Temporal Kernel Consistency for Blind Video Super-Resolution

Lichuan Xiang[1*], Royson Lee[2*], Mohamed S. Abdelfattah[3],
Nicholas D. Lane[2,3], Hongkai Wen[1,3]
[1]University of Warwick [2]University of Cambridge [3]Samsung AI Center, Cambridge
l.xiang.2@warwick.ac.uk

Abstract

Deep learning-based blind super-resolution (SR) methods have recently achieved unprecedented performance in upscaling frames with unknown degradation. These models are able to accurately estimate the unknown downscaling kernel from a given low-resolution (LR) image in order to leverage the kernel during restoration. Although these approaches have largely been successful, they are predominantly image-based and therefore do not exploit the temporal properties of the kernels across multiple video frames. In this paper, we investigated the temporal properties of the kernels and highlighted its importance in the task of blind video super-resolution. Specifically, we measured the kernel temporal consistency of real-world videos and illustrated how the estimated kernels might change per frame in videos of varying dynamicity of the scene and its objects. With this new insight, we revisited previous popular video SR approaches, and showed that previous assumptions of using a fixed kernel throughout the restoration process can lead to visual artifacts when upscaling real-world videos. In order to counteract this, we tailored existing single-image and video SR techniques to leverage kernel consistency during both kernel estimation and video upscaling processes. Extensive experiments on synthetic and real-world videos show substantial restoration gains quantitatively and qualitatively, achieving the new state-of-the-art in blind video SR and underlining the potential of exploiting kernel temporal consistency.

1. Introduction

Super-resolution (SR) is an ill-posed problem that assumes the low-resolution image (LR) is derived from a high-resolution (HR) image and is recently dominated by deep learning due to its unprecedented performance [6]. In order to better restore the high-frequency details, state-of-the-art video SR methods [35, 36, 37] exploit the temporal frame information by employing a multi-frame SR (MFSR) approach. Specifically, each supporting frame is aligned with its reference frame through motion compensation before the information in these frames are merged for upscaling.

Most of these methods, however, assume that the degradation process, applying the blur kernel and the downscaling operation, is pre-defined. Therefore, the performance of these methods significantly deteriorates for real-world videos as the downscaling kernel, which is used for upscaling, differs from the ground truth kernel, a phenomenon known as the kernel mismatch problem [8]. Although there has been significant progress to enable the usage of SR models in real-world applications, these solutions are predominantly image-based [3, 12, 22, 42]. The primary paradigm of these blind image-based solutions consists of either a two-step or an end-to-end process, starting with a kernel estimation module and followed by a SR model that aims to maximize image quality given the estimated kernel and/or noise. Hence, when upscaling videos, these works do not utilize the temporal similarity between kernels and have to estimate kernels individually per frame. This is not only computationally expensive but also less effective, since estimating kernels independently per-frame may result in inaccurate kernels, as shown in Sec. 4 and Sec. 5, and thus kernel mismatch.

Recent blind MFSR approaches, on the other hand, utilized a fixed kernel to upscale every frame [20, 29] in the same video – we hypothesize that this fixed kernel assumption can also lead to kernel mismatch. Therefore, in this work, we attempt to investigate and answer the following questions: *how does the kernel change temporally in real-world videos, and how can we leverage this change in the video restoration process?*

Towards our goal, we first investigated the temporal differences in kernels in Sec. 4. In particular, we used a recent image-based kernel estimation approach, KernelGAN [3], on frames of real-world videos and observed that videos of varying dynamicity, such as scene changes and object motion blurs, can result in corresponding variations in the

[*] Equal contributions.

downsampling kernels. We then show how videos of different dynamicity can affect the temporal consistency of their downscaling kernels. From this perspective, we re-evaluated previous MFSR approaches on real-world videos in Sec. 5. Through our experiments, we show that the common assumption of using a fixed downsampling kernel for multi-frame approaches can lead to the kernel mismatch problem, resulting in inaccurate motion compensation and hence inferior restoration results. To counteract these drawbacks, we tailored these existing techniques to exploit our new insight on *kernel temporal consistency* in Sec. 6, leading to substantial gains as compared to state-of-the-art. In summary, the main contributions of this work are:

- To the best of our knowledge, we are the first to investigate the temporal consistency of kernels in real-world videos for deep blind video SR.
- We present the limitations and drawbacks of using a fixed kernel, a scenario that is commonly assumed, for multi-frame SR approaches.
- Through tailored alterations to existing SR approaches, we underline the potential of exploiting *kernel temporal consistency* for accurate kernel estimation and motion compensation, resulting in considerable performance gains in video restoration.

2. Related Work

Single-Image Blind Super-Resolution. Previous deep learning-based image-based SR approaches [6, 7, 30, 18, 32, 1, 30] assumed a fixed and ideal downsampling degradation process, often bicubic interpolation, leading to poor performance when applied to real-world images. As a result, most blind SR approaches focused on estimating the downsampling kernel and/or utilizing it for upsampling. Efrat *et al.* [8] first highlighted the kernel mismatch problem: using the incorrect kernel during restoration had a significant impact on the performance regardless of the choice of image prior. Towards accurate downsampling kernel estimation, Michaeli *et al.* [25] exploited the inherent recurrence property of image patches and proposed an iterative algorithm to derive the kernel that maximizes the similarity of recurring patches across scales of the LR image. Bell-Kligler *et al.* [3] adopted a GAN approach [10], in which the generator learnt an estimated kernel to downscale the input image with and the discriminator learnt to differentiate between the patch distribution of the input image and its downscaled variant. The downsampling kernel can also be learned using CNNs by enforcing that the super-resolved image maps back to the LR image [13] or using a paired of real-world image dataset [4]. Exploiting the kernel mismatch phenomenon, Gu *et al.* [12] and Luo *et al.* [22] alternatively estimated the kernel from the approximated super-resolved image and restored the image by using the estimated kernel, reaching the current state-of-the-art.

Multi-Frame Super-Resolution. MFSR approaches focus on utilizing temporal information from the LR frames by aligning and fusing them in order to further boost restoration performance through CNNs or RNNs. Earlier works [19, 17] performed motion compensation by estimating optical flow using traditional off-the-shelf motion estimation algorithms [2]. As the accuracy of motion estimation directly affects the reconstruction quality of the super-resolved images, these traditional motion estimation works are superseded by more accurate CNN-based networks such as spatial transformer networks [15] or task-specific motion estimation networks [14, 27, 33], leading to approaches [21, 24, 34, 38, 28] that focused on integrating motion estimation and SR networks for end-to-end learning. Recent works [16, 35, 36, 37] decoupled this dependency on motion estimation networks and performed motion compensation by adaptively aligning the reference and supporting frames through dynamically-generated filters or deformable convolutions [5, 43]. Although majority of these works helped to elucidate the relationship between motion estimation and video restoration, they neglected the degradation process by assuming a fixed known kernel. Therefore, unlike previous MFSR works that focused on incorporating temporal information in the frames, we also utilized the temporal information in the downscaling degradation operation in order to further boost restoration performance.

Towards blind MFSR, Pan *et al.* [29] used a kernel estimation network, consisting of two fully-connected layers, to learn a fixed blur kernel for inference. However, they, similar to Liu *et al.* [20], assumed that the kernel is fixed at every timestamp, resulting in poor SR performance as shown in Sec. 5.

3. Problem Formulation

Multi-frame Super Resolution (MFSR) uses a set of $2N$ supporting LR frames $\{y_{t-N}, \cdots, y_{t-1}, y_{t+1}, \cdots, y_{t+N}\}$ to upscale the reference LR frame y_t at time t, utilizing temporal information across frames. The degradation process is usually expressed as follows:

$$y_{t+i} = \big((F_{t\rightarrow t+i}x_t) * k_{t+i}\big)\downarrow_s + n_{t+i} \qquad (1)$$

where y and x are the LR and HR image respectively, k is the blur kernel, $\downarrow_s$ is the downscaling operation (*e.g.* subsampling) using scaling factor s, n is the additive noise, $i = -N, \cdots, N$, and F is the warping matrix w.r.t the optical flow applied on x_t. The image warping process can either be done explicitly via an optical flow or implicitly via dynamically-generated filters [16] or deformable convolutions [5]. The process of applying k together with the $\downarrow_s$ is also referred to as applying a downscaling kernel or SR kernel [3, 12]. Traditionally, a prior term is individually handcrafted for x_t, k_{t+i} and $F_{t\to t+i}$, but most deep learning-based approaches capture the prior [6] through CNNs by training it using a large amount of examples.

In order to solve for k and x, state-of-the-art blind image-based algorithms split the problem into two sub-problems, estimating k and restoring x, and address each problem sequentially [3, 12] or alternately [39, 22]. MFSR solutions, on the other hand, include an additional sub-problem of estimating the motion between each supporting frame and its reference frame in order to perform motion compensation and hence leverage the temporal frame information during restoration. Although previous traditional video SR approaches [9, 23] assume that the kernel varies across frames, recent works [20, 29] assume a fixed kernel. In our work, we study and highlight the implications of both assumptions and advocate for the per-frame kernel approach, resulting in the following optimization problem:

$$\begin{aligned}
\hat{x}_t &= \arg\min_{x_t} \sum_{i=-N}^{N} \left\| y_{t+i} - \left((F_{t\to t+i} x_t) * k_{t+i} \right) \downarrow_s \right\| \\
\hat{k}_t &= \arg\min_{k_t} \left\| y_t - (x_t * k_t) \downarrow_s \right\| \\
\hat{F}_{t\to t+i} &= \arg\min_{F_{t\to t+i}} \left\| y_{t+i} - \left((F_{t\to t+i} x_t) * k_{t+i} \right) \downarrow_s \right\|
\end{aligned} \tag{2}$$

where $\hat{x}$, $\hat{k}$, and $\hat{F}$ are the estimated HR image x, kernel k, and warping matrix F respectively.

4. Kernels In Real-World Videos

In order to investigate the temporal kernel changes in real-world videos, we extracted a pool of kernel sequences from the Something-Something dataset [11], a real-world video prediction dataset. As ground truth kernels do not exist in real-world videos, we applied the state-of-the-art image-based kernel extraction method, KernelGAN [3], to extract the sequences of kernels. Through these kernel sequences, we observed that the extracted SR kernels can often be different for each frame, while on the other hand may also exhibit certain levels of temporal consistency, depending on the video's dynamicity.

Fig. 1 illustrates this phenomena, in which we show the distributions of the magnitude of kernel changes in different video sequences. Specifically, we reshaped the extracted kernels for each frame and reduced them through principal component analysis (PCA). We then computed the sum of absolute differences between the kernel PCA components of adjacent frames and plotted this difference using videos of varying dynamicity (left and middle plot groups in Fig. 1). As a baseline, in comparison with an unrealistic real-world video without any temporal consistency, we sampled random frames from different random videos at each timestamp, of which its kernel PCA changes are represented by the right plot in Fig. 1. We observed that some videos' kernel differences, namely the left group of plots showing video sequences 13, 16 and 22 from the Something-Something dataset in Fig. 1, are of *high temporal kernel consistency*, of which the kernels remain largely unchanged throughout. In contrast, the middle group of plots represent the kernel differences of videos with *low temporal kernel consistency*, namely video sequence 2, 23 and 25, of which kernel changes can be much significant. Visually, Fig. 2 shows example frames from corresponding videos of high and low temporal kernel consistency. In particular, videos with high temporal kernel consistency depict slow and steady movements with no motion blurs or scene changes - *e.g.* a video of a hand slowly reaching towards a cup or videos with almost identical frames at each time step. On the other hand, videos with low temporal kernel consistency have motion blur caused by rapid movements of the camera or object, *e.g.* large object motions caused by a man weaving a straw hat or placing a container upright and shaky camera motions, as illustrated in the right of Fig 2. Therefore, our experiments highlight that SR kernels in real-world videos are often non-uniform and can exhibit different levels of temporal consistency.

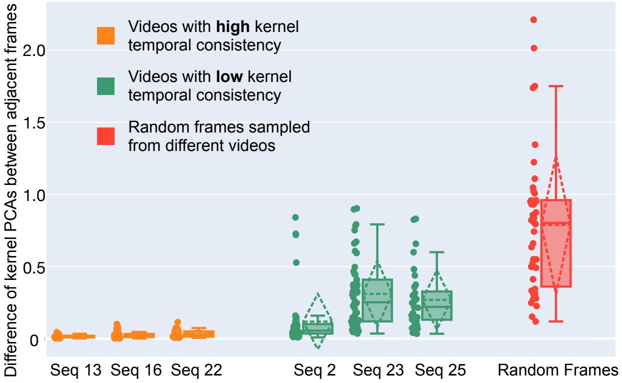

Figure 1: We quantify temporal kernel consistency by measuring kernel PCA change for adjacent frames in real-world videos with high/low kernel temporal consistency. Random frames are sampled from different videos at each timestamp as a baseline to highlight temporal kernel consistency within same video. Kernel changes are represented by solid dots while boxplots show distributions.

5. Kernel Mismatch in Previous MFSR

In order to highlight the importance of incorporating temporal kernel consistency in blind video restoration, we looked into the limitations and drawbacks of both previ-

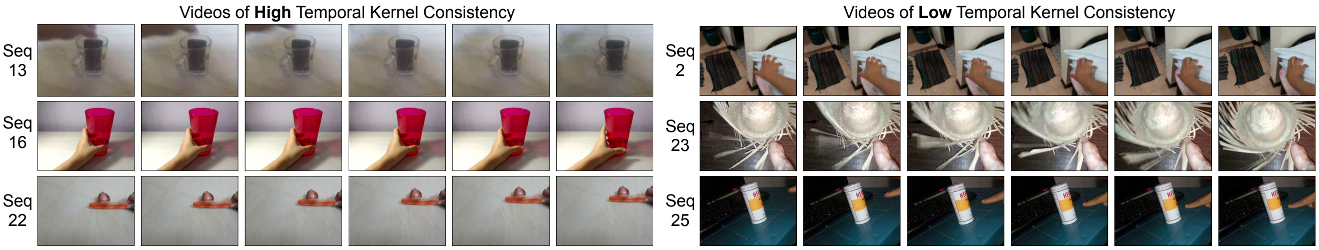
Videos of High Temporal Kernel Consistency
Seq 13
Seq 16
Seq 22
Videos of Low Temporal Kernel Consistency
Seq 2
Seq 23
Seq 25

Figure 2: Example frames from videos of high/low kernel temporal consistency as shown in Fig. 1. Videos of low kernel temporal consistency (right) contain a higher proportion of video dynamicity as compared to videos of high kernel temporal consistency (left).

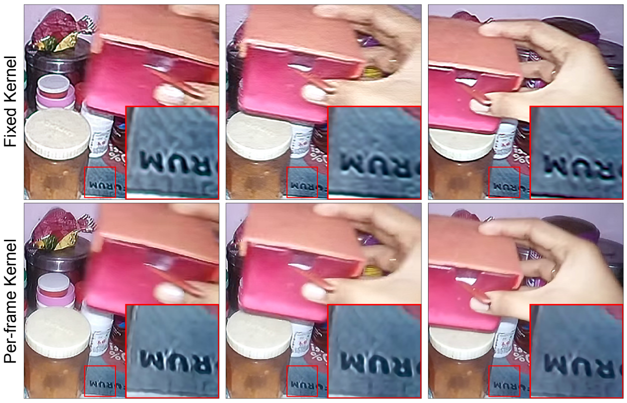

Figure 3: Examples of consecutive frames in real-world videos upscaled using a fixed kernel (top row), and different per-frame kernels (bottom row). More examples can be found in the supplementary material (s.m. Fig. 1).

ous and recent multi-frame super resolution (MFSR) approaches [19, 17, 21, 24, 34, 38, 29]. Specifically, these works assumed that either a fixed degradation operation is used for all videos or a fixed SR kernel is used to degrade all frames in each video – assumptions that do not hold for real-world videos as shown in Sec. 4. Consequently, these works suffer from the kernel mismatch phenomenon [8] when they are used to upscale real-world videos.

Impact on Frame Upscaling. Previous MFSR works exploit temporal frame information using a fixed SR kernel. We first show with a naive approach, that the use of a single kernel, even *without* utilizing temporal frame information, to restore every frame is detrimental towards the performance of frame upscaling in the video restoration process. Towards this goal, we independently computed a kernel per frame of videos taken from the Something-Something dataset using KernelGAN and restored these frames using ZSSR [31]. We compared this per-kernel approach with a single-kernel approach where we only estimated the SR kernel on the first frame in the video and used that same kernel to restore all subsequent frames in each video. Fig. 3 shows the qualitative difference between the two experiments and we observed that using a fixed kernel indeed resulted in more severe visual artifacts and unnatural textures. All experiment details and more examples can be found in the supplementary material (s.m. Sec. 1 & Fig. 3).

Impact on Motion Compensation. We then show that a fixed kernel assumption further aggravates MFSR approaches. The premise of these approaches is to utilize temporal frame information in order to boost the restoration performance. To this end, previous MFSR works used motion compensation to warp each supporting frame to its reference frame before fusing these frames together for upscaling. As mentioned in Sec. 3, the optical flow used for warping is either estimated explicitly using traditional or deep motion-estimation techniques or implicitly using adaptive filters or deformable convolutions.

In order to visualize the impact of kernel mismatch on motion compensation for real-world videos, we consider two sets of videos, one from LR sequences of the original REDS dataset [26], which are degraded using a fixed kernel, while the other from our *REDS10* testing sequence (details discussed in Sec. 6.1), which are generated using different per-frame kernels and thus better resemble the degradation characteristics of real-world videos than the former.

We then used an explicit deep motion estimation model, which is commonly used in previous MFSR approaches [21, 24, 34, 38] to compute the optical flow. Specifically, we adopt PWCNet [33] to estimate optical flow in our experiment. The optical flow is then used to warp each supporting frame and the results are shown in Fig. 4, for both fixed and per-frame degradation video sets. We observe that motion compensation performs better on the fixed degradation video set, benefiting the previous approaches that were specifically designed under the fixed kernel assumption. On the other hand, due to the kernel dynamicity in real-world videos, the warped supporting frames of those approaches often suffer from kernel mismatch when dealing with videos of varying kernels, as shown in Fig. 4 (bottom row). We further show that this phenomenon is also observed with the use of implicit motion compensation, and the errors in-

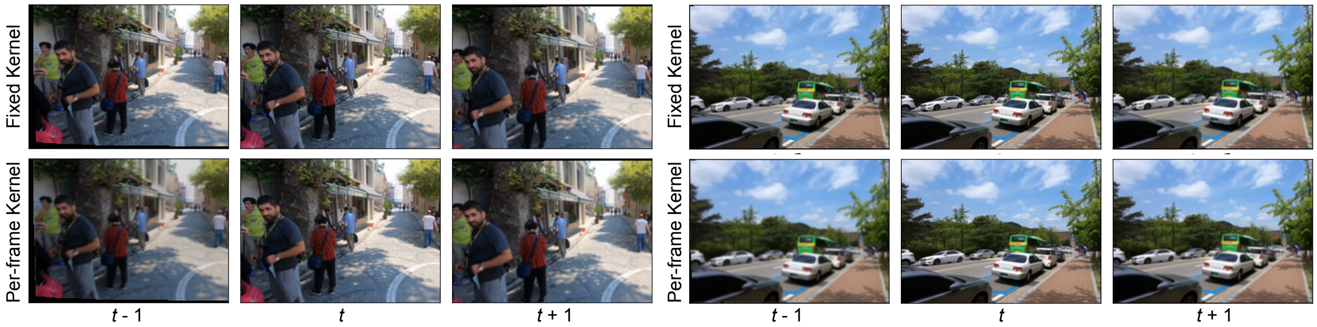

Figure 4: Example aligned frames with their reference frame at time t. Motion compensation model in current MFSR approaches performs better when considering a fixed SR kernel at every timestamp (top row), which however does not hold for real-world videos. For videos with varying kernels per frame, the aligned frames are oversmoothed and blurred (*e.g.* see the frames at t - 1 of both examples) due to kernel mismatch (bottom row). Zoom in for best results. More examples are provided in the supplementary material (s.m. Fig. 4.)

curred from inaccurate motion compensation can be propagated throughout the restoration process in Sec. 6.2.

6. Exploiting Temporal Kernel Consistency

We hypothesize that by using temporal kernel consistency, we can mitigate the limitations highlighted in Sec. 5. Towards understanding the impact of doing so, we first adopted the state-of-the-art blind image-based SR algorithm, DAN [22] and incorporated MFSR modules from EDVR [36] for temporal alignment through implicit motion compensation, fusion, and video restoration. We then tailored these approaches to exploit temporal kernel consistency and analyzed the benefits and performance impact of doing so through an ablation study.

6.1. Experiment Setup

Models. DAN [22] is an end-to-end learning approach that estimates the kernel k, and restores the image x, alternately. The key idea, as shown in black in Fig. 5, is to have two convolutional modules: 1) a *restorer* that reconstructs x given the LR image y, and the PCA of k; and 2) an *estimator* that learns the PCA of k, based on y and the resulting super-resolved image $\hat{x}$. The basic block for both components is the conditional residual block (CRB), which concatenates the basic and conditional inputs channel-wise and then exploit the inter-dependencies among feature maps through a channel attention layer [41]. The alternating algorithm executes both components iteratively, starting with an initial kernel, Dirac, and resulting in the following expression:

$$\begin{aligned} x^{(j+1)} &= \arg\min_{x} \left\| y - (x * k^{(j)}) \downarrow_s \right\|_1 \\ k^{(j+1)} &= \arg\min_{k} \left\| y - (x^{(j+1)} * k) \downarrow_s \right\|_1 \end{aligned} \tag{3}$$

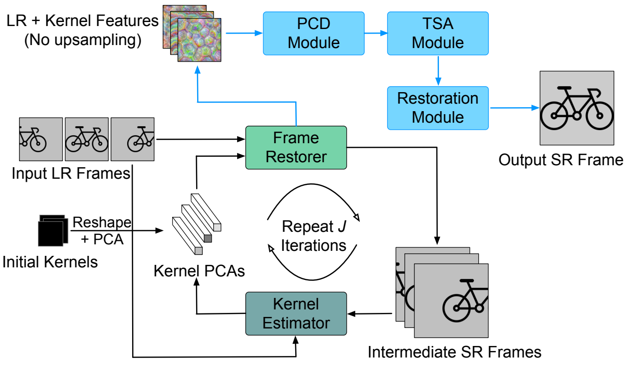

Figure 5: Our experiment setup of utilizing multiple frames for temporal kernel estimation (shown in black) and using temporal kernels for multi-frame restoration (shown in blue). See text for details and the supplementary material (s.m. Fig. 1) for a more detailed architecture diagram.

where j presents the iteration round, $j \in [1, J]$. Both components are trained using the sum of the absolute difference, L_1 loss, between k and $\hat{k}$, and between x and $\hat{x}$ estimated by the last iteration.

For multi-frame experiments, as shown in blue in Fig. 5, we used the LR feature maps at the last *restorer* iteration before upsampling and adopted EDVR's *PCD Module*, *TSA Module*, and *Restoration Module* for temporal alignment, fusion, and video restoration respectively. In other words, we merged kernel estimation and blind image restoration techniques with MFSR motion compensation methods and made alterations in order for these modules to utilize temporal kernel consistency. Further details of these modules and the architecture can be found in the supplementary material (s.m. Sec. 2 & s.m. Fig. 1).

Training Data. We combined both REDS [26] training and validation set and randomly sampled 250 for train and 10 for test. Following [3], we generated anisotropic Gaussian kernels with a size of 13×13. The lengths of both axes were uniformly sampled in (0.6, 5), and then rotated by a random angle uniformly distributed in [-π, π]. For real-world videos, we further added uniform multiplicative noise, up to 25% of each pixel value of the kernel, to the generated noise-free kernel, and normalized it to sum to one. Each frame of each HR video was degraded with a randomly generated kernel and then downsampled using bicubic interpolation to form the synthetic LR videos. Following previous works [40, 12, 22], we reshaped the kernels and reduced them through principal component analysis (PCA) before feeding into the network. We adopted this frame-wise synthesis approach for two reasons: 1) to the best of our knowledge, there is no video dataset with real-world kernels available, and extracting large amount of kernel sequences from video benchmarks for training is costly. 2) the synthetic training kernels generated as mentioned above can create various degradation in the individual frames, and thus are able to model real-world videos with varying levels of kernel temporal consistency.

Testing Data. We created our testing set with 10 sequences from the REDS testing set (000 and 010-018), denoted as *REDS10*, aiming to mimic the actual degradation of real-world videos that are of varying video dynamicity. Concretely, following our experiments in Sec. 4, we first sampled videos from the Something-Something dataset [11][1]. The sequences from Something-Something dataset were randomly sampled such that their estimated kernels had differing temporal kernel consistency. These kernels were then used to degrade our test set to mimic the degradation characteristics of real-world videos. We then randomly sampled a sequence from these estimated real-world kernel sequences and used it to downsample each selected video in *REDS10*[2]. As a result, our testing set has the similar degradation characteristics as that of real-world videos, while allow us to perform quantitative evaluations. The kernel temporal consistency of this test set can be found in the supplementary material (s.m. Fig. 2). For real-world video evaluations, we used videos from the Something-Something dataset. All implementation details can be found in the supplementary material (s.m. Sec. 3).

6.2. Effectiveness of Temporal Kernel Consistency

Temporal Kernel Estimation. We first studied the effectiveness of taking multiple frames into account for kernel estimation. In other words, instead of estimating kernels individually for each frame, we leveraged our key insight that the downsampling kernels of frames within a video are temporally consistent to achieve a faster and more accurate kernel estimation for videos. To this end, we modified the *estimator* to take in multiple LR frames, $\{y_{t+i}\}_{i=-N}^{i=N}$, and generated their corresponding estimated kernels, $\{\hat{k}_{t+i}^{(j)}\}_{i=-N}^{i=N}$. We then utilized the existing channel attention block in DAN by adopting an early fusion approach, which merges information at the beginning of the block, to exploit the inter-channel relationships not only between basic and conditional inputs, but also among temporal inputs. Specifically, the features of the HR frames are concatenated with the LR features in every CRB in order to leverage the existing structure of DAN's *estimator* without adding additional channels or layers as shown in the supplementary material (s.m. Fig. 1).

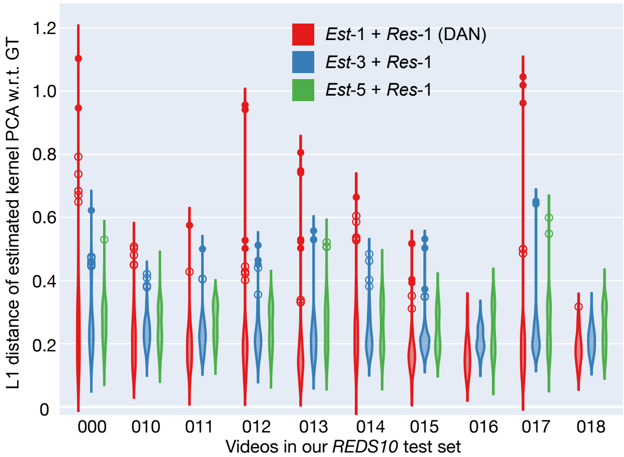

Figure 6: Distribution of kernel estimation errors of different *estimators* for each video sequence in our test set. Single-frame *estimator* (*Est*-1+*Res*-1) tends to perform worse than multi-frame estiamtors (*Est*-3/5+*Res*-1) by having larger error variance with many outliers.

We experimented with different number of input frames on the *estimator*, labelled as *Est*-α where α is the number of frames used for kernel estimation. Likewise, we labelled β as the number of frames used for restoration, *Res*-β. For a fair comparison, here we used DAN's *restorer*, $\beta = 1$, which is single-frame and therefore not including our adopted EDVR components. Fig. 6 shows the distribution of kernel estimation errors of the aforementioned models in terms of the absolute sum of PCA difference between the estimated kernels and their respective ground truth kernels for all frames in each sequence found in *REDS10*. We observed that independent kernel estimation per-frame can lead to a larger variance and numerous outliers as compared to temporal kernel estimation. Notably, temporal kernel estimation results in, on average, more accurate kernels for videos with high dynamicity, *i.e.* low kernel temporal consistency, while performs similarly for videos with high kernel temporal consistency. The performance increase in kernel estimation, however, did not improve performance significantly in video restoration as shown in Table 1. This

[1] In particular, sequences 13, 16, 21, 35, 37, 49, 52, 55, 63, and 71.

[2] For cases in which the length of the video is longer than the selected kernel sequence, we loop over the same kernel sequence for the remaining frames.

Models	PSNR/SSIM
Est-1 + *Res*-1 (DAN)	26.28/0.7118
Est-3 + *Res*-1	26.30/0.7124
Est-5 + *Res*-1	26.31/0.7213
Est-1 + *Res*-3	26.37/0.7170
Est-1 + *Res*-5	26.54/0.7287
***Est*-3 + *Res*-3**	**26.62/0.7364**
***Est*-5 + *Res*-5**	**26.76/0.7400**

Table 1: Ablation study on the impact of utilizing temporal kernel estimation on video restoration for both kernel estimation and motion compensation. Red for best performing model and blue for second best. Although estimating more accurate kernels did not significantly improve the performance of a single-image restorer, it is critical for motion compensation and hence benefiting multi-frame restorers.

phenomenon is also observed in recent blind iterative image SR works [12, 22] and these works reported that this was due to the *restorer*'s robustness to the kernel estimation errors of the *estimator* since they were jointly trained. Although having a more accurate kernel estimation did not drastically impact a single-frame video restoration performance, we show that it is essential at improving the performance of a multi-frame restoration approach.

Incorporating Temporal Kernels for MFSR. The performance gain of utilizing the temporal information of multiple frames is dependent on the accuracy of its motion estimation; an inaccurate flow can result in misaligned frames after motion compensation and thus artifacts in the restored video [34, 38, 35, 16]. As shown in Sec. 5, performing motion compensation under the assumption of a fixed SR kernel directly on real-world videos can result in regular artifacts in the warped frames. To mitigate this, instead of following the convention of employing motion compensation on the LR frames or features directly, we performed motion compensation on the LR frames *after* considering their corresponding kernels. Specifically, we utilized the feature maps at the last *restorer* iteration as shown in Fig. 5 which embed both LR frame and the corresponding kernel features from the *estimator*, and then adopted EDVR for temporal alignment, fusion, and restoration as mentioned in Sec. 6.1. This approach mitigates the problem of inaccurate motion compensation caused by kernel variation in real-world videos, but the restoration performance may still depend on the accuracy of estimated kernels; errors in kernel estimation would propagate and result in inaccurate motion compensation.

To verify this, we first ran our multi-frame *restorer*, $\beta = \{3, 5\}$, with a single-frame *estimator*, $\alpha = 1$ and compared it with running the multi-frame *restorer* together with the multi-frame *estimator*. The results are shown in Table 1. As expected, having a multi-frame *restorer* resulted in an improvement in video restoration similar to that of previous works [19, 17, 21, 24, 34]. However, these per-frame *estimator* MFSR models did not perform as well as their temporal estimator counterparts. In particular, although our per-frame *estimator* MFSR model utilized information from 5 frames (*Est*-1 + *Res*-5) to restore each frame, it did not outperform our temporal *estimator* MFSR model that only exploited information from 3 frames (*Est*-3 + *Res*-3). Hence, we can conclude that the kernel mismatch errors incurred during kernel estimation propagated through the implicit motion compensation module of EDVR, affecting temporal alignment, fusion, and thus restoration. In other words, more accurate estimated kernels through the temporal kernel *estimator* enable the multi-frame *restorer* to leverage temporal frame information better. Therefore, the interplay between accurate kernel estimation and motion compensation is the key to utilize temporal kernel consistency for video restoration.

Proposed for	Models	PSNR/SSIM
MFSR	TDAN [35]	25.93/0.6867
	EDVR [36]	26.21/0.7060
Blind SISR	IKC [12]	26.22/0.7021
	DAN [22]	26.28/0.7118
Blind MFSR	DBVSR [29]	26.11/0.6986
	***Est*-3 + *Res*-3 (Ours)**	**26.62/0.7364**
	***Est*-5 + *Res*-5 (Ours)**	**26.76/0.7400**

Table 2: We compare our model with state-of-the-art models from MFSR, which assume a fixed bicubic degradation, and blind single-image SR methods, which restore each frame independently.

Comparisons with Previous Works. We compared our approach, with existing works on both our test set *REDS10* and real-world videos taken from the Something-Something dataset. Specifically, we considered state-of-the-art MFSR methods, TDAN [35] and EDVR [36], blind image-based SR methods, IKC [12] and DAN [22], and a recently proposed blind MSFR approach, DBVSR [29].

From the quantitative and qualitative comparisons based on *REDS10* (Table. 2 & Fig. 7) and real-world qualitative examples (Fig. 8), we observe that existing MFSR approaches are lacking due to kernel mismatch, affecting both motion compensation and video restoration as shown in Sec. 5. Both TDAN and EDVR, in particular, were trained using the fixed bicubic degradation assumption and DBVSR assumed a fixed temporally uniform kernel. Blind SISR approaches, on the other hand, restore each frame independently and hence perform slightly better than existing MFSR approaches. Our approach, which exploits kernel temporal consistency for accurate kernel estimation and mitigates the effects of kernel mismatch on motion compensation, leads to a dominant solution for real-world video restoration. We provided additional examples in the supplementary material (s.m. Fig. 5 & Fig. 6).

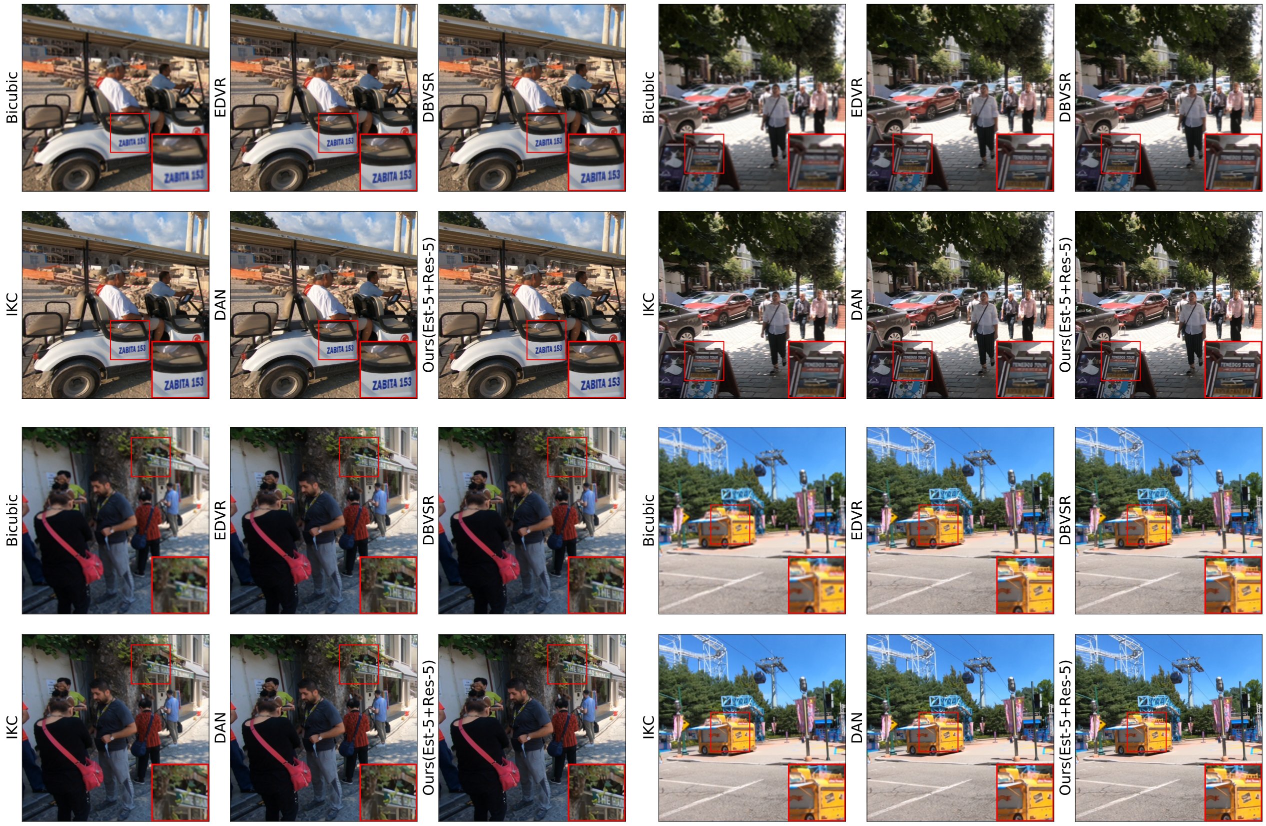

Figure 7: Qualitative comparison among existing models, along with bicubic upscaling, on our benchmark test sequences. Zoom in for best results.

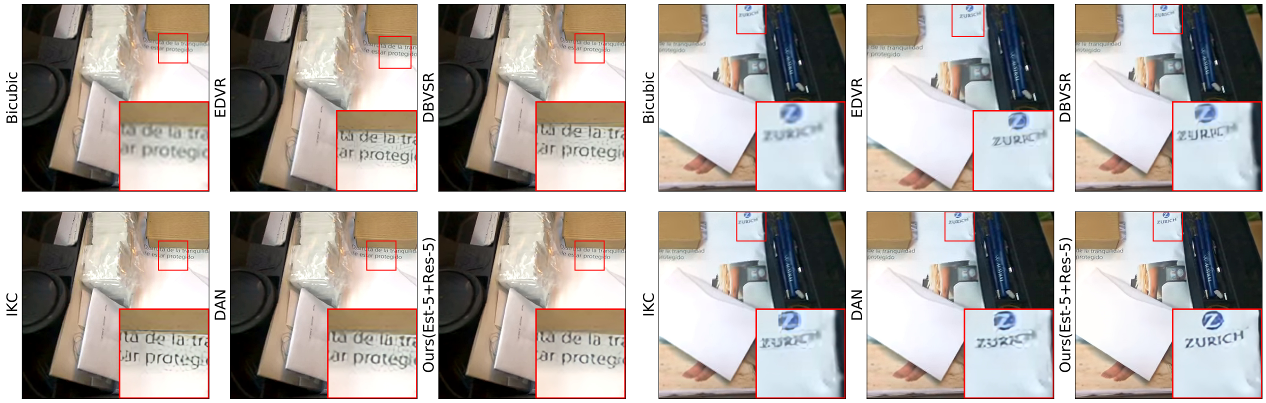

Figure 8: Real-world qualitative comparisons among existing models, along with bicubic upscaling. Zoom in for best results. Note that there is no ground-truth available.

7. Conclusion

In this paper, we presented the temporal kernel changes in videos and showed that they varied in their consistency depending on the video's dynamicity. Through our experiments, we highlighted the importance of estimating kernels per-frame to tackle the effects of temporal kernel mismatch in previous works. We then showed how temporal kernel consistency can be generally incorporated into existing works through the interaction between both kernel estimation and motion compensation in order to leverage both temporal kernel and frame information for blind video SR. We hope to influence future blind video SR model design by emphasizing the potential of leveraging kernel temporal consistency in restoring videos.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network. In *ECCV*, 2018. 2

[2] S. Baker, D. Scharstein, J. Lewis, S. Roth, Michael J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 2007. 2

[3] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*. 2019. 1, 2, 3, 5

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[5] Jifeng Dai, Haozhi Qi, Y. Xiong, Y. Li, Guodong Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1, 2, 3

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the Super-Resolution Convolutional Neural Network. In *ECCV*, 2016. 2

[8] N. Efrat, Daniel Glasner, Alexander Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1, 2, 4

[9] Sina Farsiu, M. D. Robinson, Michael Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 2004. 3

[10] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 2

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The Something Something Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 2017. 3, 6

[12] Jinjin Gu, Hannan Lu, W. Zuo, and C. Dong. Blind super-resolution with iterative kernel correction. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 6, 7

[13] Yong Guo, Jian Chen, J. Wang, Q. Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[14] Eddy Ilg, N. Mayer, Tonmoy Saikia, Margret Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[15] Max Jaderberg, K. Simonyan, Andrew Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015. 2

[16] Younghyun Jo, S. Oh, Jaeyeon Kang, and S. Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2, 7

[17] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and A. Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2016. 2, 4, 7

[18] Royson Lee, L. Dudziak, M. Abdelfattah, Stylianos I. Venieris, H. Kim, Hongkai Wen, and N. Lane. Journey towards tiny perceptual super-resolution. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[19] Renjie Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video super-resolution via deep draft-ensemble learning. *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 4, 7

[20] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 1, 2, 3

[21] Ding Liu, Zhaowen Wang, Yuchen Fan, X. Liu, Zhangyang Wang, S. Chang, and T. Huang. Robust video super-resolution with learned temporal dynamics. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 4, 7

[22] Zhengxiong Luo, Y. Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. In *Advances in Neural Information Processing Systems*. 2020. 1, 2, 3, 5, 6, 7

[23] Z. Ma, Renjie Liao, X. Tao, L. Xu, J. Jia, and Enhua Wu. Handling motion blur in multi-frame super-resolution. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[24] Osama Makansi, Eddy Ilg, and T. Brox. End-to-end learning of video super-resolution with motion compensation. In *German Conference on Pattern Recognition (GCPR)*, 2017. 2, 4, 7

[25] T. Michaeli and M. Irani. Nonparametric blind super-resolution. *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2

[26] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. 4, 5

[27] A. Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[28] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and M. Brown. Frame-recurrent video super-resolution. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[29] Jin shan Pan, Songsheng Cheng, Jiawei Zhang, and J. Tang. Deep blind video super-resolution. *ArXiv*, 2020. 1, 2, 3, 4, 7

[30] W. Shi, J. Caballero, Ferenc Huszár, J. Totz, A. Aitken, R. Bishop, D. Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[31] Assaf Shocher, N. Cohen, and M. Irani. ”zero-shot” super-resolution using deep internal learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[32] Dehua Song, Chang Xu, Xu Jia, Yiyi Chen, Chunjing Xu, and Yunhe Wang. Efficient residual dense block search for image super-resolution. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020. 2

[33] Deqing Sun, X. Yang, Ming-Yu Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 4

[34] X. Tao, H. Gao, Renjie Liao, J. Wang, and J. Jia. Detail-revealing deep video super-resolution. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 4, 7

[35] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 7

[36] Xintao Wang, Kelvin C. K. Chan, K. Yu, C. Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 1, 2, 5, 7

[37] X. Xiang, Yapeng Tian, Yulun Zhang, Y. Fu, J. Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[38] Tianfan Xue, B. Chen, Jiajun Wu, D. Wei, and W. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 2018. 2, 4, 7

[39] K. Zhang, L. Gool, and R. Timofte. Deep unfolding network for image super-resolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[40] Kai Zhang, W. Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[41] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[42] Ruofan Zhou and S. Süsstrunk. Kernel modeling super-resolution on real low-resolution images. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1

[43] X. Zhu, H. Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2