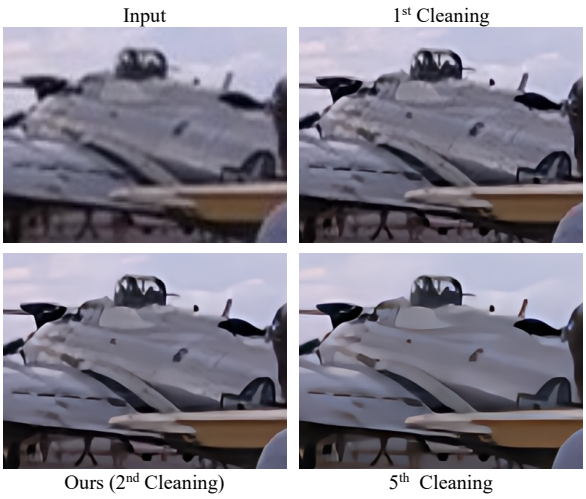
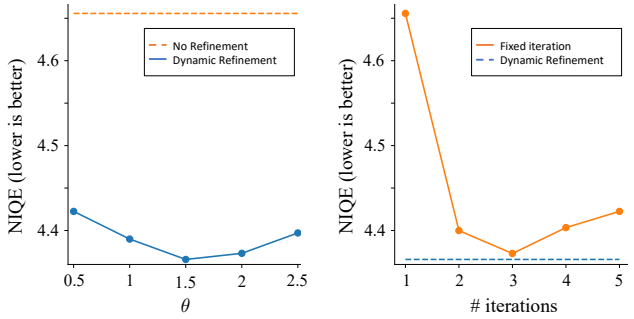


Figure 5. **Effect of Dynamic Refinement.** Our dynamic refinement scheme automatically stops the cleaning process to avoid over-smoothing and unnaturally flat regions. More examples are provided in the supplementary material. **(Zoom-in for best view)**



(a) Threshold - NIQE (b) # iterations - NIQE

Figure 6. **Ablations on Refinement.** (a) The NIQE is significantly lower with our dynamic refinement. The thresholds control the levels of details, leading to different NIQE. (b) Our dynamic refinement scheme obtains a better NIQE than fixed iterations. NIQE is computed on our VideoLQ dataset.

and the original content is distorted, showing the importance of the image cleaning loss. Note that additional losses such as adversarial loss and perceptual loss can be adopted, but we find the simplest pixelwise loss suffices.

Second, we keep the image cleaning loss and replace our cleaning module with a recurrent network. Even with the cleaning loss, the network fails to remove the unwanted degradations, also leading to distorted outputs. This observation is coherent to our hypothesis that video-based networks tend to exaggerate artifacts through aggregation, and demonstrates the importance of adopting an image-based network as the cleaning module. When compared to the aforementioned variants, our designs produce much smoother outputs, and preserve more image content.

Dynamic Refinement. In Fig. 5 we show an example using our dynamic refinement scheme. On one hand, when applying the cleaning module only once, the noises cannot be completely removed despite more details are shown. On the other hand, it is observed that the outputs are unnaturally flat and details disappear when the cleaning module is applied five times. In contrast, with our dynamic refinement scheme, the cleaning stage is halted automatically to avoid over-smoothing. We see that the outputs contain fewer artifacts while preserving necessary details. We observe that at most three iterations are needed in most scenarios.

We then study the effect of the threshold θ in Fig. 6(a). First, our dynamic refinement scheme leads to a significantly lower NIQE for all thresholds we used. Second, it is observed that different choices of thresholds lead to different levels of details, and hence different NIQE. In Fig. 6(b) we compare our scheme with fixed numbers of iterations. Our dynamic refinement scheme determines an image-specific threshold, yielding better performance. It is noteworthy that one can design a more sophisticated decision process, or manually determine the number of passes to the cleaning module. More elaborative designs of the refinement scheme are left as our future work.

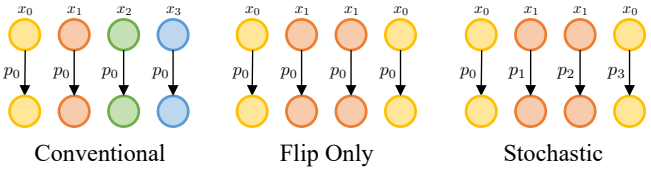


Figure 7. **Stochastic Degradation Scheme.** By loading fewer frames per iteration and using temporally-varying degradations, our stochastic degradation scheme reduces the training time by *40%* without sacrificing performance. Each circle represents one video frame, and $p_i = p_{i-1} + r_i$ (Eqn. (6)).

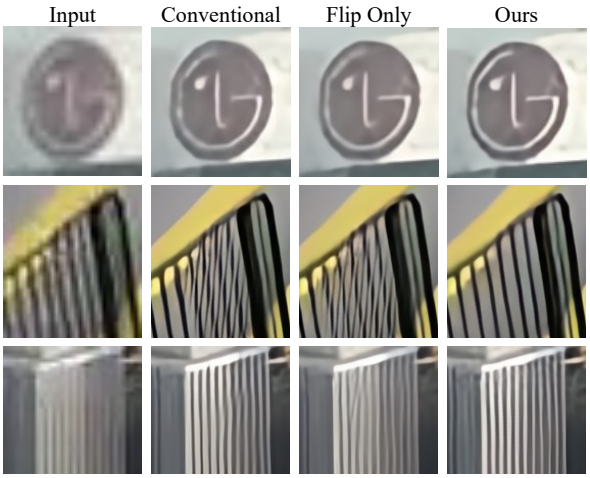


Figure 8. **Results Using Stochastic Degradation.** While directly flipping the sequence results in degraded performance, applying our stochastic degradation scheme leads to improved performance with up to 40% reduction of training time.

4. Tradeoff in Training

In real-world VSR, networks are required to deal with diverse degradations, and hence they are usually trained with multiple degradations. As a result, these networks are often trained with increased batch size to produce a stable gradient. Therefore, training real-world VSR networks often require more computational resources than the non-blind counterparts. In this work, we delve into two challenges induced by the increased computational budgets, namely 1) speed-performance tradeoff and 2) batch-length tradeoff.

4.1. Training Speed vs. Performance

When training with batch size B and sequence length L, the CPU needs to load $B\times L$ images in each iteration. With increased B in real-world VSR, severe I/O bottleneck is introduced, substantially slowing down training. Usually, the bottleneck is circumvented by reducing either the batch size or sequence length, resulting in degraded performance. In this work, we propose a stochastic degradation scheme, which significantly improves the training speed without sacrificing performance. The graphical illustration of L=4 is shown in Fig. 7.

In our stochastic degradation scheme, instead of loading L frames in each iteration, we load $L/2$ frames and flip the sequence temporally. This design allows us to train with sequences with the same length while reducing the workload of the CPU by half. However, in such a setting, the network perceives content with less variation, and the network can potentially make use of the shortcut that the sequences are temporally flipped. To improve the diversity of data, we model the degradations to each frame as a *random walk*. Specifically, let p_i be the parameters corresponding to the degradations applied to the i-th frame, we have

$$p_{i+1} = p_i + r_{i+1}. \tag{6}$$

Here r_{i+1} represents the differences between the parameters for the $(i+1)$-th and i-th frames.

As shown in Fig. 8, when compared to the conventional training scheme, directly flipping the sequence results in similar or degraded performance qualitatively. For instance, the orientations of the line patterns are distorted due to the aliasing effect in the inputs. When our stochastic degradation scheme is applied, the network is more robust to the variation of degradations, leading to improved performance. In addition, as depicted in Table 1, by reducing the number of images processed, the workload of the CPUs is significantly reduced. As a result, the I/O bottleneck is ameliorated, and the training time is reduced by up to 40%[2] without sacrificing performance.

[2]Different hardware could lead to different levels of bottleneck, and hence different levels of speedup.

Table 1. **Comparison to Stochastic Degradation Scheme.** Our scheme leads to 40% reduction of training time while maintaining comparable performance.

	Time per iteration ↓	NIQE ↓
Conventional Scheme	∼2.5s	4.7191
Flip Only	∼1.5s	4.6926
Stochastic Degradation	∼1.5s	4.6836

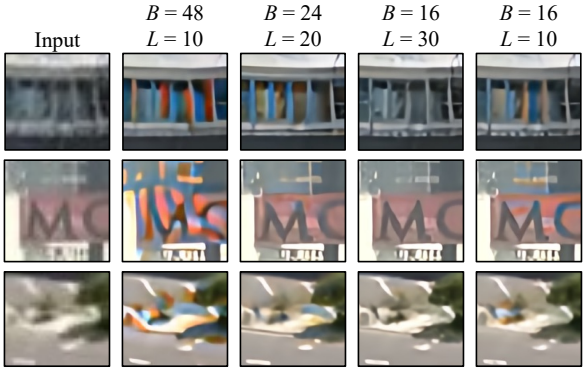


Figure 9. **Tradeoff Between Batch and Length.** With a fixed computational constraint, training with large batch size and short sequence results in color artifacts and blurrier outputs. Surprisingly, when the length is small, training with large batch size harms the performance.

4.2. Batch Size vs. Sequence Length

With a fixed computation budget, the increased batch size when training real-world VSR models inevitably leads to a decrease in sequence length. On one hand, training with a larger batch size enables the network to perceive more degradations and scene content in each iteration, leading to more stable gradients. On the other hand, training with longer sequences allows the network to employ long-term information more effectively. However, one must choose between a larger batch or a longer sequence when computational resources are limited. We are interested in the tradeoff between them, with an aim to provide an effective setting for future works. In this section, we train RealBasicVSR with a constraint of $B\times L$=480 and discuss the performance of these models. Our stochastic degradation scheme is used.

As shown in Fig. 9, when training with B=48, L=10, it is observed that the outputs contain severe color artifacts and distorted details. This undesirable effect reduces when we increase the sequence length. In particular, the color artifacts are significantly reduced when L increases from 10 to 20, and are further eliminated when L increases to 30.

The above comparison shows that training with longer sequences is preferable. We speculate that networks trained with short sequences cannot adapt to long sequences during inference, due to the domain gap between training and inference. To further demonstrate that the importance of long

Table 2. **Quantitative Comparison.** RealBasicVSR obtains the best performance on all four metrics than existing methods with faster speed. Runtime is computed with an output size of 720×1280, with an Nvidia V100 GPU. **Green** and blue colors represent the best and second best performance, respectively.

	Bicubic	BasicVSR++ [6]	RealVSR [42]	DAN [28]	DBVSR [34]	BSRGAN [45]	Real-ESRGAN [37]	RealSR [18]	**RealBasicVSR**
Params (M)	-	7.3	**2.7**	4.3	25.5	16.7	16.7	16.7	6.3
Runtime (ms)	-	77	1082	185	239	149	149	149	**63**
NRQM [29] ↑	2.8545	3.6807	2.5322	3.4347	3.4850	5.8197	5.8129	5.7030	**6.1408**
NIQE [32] ↓	5.2762	4.3424	4.9484	4.7844	4.5383	3.2216	3.1263	3.0285	**2.5693**
PI [1] ↓	6.2109	5.3309	6.2081	5.6749	5.5267	3.7010	3.6567	3.6628	**3.2143**
BRISQUE [31] ↓	55.225	50.665	55.317	51.875	50.937	27.832	30.679	29.638	**27.697**

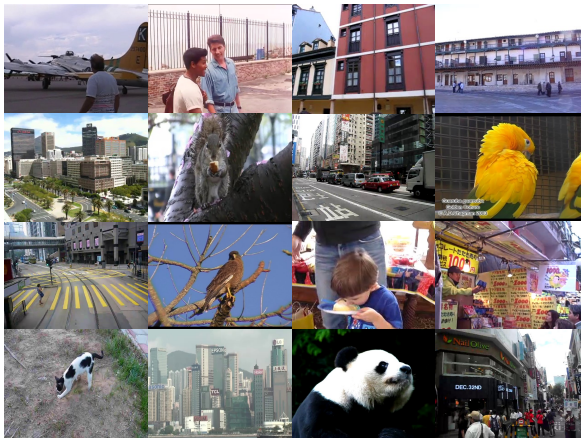


Figure 10. **VideoLQ Dataset.** Our *VideoLQ* dataset consists of videos with a wide range of content and resolutions, collected from different video hosting sites such as Flickr and YouTube. It can be served as a common ground for future comparison.

sequences in training, we fix B to 16 and reduce L from 30 to 10. It is observed that the corresponding regions shows the same color artifacts and blur when L is reduced. Therefore, it is suggested to employ a longer sequence when a computational constraint is imposed.

5. VideoLQ Dataset and Benchmark

To assess the generalizability of real-world VSR methods, a benchmark that covers a wide range of *degradations*, *content*, and *resolution* is indispensable. Most existing datasets [26, 33, 41, 43] focus only on synthetic degradations such as bicubic downsampling, and hence they cannot reflect the efficacy of real-world VSR methods. The recent RealVSR dataset [42] consists of LR-HR pairs of videos captured by the dual-camera system in iPhone. Although the data is not constructed by synthetic degradations, the sequences are captured by a single camera, and hence the LR videos contain only camera-specific degradations. Hence, there is no guarantee that methods performing superiorly in the RealVSR dataset can generalize to videos in the wild.

In this work, we propose the *VideoLQ* dataset. Examples of the videos are shown in Fig. 10. The videos in our VideoLQ dataset are downloaded from various video-hosting sites such as Flickr and YouTube, with a Creative Common license. To ensure diversity of the videos, we select videos with different resolutions and contents to cover as many degradations as possible. For each video, we extract a sequence of 100 frames with no scene changes allowed, so that methods relying on long-term propagation can be assessed. The sequences are selected to contain enough textures or texts for ease of comparison. Additionally, the ground-truth videos in Vid4 [26] are also included.

5.1. Experimental Settings

We conduct experiments on our VideoLQ dataset. We compare our RealBasicVSR with seven state of the arts, including four image models: RealSR [18], DAN [28], Real-ESRGAN [37], BSRGAN [45] and three video models: BasicVSR++[3] [6], RealVSR [42], DBVSR [34]. More discussion are provided in the supplementary material.

Training Degradations. Following Real-ESRGAN [37], we adopt the second-order order degradation model, and we apply random blur, resize, noise, and JPEG compression as image-based degradations. In addition, we incorporate video compression, which is a common technique to reduce video size. Unlike the aforementioned degradations, video compression implicitly considers the inter-dependencies between video frames, providing us with temporally and spatially varying degradations. We apply compression with randomly selected codecs and bitrates during training, and we observe performance gain with video compression included. The detailed settings are provided in the supplementary material. For the methods in comparison, we use their publicly available code.

Training Settings. Following DBVSR [34], we use the REDS dataset [33] for training. We adopt Adam optimizer [22] with constant learning rates. The patch size of input LR frames is 64×64. We apply our stochastic degradation scheme with temporal length 30[4]. The training is divided into two stages: We first pre-train RealBasicVSR with only output loss and image cleaning loss for 300K iterations, with batch size 16 and learning rate 10^{-4}. We then finetune the network with also perceptual loss [20] and adversarial loss [11] for 150K iterations. The batch size is reduced to 8. The learning rates of the generator and discriminator are set to 5×10^{-5} and 10^{-4}.

[3] Trained with bicubic downsampling as a reference.

[4] That means, the CPU loads 15 images in each iteration.

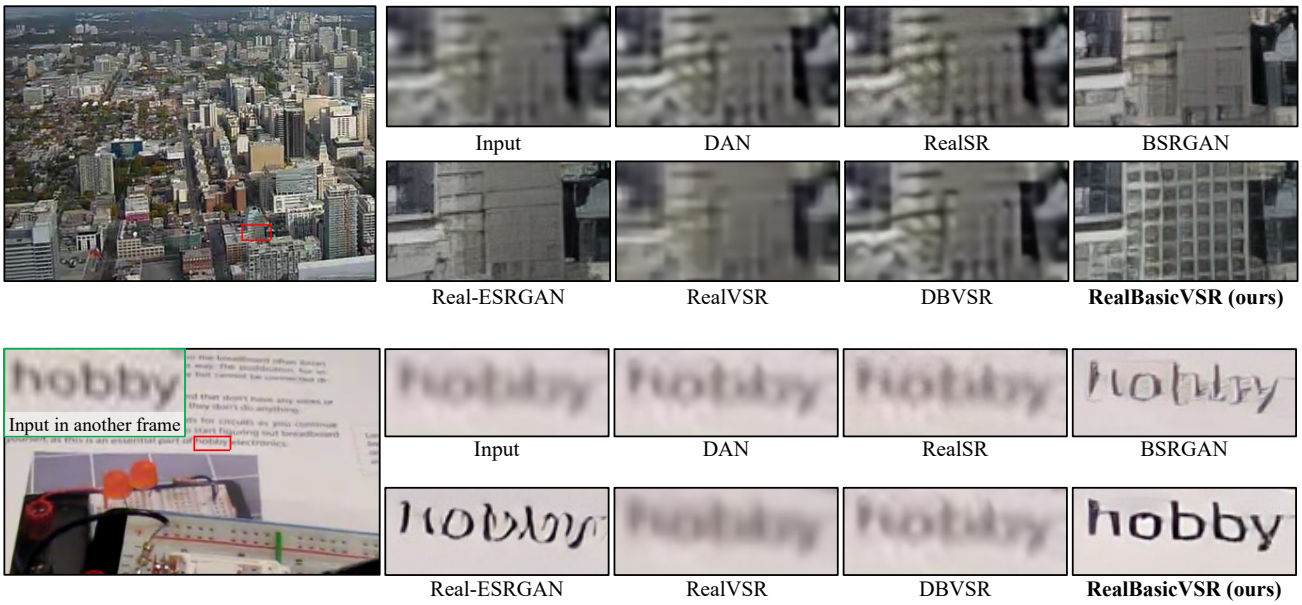


Figure 11. **Qualitative Comparison.** Our RealBasicVSR is able to aggregate long-term information effectively. It generates much more details when compared to existing works. In particular, by aggregating long-term information through propagation, RealBasicVSR successfully restores the word "hobby", which can be clearly seen in latter frames . **(Zoom-in for best view)**

Architecture. In the adversarial training, we use RealBasicVSR as the generator, and adopt the discriminator of Real-ESRGAN. For the generator, our image cleaning module C consists of 20 residual blocks. We use BasicVSR as our VSR network S, with the number of residual blocks set to 40. The number of feature channels is 64. Detailed experimental settings and model architectures are provided in the supplementary material.

Quantitative Metric. As ground-truths are not available for real-world videos, common metrics such as PSNR and SSIM cannot be used in this task. Therefore, we adopt four commonly used non-reference metrics NIQE [32], NRQM [29], PI [1], and BRISQUE [31] to supplement our qualitative comparison.

5.2. Comparison to State of the Arts

We show real-world examples on our VideoLQ dataset in Fig. 11. Equipped with the image cleaning module, RealBasicVSR is able to aggregate long-term information through propagation effectively. As a result, it generates much more details in fine regions, improving visual quality. For instance, only RealBasicVSR is able to recover the word "hobby", which can be clearly seen in other frames.

In addition to qualitative results, we also provide quantitative measures as a reference. When compared to existing methods, RealBasicVSR achieves better performance on all metrics with faster speed. In particular, RealBasicVSR outperforms the recent RealVSR [42] with 17× faster speed. When compared to Real-ESRGAN [37], which uses a similar training pipeline, RealBasicVSR performs superiorly with lower complexity and faster speed.

The above methods employ only either single-image or short-term information. While these methods demonstrate significant improvements in terms of degradation removal, they cannot effectively recover the details beyond the input image and its local neighbors, which require aggregation of information from distant frames. In contrast, RealBasicVSR explores the possibility of exploiting long-term information in real-world VSR, and both our qualitative and quantitative results show the effectiveness of RealBasicVSR in exploiting such information for detail synthesis.

6. Discussion

A common belief in existing VSR studies [4, 6] is that long-term propagation is beneficial to restoration performance. Yet, such discussion is limited to non-blind VSR. In this work, we examine the contributions of temporal propagation in real-world VSR and find that long-term information is also beneficial to this task but do not come for free, due to the diverse and complicated degradations in the wild. As an explorational study, we reveal several challenges in real-world VSR. We find that the domain gap on degradations and the increased computational costs result in various challenges and tradeoffs. We then provide respective solutions to the challenges including the cleaning module and stochastic degradation scheme, which are easy to implement. We hope our study and findings in our work as well as our VideoLQ dataset will lay a good foundation and inspire future works in real-world VSR.

References

[1] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 PIRM challenge on perceptual image super-resolution. In *ECCVW*, 2018. 7, 8

[2] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021. 2

[3] Kelvin C.K. Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. GLEAN: Generative latent bank for large-factor image super-resolution. In *CVPR*, 2021. 2

[4] 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. 2, 3, 4, 8, 11

[5] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. In *AAAI*, 2021. 2

[6] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. *arXiv preprint arXiv:2104.13371*, 2021. 2, 7, 8, 11

[7] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 4, 11

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 2

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2016. 2

[10] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 4, 7, 11

[12] Jinjin Gu, Hannan Lu, Zuo Wangmeng, and Chao Dong. Blind super-resolution with iterative kernel correction. In *CVPR*, 2019. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[14] Zheng Hui, Jie Li, Xiumei Wang, and Xinbo Gao. Learning the non-differentiable optimization for blind super-resolution. In *CVPR*, 2021. 2

[15] 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

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

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

[18] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *CVPRW*, 2020. 2, 7, 11

[19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 2

[20] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2017. 4, 7, 11

[21] Gwantae Kim, Jaihyun Park, Kanghyu Lee, Junyeop Lee, Jeongki Min, Bokyeung Lee, David K Han, and Hanseok Ko. Unsupervised real-world super resolution with cycle generative adversarial network and domain discriminator. In *CVPRW*, 2020. 3

[22] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[23] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017. 11

[24] Jingyun Liang, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Mutual affine network for spatially variant kernel estimation in blind image super-resolution. In *ICCV*, 2021. 2

[25] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *CVPR*, 2021. 2

[26] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *TPAMI*, 2014. 2, 7

[27] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *CVPRW*, 2019. 3

[28] Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. In *NeurIPS*, 2020. 2, 7, 11

[29] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16, 2017. 7, 8

[30] Shunta Maeda. Unpaired image super-resolution using pseudo-supervision. In *CVPR*, 2020. 3

[31] Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. 2011. 7, 8

[32] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 2013. 7, 8

[33] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019. 2, 7

[34] Jinshan Pan, Songsheng Cheng, Jiawei Zhang, and Jinhui Tang. Deep blind video super-resolution. In *ICCV*, 2021. 7, 11

[35] Mohammad Saeed Rad, Thomas Yu, Claudiu Musat, Hazim Kemal Ekenel, Behzad Bozorgtabar, and Jean-Philippe Thiran. Benefiting from bicubically down-sampled

images for learning real-world image super-resolution. In *WACV*, 2021. 3

[36] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019. 2

[37] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021. 3, 7, 8, 11

[38] 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. 2

[39] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 3

[40] Xiangyu Xu, Yongrui Ma, and Wenxiu Sun. Towards real scene super-resolution with raw images. In *CVPR*, 2019. 2

[41] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019. 2, 7

[42] Xi Yang, Wangmeng Xiang, Hui Zeng, and Lei Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *ICCV*, 2021. 2, 7, 8, 11

[43] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, 2019. 2, 7

[44] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *CVPRW*, 2018. 3

[45] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, 2021. 3, 7, 11

[46] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. In *NeurIPS*, 2020. 2

A. Architecture and Experimental Settings

Architecture. We use a simple architecture in this work for explorational purpose. First, a convolution is used to extract shallow features from the input image. A stack of 20 residual blocks are then used to extract deep features. A final convolutional layer is then used to produce the clean image. We adopt BasicVSR [4] as the VSR network. We reduce the number of residual blocks from 60 to 40 to maintain comparable complexity to the original BasicVSR.

Loss Function. For the output fidelity loss $\mathcal{L}_{pix}$ and image cleaining loss $\mathcal{L}_{clean}$, we use Charbonnier loss [7] since it better handles outliers and improves the performance over the conventional ℓ_2 loss [23]. In addition, we use perceptual loss [20] $\mathcal{L}_{per}$ and adversarial loss [11] $\mathcal{L}_{adv}$ to achieve better visual quality.

In the first stage, we pretrain the generator (*i.e.*, RealBasicVSR) with the fidelity loss and image cleaning loss:

$$\mathcal{L}_{1st} = \mathcal{L}_{pix} + \mathcal{L}_{clean}. \tag{7}$$

We then finetune the network with also perceptual loss and adversarial loss:

$$\mathcal{L}_{2nd} = \mathcal{L}_{pix} + \mathcal{L}_{clean} + \lambda_{per}\mathcal{L}_{per} + \lambda_{adv}\mathcal{L}_{adv}. \tag{8}$$

In our experiments, λ_{per}=1 and λ_{adv}=5×10^{-2}. Note that in the second stage, the weights of the cleaning module are kept fixed.

Training Degradations. Following Real-ESRGAN [37], we adopt the second-order order degradation model, and we apply random blur, resize, noise, and JPEG compression as image-based degradations. In addition, we incorporate video compression, which is a common technique to reduce video size. Unlike the aforementioned degradations, video compression implicitly considers the interdependencies between video frames, providing us with temporally and spatially varying degradations. The settings of image-based degradations follow Real-ESRGAN [37]. For the video compression, in each iteration, we randomly select one of the following codecs: "libx264", "h264", and "mpeg4". The bitrate is uniformly selected from the range $[10^4, 10^5]$. Video compression is added right after JPEG compression.

Implementation. We implement our models with PyTorch and train the models using eight NVIDIA Tesla V100 GPUs. Code will be made publicly available.

B. Discussion of Baselines

In this work, we compare our RealBasicVSR with seven state of the arts, including four image models: RealSR [18], DAN [28], Real-ESRGAN [37], BSRGAN [45] and three video models: BasicVSR++[5] [6], RealVSR [42], DBVSR [34]. They are representative methods in image and video super-resolution that achieve promising performance.

With specific designs in training, these methods demonstrate significant improvements when compared to non-blind methods. However, while these methods succeed in removing degradations in the input images, they are inferior in recovering details beyond the image itself or its local neighbors, due to the fact that they do not exploit long-term information available in videos.

Despite being extensively discussed in non-blind VSR, the use of long-term information has not been explored in real-world VSR. In this work, we find that such long-term information, if used with designated designs, is also useful in real-world VSR. With the benefits of our findings and designs, RealBasicVSR is able to restore more details than the methods in comparison, as shown in Fig. 12 and Fig. 13.

C. Dynamic Refinement

In this section, we show additional examples demonstrating the effects of our dynamic refinement. As shown in Fig. 14, unpleasant artifacts remain in the outputs when applying cleaning once, and unnatually flat outputs due to over-cleaning are observed when our cleaning module is applied five times. In contrast, our refinement scheme automatically stops the refinement to avoid over-smoothing while cleaning excessive artifacts, leading to improved performance. More sophisticated decision processes are left as our future work.

[5] Trained with bicubic downsampling, as a reference.

Figure 12. **Qualitative Comparison.** By employing the long-term information effectively, RealBasicVSR restores more details when compared to existing state of the arts. **(Zoom-in for best view)**

Figure 13. **Qualitative Comparison.** By employing the long-term information effectively, RealBasicVSR restores more details when compared to existing state of the arts. **(Zoom-in for best view)**

Figure 14. **Dynamic Refinement.** Our dynamic refinement scheme removes remaining noises and artifacts in the first cleaning while avoiding over-smoothing. **(Zoom-in for best view)**