BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond

Kelvin C.K. Chan[1] Xintao Wang[2] Ke Yu[3] Chao Dong[4,5] Chen Change Loy[1*]
[1]S-Lab, Nanyang Technological University [2]Applied Research Center, Tencent PCG
[3]CUHK – SenseTime Joint Lab, The Chinese University of Hong Kong
[4]Shenzhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[5]SIAT Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society
{chan0899, ccloy}@ntu.edu.sg xintao.wang@outlook.com
yk017@ie.cuhk.edu.hk chao.dong@siat.ac.cn

Abstract

Video super-resolution (VSR) approaches tend to have more components than the image counterparts as they need to exploit the additional temporal dimension. Complex designs are not uncommon. In this study, we wish to untangle the knots and reconsider some most essential components for VSR guided by four basic functionalities, i.e., Propagation, Alignment, Aggregation, and Upsampling. By reusing some existing components added with minimal redesigns, we show a succinct pipeline, BasicVSR, that achieves appealing improvements in terms of speed and restoration quality in comparison to many state-of-the-art algorithms. We conduct systematic analysis to explain how such gain can be obtained and discuss the pitfalls. We further show the extensibility of BasicVSR by presenting an information-refill mechanism and a coupled propagation scheme to facilitate information aggregation. The BasicVSR and its extension, IconVSR, can serve as strong baselines for future VSR approaches.

1. Introduction

Compared to single-image super-resolution, which focuses on the intrinsic properties of a single image for the upscaling task, video super-resolution (VSR) poses an extra challenge as it involves aggregating information from multiple highly-related but misaligned frames in video sequences.

Various approaches have been proposed to address the challenge. Some designs can be highly complex. For instance, in the representative method EDVR [32], a multi-scale deformable alignment module and multiple attention layers are adopted for aligning and integrating the features from different frames. In RBPN [9], multiple projection modules are used to sequentially aggregate features from multiple frames. Such designs are effective but inevitably increase the runtime and model complexity (see Fig. 1). In addition, unlike SISR, the potentially complex and dissimilar designs of VSR methods pose difficulties in implementing and extending existing approaches, hampering reproducibility and fair comparisons.

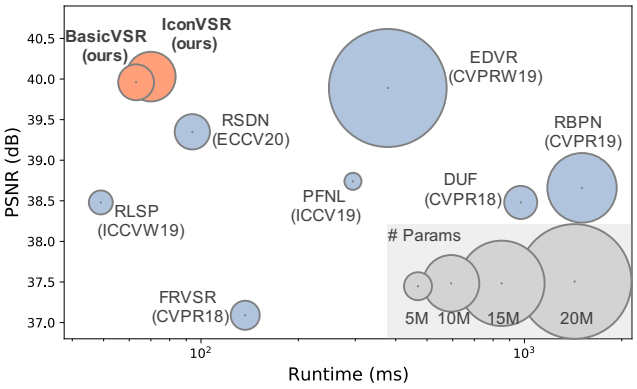


Figure 1. **Speed and performance comparison.** Without bells and whistles, BasicVSR outperforms state-of-the-art methods with high efficiency. Built upon BasicVSR, IconVSR further improves the performance. Comparisons are performed on UDM10 dataset [34].

There is a need to step back and reconsider the diverse designs of VSR models, with the aim to search for a more generic, efficient, and easy-to-implement baseline for VSR. We start our search by decomposing popular VSR approaches into submodules based on functionalities. As summarized in Table 1, most existing methods entail four inter-related components, namely, *propagation, alignment, aggregation, and upsampling*. Such a decomposition allows

*Corresponding author

Table 1. **Components in existing VSR methods**. We categorize components based on their functionalities: i) *Propagation* refers to the way in which features are propagated temporally, ii) *Alignment* concerns on the spatial transformation applied to misaligned images/features, iii) *Aggregation* defines the steps to combine aligned features, and iv) *Upsampling* describes the method to transform the aggregated features to the final output image. Bolded texts correspond to designs that were reported to achieve better performance in the literature.

	Sliding-Window			Recurrent				
	EDVR [31]	MuCAN [20]	TDAN [30]	BRCN [10, 11]	FRVSR [25]	RSDN [12]	**BasicVSR**	**IconVSR**
Propagation	Local	Local	Local	**Bidirectional**	Unidirectional	Unidirectional	**Bidirectional**	**Bidirectional** (coupled)
Alignment	**Yes** (DCN)	**Yes** (correlation)	**Yes** (DCN)	No	**Yes** (flow)	No	**Yes** (flow)	**Yes** (flow)
Aggregation	Concatenate + **TSA**	Concatenate	Concatenate	Concatenate	Concatenate	Concatenate	Concatenate	Concatenate + **Refill**
Upsampling	Pixel-Shuffle	Pixel-Shuffle	Pixel-Shuffle	Pixel-Shuffle	Pixel-Shuffle	Pixel-Shuffle	Pixel-Shuffle	Pixel-Shuffle

us to systematically study various options under each component and understand their pros and cons.

Through extensive experiments, we find that with minimal redesigns of existing options, one could already reach a strong yet efficient baseline for VSR without bells and whistles. In this paper, we highlight one of such possibilities, named **BasicVSR**. We observe that, among the four aforementioned components, the choices of propagation and alignment components could lead to a big swing in terms of performance and efficiency. Our experiments suggest the use of bidirectional propagation scheme to maximize information gathering, and an optical flow-based method to estimate the correspondence between two neighboring frames for feature alignment. By simply streamlining these propagation and alignment components with the commonly-adopted designs for aggregation (*i.e*. feature concatenation) and upsampling (*i.e*. pixel-shuffle [27]), BasicVSR outperforms existing state of the arts [9, 12, 32] in both performance (up to 0.61 dB) and efficiency (up to $24\times$ speedup).

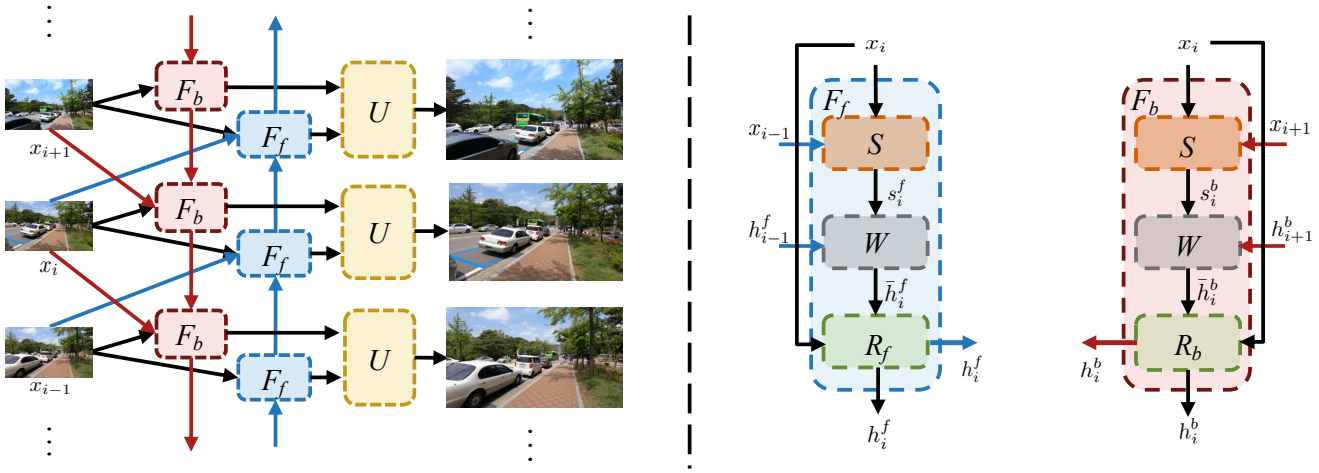
Thanks to its simplicity and versatility, BasicVSR provides a viable starting point for extending to more elaborated networks. By using BasicVSR as a foundation, we present **IconVSR** that comprises two novel extensions to improve the aggregation and the propagation components. The first extension is named *information-refill*. This mechanism leverages an additional module to extract features from sparsely selected frames (keyframes), and the features are then inserted into the main network for feature refinement. The second extension is a *coupled propagation* scheme, which facilitates information exchange between the forward and backward propagation branches. The two modules not only reduce error accumulation during propagation due to occlusions and image boundaries, but also allow the propagation to access complete information in a sequence for generating high-quality features. With these two new designs, IconVSR surpasses BasicVSR with a PSNR improvement of up to 0.31 dB.

We believe that our work is timely, given the increasing number of approaches centered around the research of VSR. A strong, simple yet extensible baseline is needed. Guided by the main functionalities in VSR approaches, we reconsider some essential components in existing pipelines and present an efficient baseline for VSR. We show that simple components, when integrated properly, would synergize and lead to state-of-the-art performance. We further present an example of extending BasicVSR with two novel modules to refine the propagation and aggregation components.

2. Related Work

Existing VSR approaches [10, 21, 28, 34, 20, 12, 13] can be mainly divided into two frameworks – *sliding-window* and *recurrent*. Earlier methods [1, 29, 33] in the sliding-window framework predict the optical flow between low-resolution (LR) frames and perform spatial warping for alignment. Later approaches resort to a more sophisticated approach of implicit alignment. For example, TDAN [30] adopts deformable convolutions (DCNs) [5, 37] to align different frames at the feature level. EDVR [32] further uses DCNs in a multi-scale fashion for more accurate alignment. DUF [16] leverages dynamic upsampling filters to handle motions implicitly. Some approaches take a recurrent framework. RSDN [12] proposes a recurrent detail-structural block and a hidden state adaptation module to enhance the robustness to appearance change and error accumulation. RRN [14] adopts a residual mapping between layers with identity skip connections to ensure a fluent information flow and preserve the texture information over long periods. The aforementioned studies have led to many new and sophisticated components to address the propagation and alignment problems in VSR. Here, we reinvestigate some of the components and find that bidirectional propagation coupled with a simple optical flow-based feature alignment suffice to outperform many state-of-the-art methods.

The information-refill mechanism in IconVSR is reminiscent of the concept of interval-based processing [4, 15, 26, 35, 36, 38, 39]. These methods divide video frames into independent intervals characterized by keyframes and non-keyframes. The keyframes and non-keyframes are then processed by different pipelines. For instance, FAST [35] applies SRCNN [6, 7] to super-resolve the keyframes. Non-keyframes are then restored using the upscaled keyframes and the motion vectors stored in the compressed video codec. IconVSR inherits the concept of keyframes, but unlike existing methods that process the intervals indepen-



(a) BasicVSR architecture

(b) Forward and backward propagation branches

Figure 2. **An overview of BasicVSR.** BasicVSR is a generic and efficient baseline for VSR. With minimal redesigns of existing components including optical flow and residual blocks, it outperforms existing state of the arts with high efficiency. **(a)** BasicVSR adopts a typical bidirectional recurrent network. The upsampling module U contains multiple pixel-shuffle and convolutions. The red and blue colors represent the backward and forward propagations, respectively. **(b)** The propagation branches contain only generic components. S, W, and R refer to the flow estimation module, spatial warping module, and residual blocks, respectively.

dently, we make one advancement by connecting the intervals through the propagation branches. With this design, long-term information can be propagated across the interconnected intervals, further improving the effectiveness.

3. Methodology

Video super-resolution, by nature, involves a long and complex processing pipeline since it needs to aggregate information from not only the spatial dimension but also the temporal dimension. Existing studies typically focus on one aspect of the functionalities to make advancement and may not collectively consider the synergy of various components. There is an urge to revisit various components macroscopically and uncover a generic baseline that inherits the strengths of existing approaches. In this work, we conduct extensive analysis and present a simple, strong and versatile baseline, BasicVSR, which can serve as a backbone with abundant flexibilities in design.

3.1. BasicVSR

Aiming at discovering generic frameworks for facilitating analysis and development of VSR methods, we confine our search to commonly-adopted elements such as optical flow and residual blocks. An overview of BasicVSR is depicted in Fig. 2.

Propagation. Propagation is one of the most influential components in VSR. It specifies how the information in a video sequence is leveraged. Existing propagation schemes can be divided into three main groups: *local*, *unidirectional* and *bidirectional* propagations. In what follows, we discuss the weaknesses of the former two to motivate our choice of bidirectional propagation in BasicVSR.

- ***Local Propagation.*** The sliding-window methods [9, 13, 32] take the LR images within a local window as inputs and employ the local information for restoration. In this design, the accessible information is restricted in a local neighborhood. The omittance of distant frames inevitably limits the potential of the sliding-window methods. To verify our claim, we start with a global receptive field (in the temporal dimension) and gradually reduce the receptive field. We separate the test sequences into K segments and use our BasicVSR to restore each segment independently. The PSNR difference to the case K=1 (global propagation) is depicted in Fig. 3.

  First, the difference in PSNR is reduced (*i.e.* better performance) when the number of segments decreases (*i.e.* temporal receptive field increases). This suggests that the information in distant frames is beneficial to the restoration and should not be neglected. Second, the difference in PSNR is the largest at the two ends of each segment, indicating the necessity of adopting long sequences to accumulate long-term information.

- ***Unidirectional Propagation.*** The aforementioned problem can be resolved by adopting a unidirectional propagation [8, 12, 14, 25], where the information is sequentially propagated from the first frame to the last

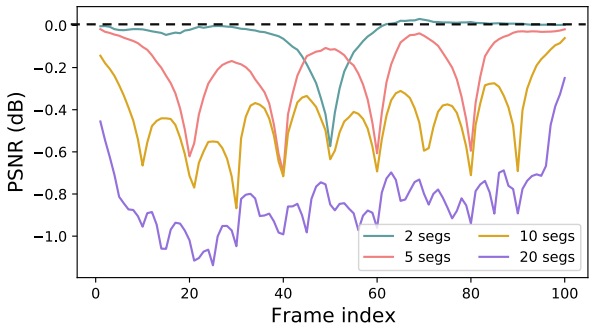


Figure 3. **Local vs Global propagation.** When the number of segments K is reduced, the increased temporal receptive field leads to higher PSNR. This demonstrates the importance of aggregating long-term information. Values smaller than zero (dotted line) indicate a lower PSNR than the case K=1.

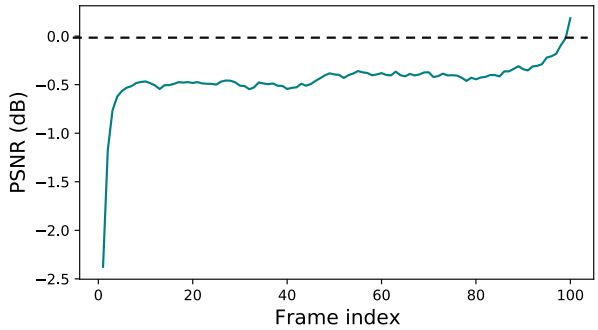


Figure 4. **Unidirectional vs Bidirectional.** In unidirectional propagation, earlier timesteps receive less information, leading to inferior performance. Values smaller than zero (dotted line) indicates a lower PSNR than the bidirectional counterpart. Note that the unidirectional model outperforms the bidirectional model only for the last frame, owing to the zero feature initialization in the bidirectional model.

frame. However, in this setting, the information received by different frames is imbalanced. Specifically, the first frame receives no information from the video sequence except itself, whereas the last frame receives information from the whole sequence. Hence, suboptimal results are expected for the earlier frames.

To demonstrate the effects, we compare BasicVSR (using bidirectional propagation) with its unidirectional variant (with comparable network complexity). From Fig. 4, we see that the unidirectional model obtains a significantly lower PSNR than bidirectional propagation at early timesteps, and the difference gradually reduces as more information is aggregated with the increase in the number of frames. Moreover, a consistent performance drop of 0.5 dB is observed with only partial information employed. These observations reveal the suboptimality of unidirectional propagation. One can improve the output quality by propagating information back from the last frame of the sequence.

- ***Bidirectional Propagation.*** The above two problems can be simultaneously addressed by bidirectional propagation, in which the features are propagated forward and backward in time independently. Motivated by this, BasicVSR adopts a typical bidirectional propagation scheme. Given an LR image x_i, its neighboring frames x_{i-1} and x_{i+1}, and the corresponding features propagated from its neighbors, denoted as h^f_{i-1} and h^b_{i+1}, we have

$$\begin{aligned} h^b_i &= F_b(x_i, x_{i+1}, h^b_{i+1}), \\ h^f_i &= F_f(x_i, x_{i-1}, h^f_{i-1}), \end{aligned} \tag{1}$$

where F_b and F_f denote the backward and forward propagation branches, respectively.

Alignment. Spatial alignment plays an important role in VSR as it is responsible to align highly related but misaligned images/features for subsequent aggregation. Mainstream works can be divided into three categories: *without alignment*, *image alignment*, and *feature alignment*. In this section, we conduct experiments to analyze each of the categories and to validate our choice of feature alignment.

- ***Without Alignment.*** Existing recurrent methods [8, 10, 11, 12, 14] generally do not perform alignment during propagation. The non-aligned features/images impede aggregation and eventually lead to substandard performance. This suboptimality can be reflected by our experiment, where we remove the spatial alignment module in BasicVSR. In this case, we directly concatenate the non-aligned features for restoration. Without proper alignment, the propagated features are not spatially aligned with the input image. As a result, the local operations such as convolutions, which have relatively small receptive fields, are inefficient in aggregating the information from corresponding locations. A drop of 1.19 dB of PSNR is observed. This result suggests that it is pivotal to adopt operations that have a large enough receptive field to aggregate information from distant spatial locations.

- ***Image Alignment.*** Earlier works [17, 33] perform alignment by computing the optical flow and warping the images before restoration. Recently, Chan *et al*. [2] show that moving the spatial alignment from the image level to the feature level yields a marked improvement. In this work, we further conduct experiments to verify their claim. We compare image warping and feature warping[1] on a variant of BasicVSR. Resulting from the inaccuracy of optical flow estimation, the warped

[1] We compute optical flow from the images and use the optical flow for feature warping.

images inevitably suffer from blurriness and incorrectness. The loss of details eventually leads to degraded outputs. In our experiments, a drop of 0.17 dB is observed when adopting image alignment. This observation confirms the necessity of shifting the spatial alignment to the feature level.

- ***Feature Alignment.*** The inferior performance of removing/image alignment motivates us to resort to feature alignment. Similar to flow-based methods [17, 25, 33], BasicVSR adopts optical flow for spatial alignment. But instead of warping the images as in previous works, we perform warping on the features for better performance. The aligned features are then passed to multiple residual blocks for refinement. Formally, we have

$$\begin{aligned} s_i^{\{b,f\}} &= S(x_i, x_{i\pm1}), \\ \bar{h}_i^{\{b,f\}} &= W(h_{i\pm1}^{\{b,f\}}, s_i^{\{b,f\}}), \\ h_i^{\{b,f\}} &= R_{\{b,f\}}(x_i, \bar{h}_i^{\{b,f\}}), \end{aligned} \tag{2}$$

and $F_{\{b,f\}} = R_{\{b,f\}} \circ W \circ S$ with a slight abuse of notations. Here S and W denote the flow estimation and spatial warping modules, respectively, and $R_{\{b,f\}}$ denotes a stack of residual blocks.

Aggregation and Upsampling. BasicVSR adopts basic components for aggregation and upsampling. Specifically, given the intermediate features $h_i^{\{b,f\}}$, an upsampling module composed of multiple convolutions and pixel-shuffle [27] is used to generate the output HR images:

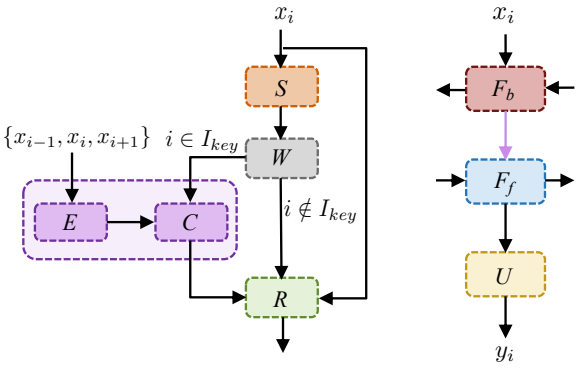
$$y_i = U(h_i^f, h_i^b), \tag{3}$$

where U denotes the upsampling module.

Summary of BasicVSR. The analysis above motivates the design choice of BasicVSR. For propagation, BasicVSR has chosen bidirectional propagation with emphasis on long-term and global propagation. For alignment, BasicVSR adopts a simple flow-based alignment but taking place at feature level. For aggregation and upsampling, popular choices on feature concatenation and pixel-shuffle suffice. Despite being a simple and succinct method, BasicVSR achieves great performance in both restoration quality and efficiency. BasicVSR is also highly versatile as it can readily accommodate additional components to handle more challenging scenarios, as we show next.

3.2. From BasicVSR to IconVSR

Using BasicVSR as a backbone, we introduce two novel components – *Information-refill mechanism* and ***co**upled propagation* (**IconVSR**), to mitigate error accumulation during propagation and to facilitate information aggregation.



(a) Information-Refill (b) Coupled Propagation

Figure 5. **(a)** An additional feature extractor is used for feature refinement, alleviating the error accumulation during propagation. I_{key} denotes the set of indices of the selected keyframes. E and C denote the feature extractor and convolution, respectively. **(b)** The inter-connected propagation branches facilitate the information exchange by passing the outputs of the backward branches to the forward branches. The proposed components are colored in purple.

Information-Refill. Inaccurate alignment in occluded regions and on image boundaries is a prominent challenge that can lead to error accumulation, especially if we adopt long-term propagation in our framework. To alleviate undesirable effects brought by such erroneous features, we propose an information-refill mechanism for feature refinement.

As shown in Fig. 5(a), an additional feature extractor is used to extract deep features from a subset of input frames (keyframes) and their respective neighbors. The extracted features are then fused with the aligned features $\bar{h}_i$ (Eq. 2) by a convolution:

$$\begin{aligned} e_i &= E(x_{i-1}, x_i, x_{i+1}), \\ \hat{h}_i^{\{b,f\}} &= \begin{cases} C\left(e_i, \bar{h}_i^{\{b,f\}}\right) & \text{if } i \in I_{key}, \\ \bar{h}_i^{\{b,f\}} & \text{otherwise,} \end{cases} \end{aligned} \tag{4}$$

where E and C correspond to the feature extractor and convolution, respectively. I_{key} denotes the set of indices of the selected keyframes. The refined features are then passed to the residual blocks for further refinement:

$$h_i^{\{b,f\}} = R_{\{b,f\}}(x_i, \hat{h}_i^{\{b,f\}}). \tag{5}$$

It is noteworthy that the feature extractor and feature fusion are applied to the sparsely-selected keyframes only. Hence, the computational burden brought by the information-refill mechanism is insignificant.

While information-refill inherits the idea of keyframes, we remark here that unlike existing interval-based methods [15, 35] that isolate the intervals for independent

processing, the intervals (separated by the keyframes) in IconVSR are connected to maintain a global information propagation.

Coupled Propagation. In bidirectional settings, features are typically propagated in two opposite directions independently. In this design, the features in each propagation branch are computed based on partial information, from either previous frames or future frames. To exploit the information in the sequences, we propose a coupled propagation scheme, where the propagation modules are interconnected. As depicted in Fig. 5(b), in coupled propagation, the features propagated backward h_i^b are taken as inputs in the forward propagation module (*c.f.* Eq. 1, 3):

$$\begin{aligned} h_i^b &= F_b(x_i, x_{i+1}, h_{i+1}^b), \\ h_i^f &= F_f(x_i, x_{i-1}, h_i^b, h_{i-1}^f), \\ y_i &= U(h_i^f). \end{aligned} \tag{6}$$

With coupled propagation, the forward propagation branch receives information from both past and future frames, leading to features of higher quality and hence better outputs. More importantly, since coupled propagation requires only changes of the branch connections, the performance gain can be obtained without introducing computational overhead.

4. Experiments

Datasets and Settings We consider two widely-used datasets for training: REDS [23] and Vimeo-90K [33]. For REDS, following [32], we use the REDS4 dataset[2] as our test set. We additionally define REDSval4[3] as our validation set. The remaining clips are used for training. We use Vid4 [21], UDM10 [34], and Vimeo-90K-T [33] as test sets along with Vimeo-90K. We test our models with $4\times$ downsampling using two degradations – Bicubic (BI) and Blur Downsampling (BD).

We use pre-trained SPyNet [24] and EDVR-M[4] [32] as our flow estimation module and feature extractor, respectively. We adopt Adam optimizer [18] and Cosine Annealing scheme [22]. The initial learning rates of the feature extractor and flow estimator are set to 1×10^{-4} and 2.5×10^{-5}, respectively. The learning rate for all other modules is set to 2×10^{-4}. The total number of iterations is 300K, and the weights of the feature extractor and flow estimator are fixed during the first 5,000 iterations. The batch size is 8 and the patch size of input LR frames is 64×64. We use Charbonnier loss [3] since it better handles outliers and improves the performance over the conventional ℓ_2 loss [19]. Detailed experimental settings are provided in the appendix.

[2]Clips 000, 011, 015, 020 of REDS training set.
[3]Clips 000, 001, 006, 017 of REDS validation set.
[4]A lightweight version of EDVR.

4.1. Comparisons with State-of-the-Art Methods

We conduct comprehensive experiments by comparing BasicVSR and IconVSR with 14 models: VESPCN [1], SPMC [29], TOFlow [33], FRVSR [25], DUF [16], RBPN [9], EDVR-M [32], EDVR [32], MuCAN [20], PFNL [34], RLSP [8], TGA [13], RSDN [12], and RRN [14]. The quantitative results are summarized in Table 2 and the speed and performance comparison is provided in Fig. 1. Note that the parameters of BasicVSR and IconVSR are inclusive of that in the optical flow network, SPyNet. So the comparison is fair.

BasicVSR. BasicVSR outperforms existing state of the arts on various datasets, including REDS4, UDM10, and Vid4. BasicVSR also demonstrates high efficiency in addition to improvements in restoration quality. As shown in Fig. 1, BasicVSR surpasses RSDN [12] by 0.61 dB on UDM10 while having a similar number of parameters. When compared with EDVR [32], which has a significantly larger complexity, BasicVSR obtains a marked improvement of 0.33 dB on REDS4 and competitive performances on Vimeo-90K-T and Vid4. We note that the performance of BasicVSR on Vimeo-90K-T is slightly lower than that achieved by sliding-window methods such as EDVR [32] and TGA [13]. This is expected since Vimeo-90K-T contains sequences with only seven frames, while the success of BasicVSR partially comes from the aggregation of long-term information (which is a realistic assumption).

IconVSR. IconVSR further improves the performance by up to 0.31 dB over BasicVSR with slightly longer runtime. The performance gain is especially obvious in Vimeo-90K-T and REDS4, showing that our proposed coupled propagation and information-refill mechanisms are beneficial in videos (1) lacking long-term information (Vimeo-90K-T) and (2) containing large and complicated motions (REDS4). Overall, both BasicVSR and IconVSR are able to achieve remarkable performance while being faster than most state of the arts.

Qualitative comparisons are shown in Figures 11 and 12. BasicVSR and IconVSR are able to recover finer details and sharper edges. For instance, only BasicVSR and IconVSR successfully recover clear square patterns in Fig. 11 and the vertical strip patterns in Fig. 12. With the proposed components, IconVSR is able to reconstruct images with sharper edges. More examples are provided in the appendix.

5. Ablation Studies

5.1. From BasicVSR to IconVSR

Information-Refill. We qualitatively visualize the features before and after information-refill to gain insights into the mechanism. As shown in Fig. 8(a), before information-refill, the boundary pixels in the warped feature essentially

Table 2. **Quantitative comparison (PSNR/SSIM).** All results are calculated on Y-channel except REDS4 [23] (RGB-channel). **Red** and blue colors indicate the best and the second-best performance, respectively. Blanked entries correspond to results unable to be reported. The runtime is computed on an LR size of 180×320.

			BI degradation			BD degradation		
	Params (M)	Runtime (ms)	REDS4 [23]	Vimeo-90K-T [33]	Vid4 [21]	UDM10 [34]	Vimeo-90K-T [33]	Vid4 [21]
Bicubic	-	-	26.14/0.7292	31.32/0.8684	23.78/0.6347	28.47/0.8253	31.30/0.8687	21.80/0.5246
VESPCN [1]	-	-	-	-	25.35/0.7557	-	-	-
SPMC [29]	-	-	-	-	25.88/0.7752	-	-	-
TOFlow [33]	-	-	27.98/0.7990	33.08/0.9054	25.89/0.7651	36.26/0.9438	34.62/0.9212	-
FRVSR [25]	5.1	137	-	-	-	37.09/0.9522	35.64/0.9319	26.69/0.8103
DUF [16]	5.8	974	28.63/0.8251	-	-	38.48/0.9605	36.87/0.9447	27.38/0.8329
RBPN [9]	12.2	1507	30.09/0.8590	37.07/0.9435	27.12/0.8180	38.66/0.9596	37.20/0.9458	-
EDVR-M [32]	3.3	118	30.53/0.8699	37.09/0.9446	27.10/0.8186	39.40/0.9663	37.33/0.9484	27.45/0.8406
EDVR [32]	20.6	378	31.09/0.8800	**37.61/0.9489**	27.35/0.8264	39.89/0.9686	37.81/0.9523	27.85/0.8503
PFNL [34]	3.0	295	29.63/0.8502	36.14/0.9363	26.73/0.8029	38.74/0.9627	-	27.16/0.8355
MuCAN [20]	-	-	30.88/0.8750	37.32/0.9465	-	-	-	-
TGA [13]	5.8	-	-	-	-	-	37.59/0.9516	27.63/0.8423
RLSP [8]	4.2	49	-	-	-	38.48/0.9606	36.49/0.9403	27.48/0.8388
RSDN [12]	6.2	94	-	-	-	39.35/0.9653	37.23/0.9471	27.92/0.8505
RRN [14]	3.4	45	-	-	-	38.96/0.9644	-	27.69/0.8488
BasicVSR (ours)	6.3	63	31.42/0.8909	37.18/0.9450	27.24/0.8251	39.96/**0.9694**	37.53/0.9498	27.96/0.8553
IconVSR (ours)	8.7	70	**31.67/0.8948**	37.47/0.9476	**27.39/0.8279**	**40.03/0.9694**	**37.84/0.9524**	**28.04/0.8570**

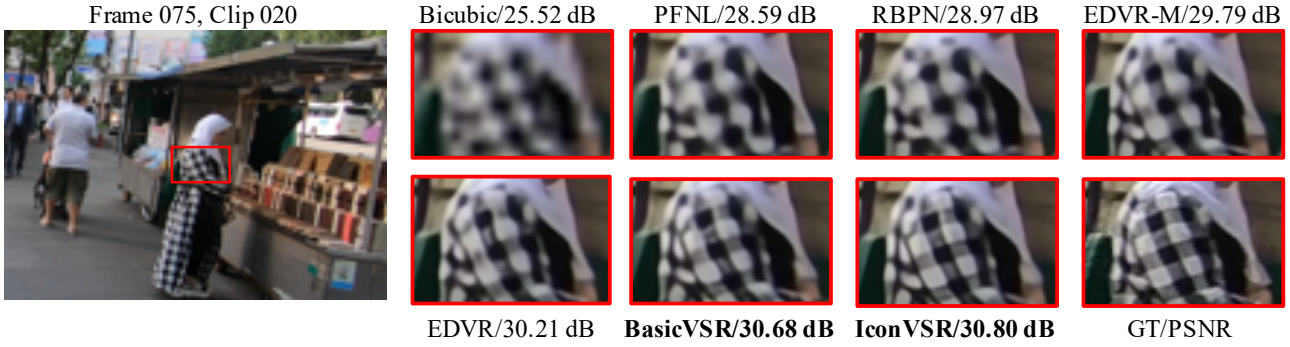


Figure 6. **Qualitative comparison on REDS4 [23].** BasicVSR and IconVSR restores clearer square patterns. IconVSR restores sharper edges. **(Zoom-in for best view)**

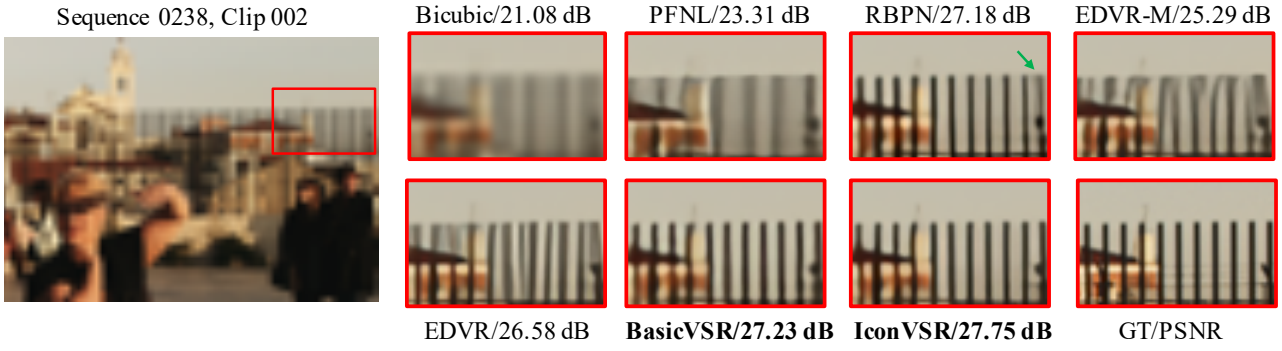
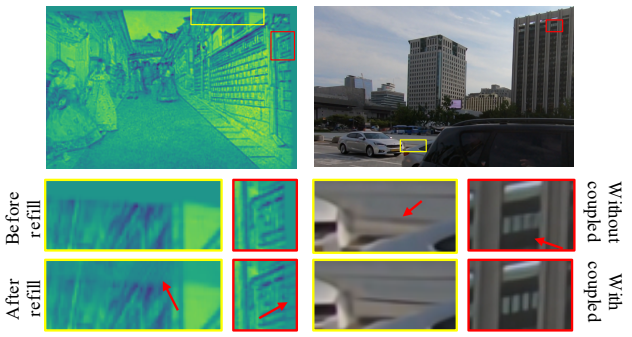


Figure 7. **Qualitative comparison on Vimeo-90K-T [33].** Only BasicVSR and IconVSR are able to recover the vertical strip patterns. IconVSR restores sharper edges. **(Zoom-in for best view)**

become zero due to non-existing correspondences. The lost information inevitably worsens the feature quality, leading to degraded outputs. With our information-refill mechanism, the additional features can be used to "refill" the lost information in regions where the features are poorly aligned. The retrieved information can then be employed for the subsequent feature refinement and propagation.

The above effect is especially obvious in regions with



(a) Information-Refill (b) Coupled Propagation

Figure 8. **(a)** Information lost during spatial warping can be compensated by the additional features. **(b)** With more effective use of the backward-propagated features, coupled propagation leads to clearer details and finer edges, especially in regions that are occluded in previous frames and regions that exist in the whole sequence. **(Zoom-in for best view)**

Figure 9. **Effect of Information-Refill.** The contribution of information-refill is more obvious in regions with fine details, where alignment is error-prone. The information from the additional feature extractor leads to marked improvements.

fine details. In those regions, information from neighboring frames cannot be effectively aggregated due to alignment error, often resulting in inferior quality. With information-refill, the additional features assist in the restoration of the details, leading to in improved quality. For example, as shown in Fig. 9, the license plate number can be reconstructed more clearly with the refill mechanism.

Coupled Propagation. To ablate the coupled propagation scheme, we disable the information-refill mechanism and compare IconVSR with BasicVSR. In Fig. 8(b), the yellow box represents a region occluded in previous frames, and the forward propagation branch in BasicVSR could not receive information of that region. The red box denotes a region that exists in all frames of the sequence, and hence abundant "snapshots" of the region can be found in latter frames. With coupled propagation, the backward-propagated features are employed more effectively, and hence more details and finer edges can be reconstructed. The PSNR improvement over BasicVSR is summarized in Table 3.

5.2. Tradeoff in IconVSR

Although IconVSR is trained with a fixed keyframe interval, one can reduce the number of keyframes for faster inference. The PSNR using different numbers of keyframes

Table 3. **Evaluations of IconVSR components.** The two components bring an improvement of up to 0.28 dB over BasicVSR. The PSNR is computed on REDS4/REDSval4.

	BasicVSR	IconVSR (w/o refill)	IconVSR
Info-Refill	✗	✗	✓
Coupled-Prop	✗	✓	✓
PSNR	31.42/30.17	31.60/30.38	31.67/30.45

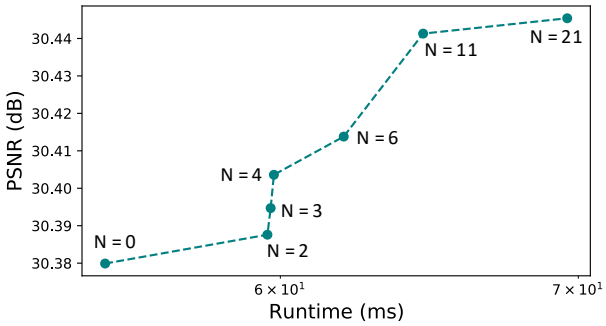


Figure 10. **Tradeoff in IconVSR.** One can reduce the number of keyframes for faster inference. The PSNR is positively correlated with the number of keyframes N, indicating the effectiveness of the information-refill mechanism. The PSNR is calculated on REDSval4. The total number of frames in each clip is 100, and the keyframes are evenly spaced.

is depicted in Fig. 10, where we see that the PSNR is positively correlated with the number of keyframes, verifying the contributions of the information-refill mechanism. In an extreme case when there is no keyframe, IconVSR degenerates to a recurrent network. Nevertheless, it still achieves a PSNR of 30.38 dB on REDSval4, which is 0.21 dB higher than BasicVSR. This demonstrates the effectiveness of our coupled propagation scheme, which can be used without introducing additional computational overhead.

6. Conclusion

This work devotes attention to the search of generic and efficient VSR baselines to ease the analysis and extension of VSR approaches. Through decomposing and analyzing existing elements, we propose BasicVSR, a simple yet effective network that outperforms existing state of the arts with high efficiency. We build upon BasicVSR and propose IconVSR with two novel components to further improve the performance. BasicVSR and IconVSR can serve as strong baselines for future works, and the discovery on the architecture designs could potentially be extended to other low-level vision tasks, such as video deblurring, denoising and colorization.

Acknowledgement. This research was conducted in collaboration with SenseTime and supported by the Singapore Government through the Industry Alignment Fund - Industry Collaboration Projects Grant. It is also partially supported by Singapore MOE AcRF Tier 1 (2018-T1-002-056) and NTU SUG.

References

[1] Jose Caballero, Christian Ledig, Aitken Andrew, Acosta Alejandro, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*, 2017. 2, 6, 7

[2] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. *arXiv preprint arXiv:2009.07265*. 4

[3] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 6, 10

[4] Kai Chen, Jiaqi Wang, Shuo Yang, Xingcheng Zhang, Yuanjun Xiong, Chen Change Loy, and Dahua Lin. Optimizing video object detection via a scale-time lattice. In *CVPR*, 2018. 2

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

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

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2016. 2

[8] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCVW*, 2019. 3, 4, 6, 7

[9] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019. 1, 2, 3, 6, 7

[10] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *NeurIPS*, 2015. 2, 4

[11] Yan Huang, Wei Wang, and Liang Wang. Video super-resolution via bidirectional recurrent convolutional networks. *TPAMI*, 2018. 2, 4

[12] 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, 3, 4, 6, 7, 10

[13] 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, 3, 6, 7

[14] Takashi Isobe, Fang Zhu, and Shengjin Wang. Revisiting temporal modeling for video super-resolution. In *BMVC*, 2020. 2, 3, 4, 6, 7

[15] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR*, 2019. 2, 5

[16] 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, 6, 7

[17] Tae Hyun Kim, Mehdi S M Sajjadi, Michael Hirsch, and Bernhard Schölkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, 2018. 4, 5

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6, 10

[19] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 5835–5843, 2017. 6, 10

[20] 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, 6, 7

[21] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *TPAMI*, 2014. 2, 6, 7, 10, 13

[22] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6, 10

[23] 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. 6, 7, 10, 11

[24] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 6, 10

[25] Mehdi S M Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, 2018. 2, 3, 5, 6, 7, 10

[26] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *ECCV*, 2016. 2

[27] Wenzhe Shi, Jose Caballero, Ferenc Huszár, 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*, pages 1874–1883, 2016. 2, 5

[28] Hiroyuki Takeda, Peyman Milanfar, Matan Protter, and Michael Elad. Super-resolution without explicit subpixel motion estimation. *TIP*, 18(9):1958–1975, 2009. 2

[29] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *CVPR*, 2017. 2, 6, 7

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

[31] Hua Wang, Dewei Su, Chuangchuang Liu, Longcun Jin, Xianfang Sun, and Xinyi Peng. Deformable non-local network for video super-resolution. *IEEE Access*, 2019. 2

[32] 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. 1, 2, 3, 6, 7, 10

[33] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019. 2, 4, 5, 6, 7, 10, 12

[34] 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. 1, 2, 6, 7, 10, 13

[35] Zhengdong Zhang and Vivienne Sze. FAST: A framework to accelerate super-resolution processing on compressed videos. In *CVPRW*, 2017. 2, 5

[36] Xizhou Zhu, Jifeng Dai, Xingchi Zhu, Yichen Wei, and Lu Yuan. Towards high performance video object detection for mobiles. *arXiv preprint arXiv:1804.05830*, 2018. 2

[37] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 2

[38] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. 2

[39] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 2

Appendix

A. Architecture and Experimental Settings

Architecture. In all our models, we adopt SPyNet [24] as our flow estimator because of its simplicity and efficiency. We use 30 residual blocks in each propagation branch. The feature channel is set to 64. In IconVSR, we adopt EDVR-M[5] [32] as the additional feature extractor since it maintains a good balance between efficiency and quality. The complexity of the components are summarized in Table 4. BasicVSR and IconVSR share the same flow estimator and main network. The main network is a lightweight network, consisting of only 4.9M parameters. The flow estimator and feature extractor are fine-tuned together with the main network. In all our experiments, every five frames are selected as keyframes. Note that the feature extractor is applied to keyframes only. Therefore, the computational burden brought by it is insignificant.

Table 4. **Model complexity of BasicVSR and IconVSR.**

	BasicVSR	IconVSR
Flow Estimator	1.4M	1.4M
Main Network	4.9M	4.9M
Feature Extractor	-	2.4M
Total	6.3M	8.7M

Datasets. We consider two widely-used datasets for training: REDS [23] and Vimeo-90K [33]. For REDS, following [32], we use the REDS4 dataset[6] as our test set. We additionally define REDSval4[7] as our validation set. The remaining clips are used for training. We use Vid4 [21], UDM10 [34], and Vimeo-90K-T [33] as test sets along with Vimeo-90K.

Experimental Settings. When training on REDS, we use a sequence of 15 frames as inputs, and loss is computed for the 15 output images. When training on Vimeo-90K, we temporally augment the sequence by flipping the original input sequence to allow longer propagation. In other words, we train with a sequence of 14 frames. During inference, we take the whole video sequence as input.

We adopt Adam optimizer [18] and Cosine Annealing scheme [22]. The initial learning rates of the feature extractor and flow estimator are set to 1×10^{-4} and 2.5×10^{-5}, respectively. The learning rate for all other modules is set to 2×10^{-4}. The total number of iterations is 300K, and the weights of the feature extractor and flow estimator are fixed during the first 5,000 iterations. The batch size is 8 and the patch size of input LR frames is 64×64.

Loss Function. We use Charbonnier loss [3] since it better handles outliers and improves the performance over the conventional ℓ_2 loss [19]:

$$\mathcal{L} = \frac{1}{N}\sum_{i=0}^{N}\rho(y_i - z_i), \tag{7}$$

where $\rho(x) = \sqrt{x^2+\epsilon^2}$, $\epsilon{=}1\times10^{-8}$, z_i denotes the ground-truth HR frame, and N denotes to the number of pixels.

Degradations. We train and test our models with $4\times$ downsampling using two degradations – Bicubic (BI) and Blur Downsampling (BD) [12, 25]. For BI, we use the MATLAB function `imresize` for downsampling. For BD, we blur the ground-truths by a Gaussian filter with $\sigma{=}1.6$, followed by a subsampling every four pixels.

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

B. Qualitative Results

B.1. Comparison with State of the Arts

In this section, we provide additional qualitative comparisons on REDS4 [23], Vimeo-90K [33], Vid4 [21], and UDM10 [34]. In Fig. 11 to Fig. 14, it is observed that BasicVSR and IconVSR successfully produce outputs with finer details and sharper edges. Furthermore, with the proposed information-refill and coupled propagation, IconVSR further improves the quality of the outputs.

B.2. BasicVSR vs IconVSR

In Fig. 15, we provide additional visual comparison of BasicVSR and IconVSR to demonstrate the effectiveness of our proposed components. We see that (1) information-refill improves the output quality on the fine regions, where alignment is error-prone, and (2) coupled propagation leads to sharper edges by better employing the long-term information in the sequence.

[5] A lightweight version of EDVR.
[6] Clips 000, 011, 015, 020 of REDS training set.
[7] Clips 000, 001, 006, 017 of REDS validation set.

Figure 11. Qualitative comparison on REDS [23].

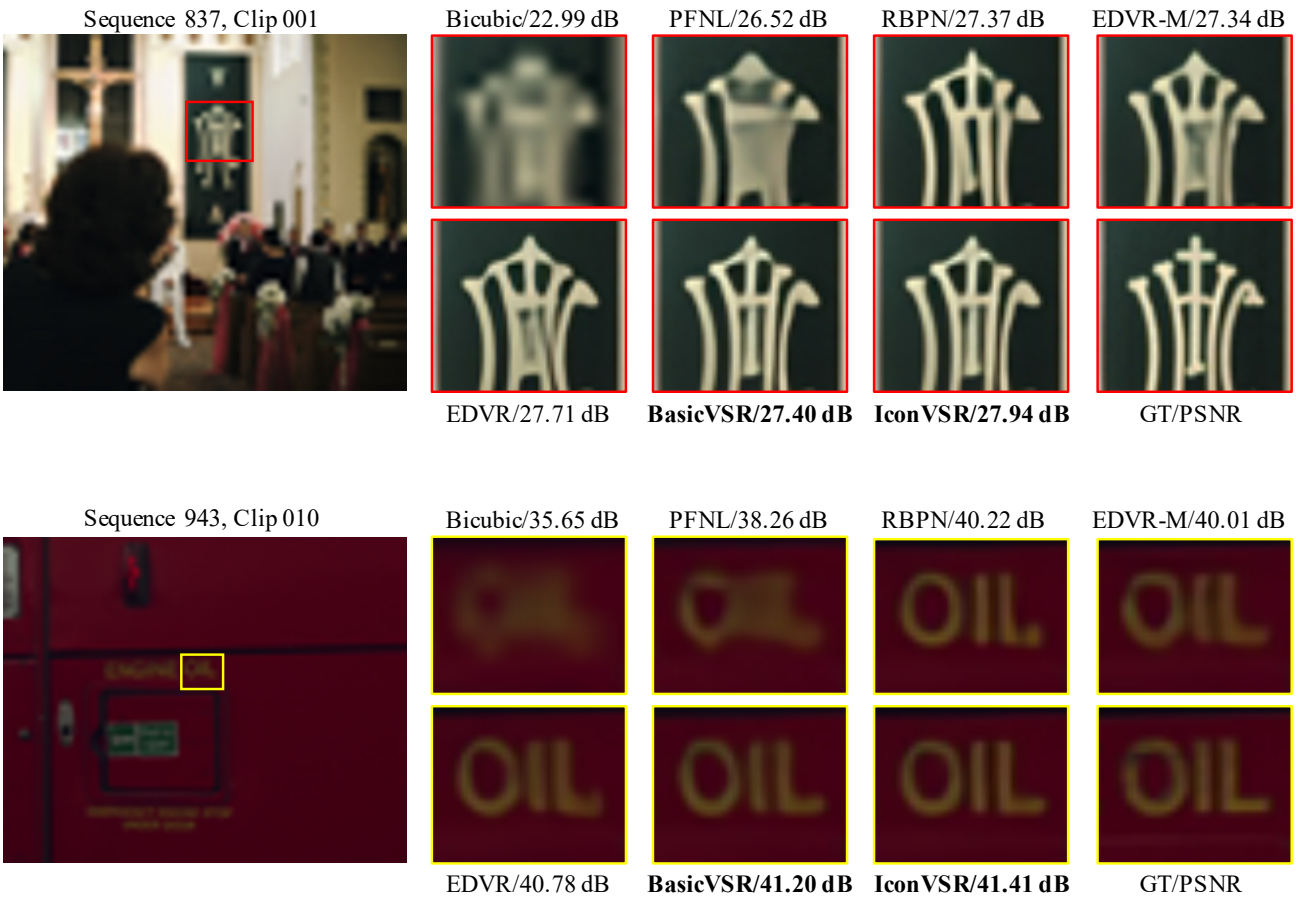


Figure 12. Qualitative comparison on Vimeo-90K [33].

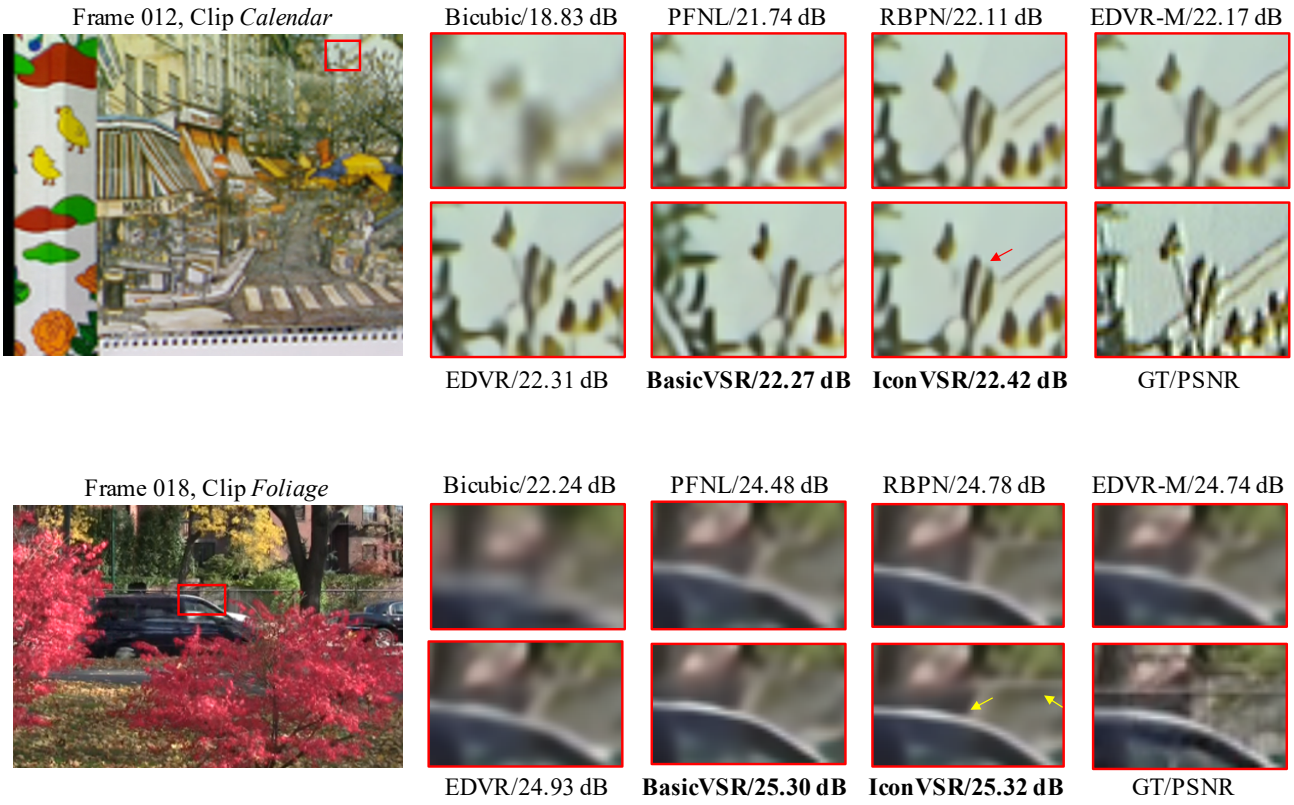


Figure 13. Qualitative comparison on Vid4 [21].

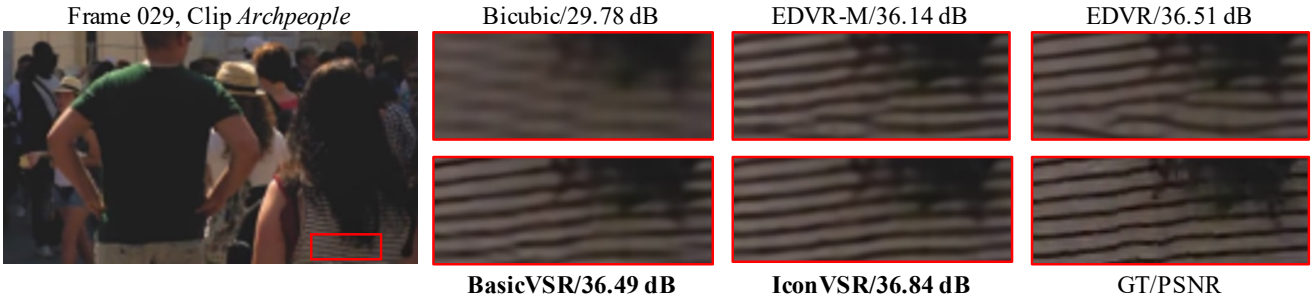
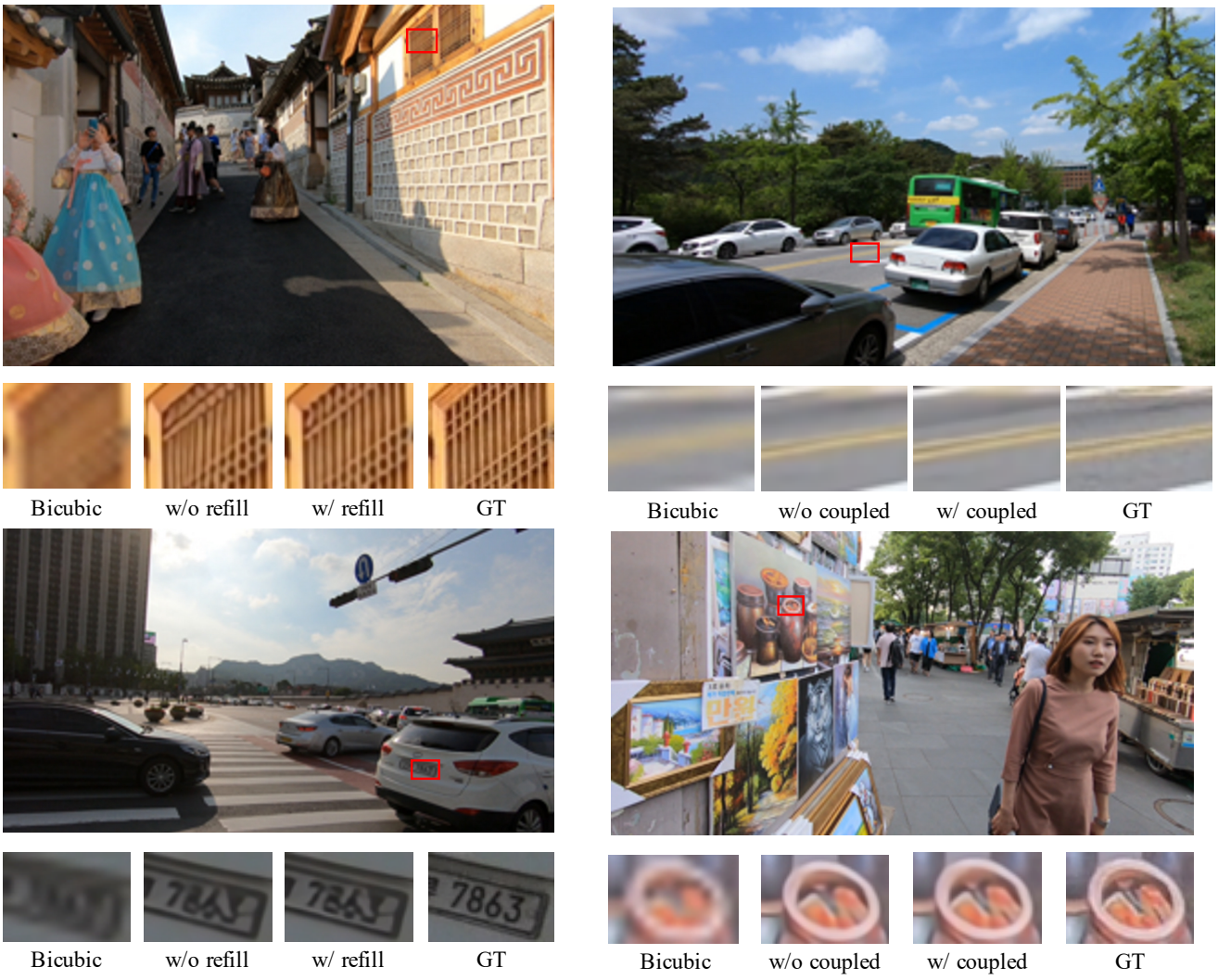


Figure 14. Qualitative comparison on UDM10 [34].



(a) Information-Refill

(b) Coupled Propagation

Figure 15. **Ablation of IconVSR.** With *information-refill* and *coupled propagation*, IconVSR produces outputs with details and sharper edges.