DeMFI: Deep Joint Deblurring and Multi-Frame Interpolation with Flow-Guided Attentive Correlation and Recursive Boosting

Jihyong Oh Munchurl Kim*
Korea Advanced Institute of Science and Technology
{jhoh94, mkimee}@kaist.ac.kr

Abstract

In this paper, we propose a novel joint deblurring and multi-frame interpolation (DeMFI) framework, called DeMFI-Net, which accurately converts blurry videos of lower-frame-rate to sharp videos at higher-frame-rate based on flow-guided attentive-correlation-based feature bolstering (FAC-FB) module and recursive boosting (RB), in terms of multi-frame interpolation (MFI). The DeMFI-Net jointly performs deblurring and MFI where its baseline version performs feature-flow-based warping with FAC-FB module to obtain a sharp-interpolated frame as well to de-blur two center-input frames. Moreover, its extended version further improves the joint task performance based on pixel-flow-based warping with GRU-based RB. Our FAC-FB module effectively gathers the distributed blurry pixel information over blurry input frames in feature-domain to improve the overall joint performances, which is computationally efficient since its attentive correlation is only focused pointwise. As a result, our DeMFI-Net achieves state-of-the-art (SOTA) performances for diverse datasets with significant margins compared to the recent SOTA methods, for both deblurring and MFI. All source codes including pretrained DeMFI-Net are publicly available at https://github.com/JihyongOh/DeMFI.

1. Introduction

Video frame interpolation (VFI) converts a low frame rate (LFR) video to a high frame rate (HFR) one between given consecutive input frames, thereby providing a visually better motion-smoothed video which is favorably perceived by human visual systems (HVS) [24, 25]. Therefore, it is widely used for diverse applications, such as adaptive streaming [52], slow motion generation [2, 18, 28, 30, 37, 44] and space-time super resolution [9, 15, 21, 22, 50, 51, 53–55].

On the other hand, motion blur is necessarily induced by either camera shake [1, 58] or object motion [33, 59] due to the accumulations of the light during the exposure period [14, 16, 49] when capturing videos. Therefore, eliminating the motion blur, called deblurring, is essential to synthesize sharp intermediate frames while increasing temporal resolution. The discrete degradation model for blurriness is generally formulated as follows [13, 19, 20, 29, 40, 41, 45]:

\[ B := \{ B_{2\tau} \}_{i=0,1,...} = \left\{ \frac{1}{2\tau + 1} \sum_{j=1}^{iK+\tau} S_j \right\}_{i=0,1,...} \quad (1) \]

where \( S_j \), \( B \), \( K \) and \( 2\tau + 1 \) denote latent sharp frame at time \( j \) in HFR, observed blurry frames at LFR, a factor that reduces frame rate of HFR to LFR and an exposure time period, respectively. However, a few studies have addressed the joint problem of video frame interpolation with blurred degradation namely as a joint deblurring and frame interpolation problem. To handle this problem effectively, five works [13, 19, 40, 41, 61] delicately have shown that joint approach is much better than the cascade of two separate tasks such as deblurring and VFI, which may lead to sub-optimal solutions. However, the methods [13, 19, 40, 41] simply perform a center-frame interpolation (CFI) between two blurry input frames. This implies that they can only pro-

*Corresponding author.
duce intermediate frames of time at a power of 2 in a recursive manner. As a result, the prediction errors are accumulatively propagated to the later interpolated frames. Also, their methods cannot produce interpolated frames at arbitrary target time instances, not at time of power of 2.

To overcome these limitations for improving the quality in terms of multi-frame interpolation (MFI) with a temporal up-scaling factor \( \times M \), we propose a novel framework for joint deblurring and multi-frame interpolation, called DeMFI-Net, to accurately generate sharp-interpolated frames at arbitrary time \( t \) based on flow-guided attentive-correlation-based feature bolstering (FAC-FB) module and recursive boosting (RB). However, using a pretrained optical flow estimator is not optimal for blurry input frames and is computationally heavy. So, our DeMFI-Net is designed to learn self-induced feature-flows (\( f_F \)) and pixel-flows (\( f_P \)) in warping the given blurry input frames for synthesizing a sharp-interpolated frame at arbitrary time \( t \), without any help of pretrained optical flow networks.

Direct estimation of flows to jointly deblur and interpolate the intermediate frame at arbitrary \( t \) from the blurry input frames is a very challenging task. To effectively handle it, our DeMFI-Net is designed by dividing and conquering the joint task into a two-stage problem as shown in Fig. 2:

- (i) the first stage (baseline version, denoted as DeMFI-Net\(_{bs} \)) jointly performs deblurring and MFI based on feature-flow-based warping and blending (FWB) by learning \( f_F \) to obtain a sharp-interpolated frame of \( t \in (0, 1) \) as well as to deblur two center-input frames (\( B_0, B_1 \)) of \( t = 0, 1 \) from four blurry input frames (\( B_{-1}, B_0, B_1, B_2 \)); and

- (ii) the second stage (recursive boosting, denoted as DeMFI-Net\(_{rb} \)) further boosts the joint performance based on pixel-flow-based warping and blending (PWB) by iteratively updating \( f_P \) with the help of GRU-based RB. It fully exploits the obtained output of DeMFI-Net\(_{bs} \) by adopting residual learning. It is trained with recursive boosting loss that enables the recursive iterations to be properly regulated during inference time by considering runtime or computational constraints, even after the training is finished.

It should be noted that (1) the FWB of DeMFI-Net\(_{bs} \) is a warping and blending operation in feature-domain where the resulting learned features tend to be more sharply constructed from the blurry inputs; and (2) the following PWB of DeMFI-Net\(_{rb} \) can be precisely performed in pixel-domain for the output of DeMFI-Net\(_{bs} \) via the residual learning to boost the performance of the joint task.

The blurry input frames implicitly contain abundant useful latent information due to an accumulation of light [14, 16, 49], as also shown in Eq. 1. Motivated from this, we propose a novel flow-guided attentive-correlation-based feature bolstering (FAC-FB) module that can effectively bolster the source feature \( F_0 \) (or \( F_1 \)) by extracting the useful information in the feature-domain from its counterpart feature \( F_1 \) (or \( F_0 \)) in guidance of self-induced flow \( f_{01} \) (or \( f_{10} \)). By doing so, the distribution of pixel information over four blurry input frames can be effectively gathered into the corresponding features of the two center-input frames which can then be utilized to restore sharp-interpolated frames and two deblurred center-input frames.

In the performance evaluation, DeMFI-Net\(_{bs} \) performs previous SOTA methods for three benchmark datasets including both diverse real-world scenes and larger-sized blurry videos. The final DeMFI-Net\(_{rb} \) further pushes its capability for MFI with large margins which has shown a strong generalization of our DeMFI-Net framework as shown in Fig. 1. Extensive experiments with diverse ablation studies have demonstrated the effectiveness of our framework. All source codes including pretrained DeMFI-Net are publicly available at https://github.com/JihyongOh/DeMFI.

2. Related Works

2.1. Center-Frame Interpolation (CFI)

The VFI methods on CFI only interpolate a center-frame between two consecutive sharp input frames. Since the interpolation is fixedly targeted at the center time position, they tend not to rely on optical flow networks. SepConv [32] generates dynamically separable filters to handle motions efficiently. CAIN [6] employs a channel attention module to extract motion information effectively without explicit estimation of motion. FeFlow [12] adopts deformable convolution [8] in the center frame generator to replace optical flows. AdaCoF [26] handles a complex motion by introducing a warping module in a generalized form.

However, all the above methods simply try to do CFI for two times (\( \times 2 \)) increase in frame rates, not for arbitrary time \( t \). This approach tends to limit the performance when being applied for MFI because they must be recursively applied after each center frame is interpolated, which causes error propagation into later-interpolated frames.

2.2. Multi-Frame Interpolation (MFI)

To effectively synthesize an intermediate frame at arbitrary time \( t \), many VFI methods on MFI for sharp input frames adopt a flow estimation-based warping operation.
SupersloMo [18] jointly combines occlusion maps and approximated intermediate flows to synthesize the intermediate frame. Quadratic video frame interpolation [27, 56] adopts the acceleration-aware approximation for the flows in quadratic form to better handle nonlinear motion. DAIN [2] proposes flow projection layer to delicately approximate the flows according to depth information. SoftSplat [31] directly performs forward warping for the feature maps of input frames with learning-based softmax weights for the occluded region. ABME [36] proposes an asymmetric bilateral motion estimation based on bilateral cost volume [35]. XVFI [44] introduces a recursive multi-scale shared structure to effectively capture large motion. However, all the above methods handle MFI problems for sharp input frames, which may not work well for blurry input frames.

2.3. Joint Deblurring and Frame Interpolation

The previous studies on the joint deblurring and frame interpolation tasks [13, 19, 40, 41, 61] have consistently shown that the joint approaches are much better than the simple cascades of two separately pretrained networks of deblurring and VFI. TNTT [19] first extracts several clear keyframes which are then subsequently used to generate intermediate sharp frames by adopting a jointly optimized cascaded scheme. It takes an approximate recurrent approach by unfolding and distributing the extraction of the frames over multiple processing stages. BIN [40] adopts a ConvLSTM-based [43] recurrent pyramid framework to effectively propagate the temporal information over time. Its extended version with a larger model size, called PRF [41], simultaneously yields the deblurred input frames and temporally center-frame at once. ALANET [13] employs the combination of both self- and cross-attention modules to adaptively fuse features in latent spaces, thus allowing for robustness and improvement in the joint task performances.

However, all the above four joint methods simply perform the CFI for blurry input frames so their performances are limited to MFI for the joint task. On the other hand, UTI-VFI [61] can interpolate the sharp frames at arbitrary time \( t \) in two-stage manner. It first extracts deblurred key-state frames at both start time and end time of the camera exposures, and then warps them to arbitrary time \( t \). However, its performance necessarily depends on the quality of flows obtained by a pretrained optical flow network which also increases the complexity of the overall network (+8.75M parameters).

Distinguished from all the above methods, our proposed framework elaborately learns self-induced \( f_P \) and \( f_F \) to effectively warp the given blurry input frames for synthesizing a sharp-interpolated frame at arbitrary time, without any pretrained optical flow network. As a result, our method not only outperforms the previous SOTA methods in structural-related metrics but also shows higher temporal consistency of visual quality performance for diverse datasets.

3. Proposed Method : DeMFI-Net

3.1. Design Considerations

Our network, DeMFI-Net, aims to jointly interpolate a sharp intermediate frame at arbitrary time \( t \) and deblur the blurry input frames. Most of the previous SOTA methods [13, 19, 40, 41] only consider CFI \((x \times 2)\) and need to perform it recursively at the power of 2 for MFI \((x \times M)\) between two consecutive input frames. Here, it should be noted that the later-interpolated frames must be sequentially created based on their previously-interpolated frames. Therefore, the errors are inherently propagated into later-interpolated frames so that they often have lower visual qualities.

Our DeMFI-Net is designed to interpolate intermediate frames at multiple time instances without dependency among them so that the error propagation problem can be avoided. That is, the multiple intermediate frames can be sequentially generated. To synthesize an intermediate frame at time \( t \in (0, 1) \) instantaneously, we adopt a warping operation which is widely used in VFI research [2, 18, 27, 44, 56] to interpolate the frames based on a backward warping [17] with estimated flows from time \( t \) to 0 and 1, respectively. However, direct usage of a pretrained optical flow network is not optimal for blurry frames and even computationally heavy. So our DeMFI-Net is devised to learn self-induced flows for robust warping in both feature- and pixel-domain. Furthermore, to effectively handle the joint task of deblurring and interpolation, we take a divide-and-conquer approach to the design of our DeMFI-Net in a two-stage manner: baseline version (DeMFI-Net\(_b\)) and recursive boosting version (DeMFI-Net\(_r\)) as shown in Fig. 2. DeMFI-Net\(_b\) first performs feature-flow-based warping and blending (FWB) to produce the deblurred input frames and a sharp-interpolated frame at the given \( t \). Then the output of DeMFI-Net\(_b\) is boosted for further improvement in DeMFI-Net\(_r\), by performing pixel-flow-based warping and blending (PWB). The DeMFI-Net\(_b\) and DeMFI-Net\(_r\) are described with more details in the following subsections.

3.2. DeMFI-Net\(_b\)

Fig. 3 (a) shows the architecture of DeMFI-Net\(_b\) that first takes four consecutive blurry input frames \((B_{-1}, B_0, B_1, B_2)\). Then, feature flow residual dense backbone (FF-RDB) module is followed which is similar to a backbone network of [40, 41], described in Appendices. Its modified 133 \((= 64 \times 2 + 2 \times 2 + 1)\) output channels are composed of 64 \( \times 2 \) for two feature maps \((F''_0, F''_1)\) followed by tanh functions, \(2 \times 2\) two bidirectional feature-domain flows \((f_{01}, f_{10})\) and 1 for an occlusion map logit \((o_{00})\).

*t-Alignment. To fully exploit the bidirectional flows \((f_{01}, f_{10})\) extracted from four blurry inputs, the intermediate
flows \( f_{0t} \) (or \( f_{1t} \)) from time 0 (or 1) to time \( t \) are linearly approximated as \( f_{0t} = t \cdot f_{01} \) (or \( f_{1t} = (1 - t) \cdot f_{10} \)). Then we apply the complementary flow reversal (CFR) [4-4] for \( f_{0t} \) and \( f_{1t} \) to finally approximate \( f_{01} \) and \( f_{11} \). Finally, we obtain \( t \)-aligned feature \( F_t \) by applying the backward warping operation \( W_b \) [17] for features \( F_0' \), \( F_1' \) followed by a blending operation with the occlusion map. This is called feature-flow-based warping and blending (FWB), which is depicted by the green box in Fig. 3 (a). The \( t \)-aligned feature \( F_t \) is computed as follows:

\[
F_t = FWB(F_{0t}, F_{1t}, f_{0t}, f_{1t}, t) = \frac{(1 - t) \cdot \sigma_{0t} \cdot W_b(F_{0t}', f_{0t}) + t \cdot \sigma_{1t} \cdot W_b(F_{1t}', f_{1t})}{(1 - t) \cdot \sigma_{0t} + t \cdot \sigma_{1t}},
\]

(2)

where \( \sigma_{0t} = \sigma(o_{0t}) \) and \( \sigma_{1t} = 1 - \sigma(o_{0t}) \), and \( \sigma \) is a sigmoid activation function.

**FAC-FB Module.** Since the pixel information is spread over the blurry input frames due to the accumulation of light [14, 16, 49] as in Eq. 1, we propose a novel FAC-FB module that can effectively bolster the source feature \( F_0' \) (or \( F_1' \)) by extracting the useful information in the feature-domain from its counterpart feature \( F_1' \) (or \( F_0' \)) in guidance of self-induced flow \( f_{01} \) (or \( f_{10} \)). The FAC-FB module in Fig. 3 (b) first encodes the two feature maps \( (F_0', F_1') \) by passing the outputs \( (F_0', F_1') \) of the FF-RDB module through its five residual blocks (ResB’s). The cascade (ResB’s) is shared for \( F_0' \) and \( F_1' \).

After obtaining the \( F_0 \) and \( F_1 \), the flow-guided attentive correlation (FAC) in Fig. 3 (b) computes attentive correlation of \( F_0 \) with respect to the positions of its counterpart feature \( F_1 \) pointed by the self-induced flow \( f_{01} \). The FAC on \( F_0 \) with respect to \( F_1 \) guided by \( f_{01} \) is calculated as:

\[
FAC_{01}(F_0, F_1, f_{01})(x) = [\sum_{c} Conv_1(F_0(x)) \odot Conv_1(F_1(x + f_{01}(x)))]
\]

(3)

where \( F_1(x + f_{01}(x)) \) is computed by bilinear sampling on a feature location \( x \). \( \odot \) and \( \sum_{c} \) denote element-wise multiplication, channel-wise summation and \( i \times i \)-sized convolution filter, respectively. The square bracket in Eq. 3 becomes a single channel scaling map which is then stretched along channel axis to be element-wise multiplied to \( Conv_1(F_1(x + f_{01}(x))) \). We block backpropagation to the flows in FAC for stable learning. Finally, the FAC-FB module produces bolstered features \( F_0^b \) for \( F_0 \) as:

\[
F_0^b = w_{01} \cdot F_0 + (1 - w_{01}) \cdot \frac{Conv_1(FAC_{01})}{\mathcal{E}_0}
\]

(4)
where \( w_{01} \) is a single channel of spatially-variant learnable weights that are dynamically generated by an embedded FAC\(_{01} \) via Conv1 (denoted as \( E_{0} \)) and \( F_{0} \) according to \( w_{01} = (\sigma \circ \text{Conv}_3 \circ \text{ReLU} \circ \text{Conv}_3)([E_{0}, F_{0}]). \) \([\cdot]\) means a concatenation along a channel axis. Similarly, FAC\(_{10} \) and \( F_{1}^{t} \) can be computed for \( F_{1} \) with respect to \( F_{0} \) by \( f_{10} \). The FAC-FB module allows the DeMFI-Net to effectively gather the distributed blurry pixel information over the blurry input frames in feature-domain to improve the joint performance. The FAC is computationally efficient because its attentive correlation is only computed in the focused locations pointed by the flows. Also, all filter weights in the FAC-FB module are shared for both \( F_{0} \) and \( F_{1} \).

**Refine Module.** After the FAC-FB Module in Fig. 3 (a), \( F_{0}^{bs}, F_{1}^{bs}, f_{01}, f_{10} \) and \( o_{0} \) are refined via the U-Net-based [39] Refine Module (RM) as \([F_{0}^{bs}, F_{1}^{bs}, f_{01}, f_{10}, o_{0}]) = \text{RM}(\text{Agg}^{1} + [F_{0}^{bs}, F_{1}^{bs}, f_{01}, f_{10}, o_{0}]), \) where \( \text{Agg}^{1} \) is the aggregation of \([F_{0}^{bs}, F_{1}^{bs}, f_{01}, f_{10}, o_{0}]) \) in the concatenated form. Then we get the refined feature \( F_{1}^{bs} \) at time \( t \) by \( F_{1}^{bs} = \text{FWB}([F_{0}^{bs}, F_{1}^{bs}, f_{01}, f_{10}, o_{0}]) \) as similar to Eq. 2. Here, we define a composite symbol at time \( t \) by the combination of two feature-flows and occlusion map logit as \( f_p \equiv ([f_{01}, f_{10}, o_{0}]) \) to be used in recursive boosting.

**Decoder I (D\(_1\)).** \( D_{1} \) is composed of ResBS\(_{5\times5} \) and it is intentionally designed to have a function: to decode a feature \( F_{i} \) at a time index \( j \) to a sharp frame \( S_{j} \). \( D_{1} \) is shared for all the three features \(([F_{0}^{bs}, F_{1}^{bs}, F_{1}^{bs}]) \). It should be noted that \( D_{1} \) decodes \( F_{0}^{bs}, F_{1}^{bs} \) and \( F_{1}^{bs} \) into sharp frames \( \{S_{0}, S_{1}, S_{1}^{1}\} \), respectively, which would be applied by L1 reconstruction loss (\( L_{D_{1}} \)) (Eq. 9). It is reminded that the architecture from the input layer to \( D_{1} \) constitutes our baseline version, called DeMFI-Net\(_{bs} \). Although DeMFI-Net\(_{bs} \) outperforms the previous SOTA methods, its extension with recursive boosting, called DeMFI-Net\(_{rb} \), can further improve the performance.

### 3.3. DeMFI-Net\(_{rb} \)

Since we have already obtained sharp frames \( \{S_{0}, S_{1}, S_{1}^{1}\} \) as the output of DeMFI-Net\(_{bs} \), they can further be sharpened based on the learned pixel-flows by recursive boosting via residual learning. It is known that feature-flows (\( f_{p} \)) and pixel-flows (\( f_{p} \)) would have similar characteristics [12, 26]. Therefore, the \( f_{p} \) obtained from the DeMFI-Net\(_{bs} \) are used as initial \( f_{p} \) for recursive boosting. For this, we design a GRU [5]-based recursive boosting for progressively updating \( f_{p} \) to perform PWB for two sharp frames at \( t = 0, 1 \) \([S_{0}, S_{1}] \) accordingly to boost the quality of a sharp intermediate frame at \( t \) via residual learning which has been widely adopted for effective deblurring [4, 10, 34, 38, 57]. Fig. 3 (c) shows \( i-th \) recursive boosting (RB) of DeMFI-Net\(_{rb} \), which is composed of Booster Module and Decoder II (\( D_{2} \)).

**Booster Module.** Booster Module iteratively updates \( f_{p} \) to perform PWB for \( \{S_{0}, S_{1}^{1}\} \) obtained from DeMFI-Net\(_{bs} \). The Booster Module is composed of Mixer and GRU-based Booster (GB), and it first takes a recurrent hidden state \((F_{i}^{rb} - 1) \) and \( f_{p} \) as an input to yield two outputs of \( F_{i}^{rb} \) and \( \Delta_{i} - 1 \) that is added on \( f_{p} - 1 \). Note that \( f_{p}^{rb} = f_{p} \) and \( \text{Agg}^{2} \) is not related to \( i-th \) recursive boosting. The updating process is given as follows:

\[
M_{i-1} = \text{Mixer}(\{\text{Agg}^{2}, f_{p}^{-i}\}) \\
[F_{i}^{rb}, \Delta_{i-1}] = \text{GB}(\{F_{i}^{rb}, M_{i-1}\}) \\
f_{p}^{rb} = f_{p}^{-i} + \Delta_{i-1},
\]

where the initial feature \( F_{0}^{rb} \) is obtained as a 64-channel feature via channel reduction for \( \text{Conv}1([F_{0}^{bs}, F_{1}^{bs}, F_{1}^{bs}]) \) of 192 channels. More details are provided for the Mixer and the updating process of GB in Appendices.

**Decoder II (D\(_2\)).** \( D_{2} \) in Fig. 3 (c) is composed of ResBS\(_{5\times5} \). It fully exploits abundant information of \( \text{Agg}^{2}_{rb} = \{S_{0}, S_{1}, S_{1}^{1}, B_{0}, B_{1}, B_{2}, f_{10}, f_{1}, f_{0}, f_{rb} \} \) to finally generate the refined outputs \( \{S_{0}, S_{1}, S_{1}^{1}\} \) via residual learning, where \( S_{1}^{1} = \text{PWB}(S_{0}, S_{1}, f_{p}^{rb}) \) is operated by only using the updated \( f_{p}^{rb} \) after the \( i-th \) recursive boosting to enforce the flows to be better boosted.

**Loss Functions.** The total loss function \( \mathcal{L}_{total} \) is given as:

\[
\mathcal{L}_{total} = \mathcal{L}_{D_{1}} + \sum_{i=1}^{N_{trn}} \mathcal{L}_{D_{2}}^{rec}
\]

\[
\mathcal{L}_{D_{1}} = (\sum_{j \in (0,t,1)} \|S_{j}^{i} - GT_{j}\|) / 3
\]

\[
\mathcal{L}_{D_{2}} = (\sum_{j \in (0,t,1)} \|S_{j}^{i} - GT_{j}\|) / 3,
\]

where \( GT_{j} \) and \( N_{trn} \) denote the ground-truth sharp frame at time \( j \) and total numbers of recursive boosting for training, respectively. We denote DeMFI-Net\(_{rb}(N_{trn}, N_{tst}) \) as DeMFI-Net\(_{rb} \) that is trained with \( N_{trn} \) and is tested by \( N_{tst} \) recursive boosting. The second term in the right-hand side of Eq. 8 is called as a recursive boosting loss. It should be noted that DeMFI-Net\(_{rb} \) is jointly trained with the architecture of DeMFI-Net\(_{bs} \) in an end-to-end manner using Eq. 8 without any complex learning schedule. Note that DeMFI-Net\(_{rb} \) is trained with only Eq. 9 from the scratch.

On the other hand, the design consideration for Booster Module was partially inspired from the work [48] which is here carefully modified for more complex process of DeMFI; (i) Due to the absence of ground-truth for the pixel-flows from \( t \) to 0 and 1, self-induced pixel-flows are instead learned by adopting \( D_{2} \) (Decoder II) and the recursive boosting loss; (ii) \( f_{p} \) is not necessarily to be learned precisely, instead to improve the final joint performance of sharpening the \( S_{0}, S_{1}, S_{1}^{1} \) via PWB and \( D_{2} \) as shown in Fig. 3 (c). So, we do not block any backpropagation to \( f_{p} \) per every recursive boosting unlike in [48], to fully focus on boosting the performance.
4. Experiment Results

4.1. Implementation Details

Training Dataset. To train our network, we use Adobe240 dataset [45] which contains 120 videos of 1,280×720 @ 240fps. We follow a blurry formation setting of [13,40,41] by averaging 11 consecutive frames at a stride of 8 frames over time to synthesize blurry frames captured by a long exposure, which finally generates blurry frames of 30fps with $K = 8$ and $\tau = 5$ in Eq. 1. The resulting blurry frames are downsized to 640×352 as done in [13,40,41].

Training Strategy. Each training sample is composed of four consecutive blurry input frames ($B_{t-1}, B_0, B_1, B_2$) and three sharp-target frames ($GT_0, GT_t, GT_1$) where $t$ is randomly determined in multiple of 1/8 with $0 < t < 1$. The filter weights of the DeMFI-Net are initialized by the Xavier method [11] and the mini-batch size is set to 2. DeMFI-Net is trained with a total of 420K iterations (7,500 epochs) by using the Adam optimizer [23] with the initial learning rate set to $10^{-4}$ and reduced by a factor of 2 at the 3,750- and 6,250-epoch marks in both spatial and temporal directions for data augmentation. Training takes about five days for DeMFI-Net, and two weeks for DeMFI-Netsv by using a single GPU with PyTorch in an NVIDIA DGX™ platform.

4.2. Comparison to Previous SOTA Methods

We mainly compare our DeMFI-Net with five previous joint SOTA methods; TNTT [19], UTI-VFI [61], BIN [40], PRF [41] (a larger-sized version of BIN) and ALANET [13], which all have adopted joint learning for deblurring and VFI. They all have reported better performance than the cascades of separately trained VFI [2,3,18] and deblurring [47,51] networks. It should be noted that the four methods of TNTT, BIN, PRF and ALANET simply perform CFI ($\times 2$), not at arbitrary $t$ but at the center time $t = 0.5$. So, they have to perform MFI ($\times 8$) recursively based on previously interpolated frames, which causes to propagate interpolation errors into later-interpolated frames. For experiments, we compare them in two aspects of CFI and MFI. For MFI performance, temporal consistency is measured such that the pixel-wise difference of motions are calculated in terms of tOF [7,44] (the lower, the better) for all 7 interpolated frames and deblurred two center frames for each blurry test sequence (scene). We also retrain the UTI-VFI with the same blurry formation setting for the Adobe240 for fair comparison, to be denoted as UTI-VFI*.

Test Dataset. We use three datasets for evaluation: (i) Adobe240 dataset [45], (ii) YouTube240 dataset and (iii) GoPro (HD) dataset (CC BY 4.0 license) [29] that contains large dynamic object motions and camera shakes. For the YouTube240, we directly selected 60 YouTube videos of 1,280×720 at 240fps by considering to include extreme scenes captured by diverse devices. Then they were resized to 640×352 as done in [13,40,41]. The Adobe240 contains 8 videos of 1,280×720 resolution at 240 fps and was also resized to 640×352, which is totally composed of 1,303 blurry input frames. On the other hand, the GoPro has 11 videos with total 1,500 blurry input frames but we used the original size of 1,280×720 for an extended evaluation in larger-sized resolution. All test datasets are also temporally downsampled to 30 fps with the blurring as [13,40,41].

Quantitative Comparison. Table 1 shows the quantitative performance comparisons for the previous SOTA methods including the cascades of deblurring and VFI methods with the Adobe240, in terms of deblurring and CFI ($\times 2$). Most results of the previous methods in Table 1 are from [13,40,41], except those of UTI-VFI (pre-trained, newly tested), UTI-VFI* (retrained, newly tested) and DeMFI-Nets (ours). Please note that all runtimes ($R_t$) in Table 1 were measured for 640×352-sized frames in the setting of [40,41] with one NVIDIA RTX™ GPU. As shown in Table 1, our proposed DeMFI-Netsv and DeMFI-Netsv clearly outperform all the previous methods with large margins in both deblurring and CFI performances, and the number of model parameters (#P) for our methods are the second- and third-smallest with smaller $R_t$ compared to PRF. In particular, DeMFI-Netsv(5,3) outperforms ALANET by 1dB and 0.0093 in terms of PSNR and SSIM, respectively for average performances of deblurring.

Table 1. Quantitative comparisons on Adobe240fps [45] for deblurring and center-frame interpolation ($\times 2$).

| Method          | $R_t$ | #P | Deblurring | CFI ($\times 2$) | Average |
|-----------------|-------|----|------------|-----------------|---------|
|                 |       |    | PSNR       | SSIM            | PSNR    | SSIM    |
| DeMFI-Netsv(5,3)| 2.02  | 198| 31.03      | 0.9172          | 31.03   | 0.9172  |
| DeMFI-Netsv(5,1)| 2.01  | 173| 31.34      | 0.9258          | 31.34   | 0.9258  |
| DeMFI-Netsv(1,1)| 2.05  | 157| 31.03      | 0.9287          | 31.03   | 0.9287  |
| DeMFI-Netsv(1,3)| 2.00  | 162| 31.34      | 0.9287          | 31.34   | 0.9287  |
| SRN [47]+[18]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[3]    | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [13]+[2]    | 5.20  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| DAIN [2]        | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[2]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[3]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[18]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[3]    | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [13]+[2]    | 5.20  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| DAIN [2]        | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[2]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[3]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[18]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[3]    | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [13]+[2]    | 5.20  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| DAIN [2]        | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[2]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[3]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[18]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [47]+[3]    | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| SRN [13]+[2]    | 5.20  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| DAIN [2]        | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[2]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
| EDVR [51]+[3]   | 5.12  | -  | 28.87      | 0.8673          | 28.87   | 0.8673  |
and CFI, and especially by average 1.51dB and 0.0124 for center-interpolated frames attributed to our warping-based framework with self-induced flows. Furthermore, even our DeMFI-Net\_bs is superior to all previous methods which are dedicatedly trained for CFI.

Table 2 shows quantitative comparisons of the joint methods for the three test datasets in terms of deblurring and MFI (×8). As shown in Table 2, all the three versions of DeMFI-Net significantly outperform the previous joint methods, which shows a good generalization of our DeMFI-Net framework. Fig. 1 shows PSNR profiles for MFI results (×8). As shown, the CFI methods such as TNTT and PRF tend to synthesize worse intermediate frames than the methods of interpolation at arbitrary time like UTI-VFI and our DeMFI-Net. This is because the error propagation is accumulated recursively due to the inaccurate interpolations by the CFI methods, which also has been inspected in VFI for sharp input frames [44]. Although UTI-VFI can interpolate the frames at arbitrary \( t \) by adopting the PWB combined with QVI [56], its performances inevitably depend on \( f_p \) quality obtained by PWC-Net [46], where adoption of a pre-trained net brings a disadvantage in terms of both \( R_t \) and CFI, and especially by average 1.51dB and 0.0124 for center-interpolated frames attributed to our warping-based framework with self-induced flows. Furthermore, even our DeMFI-Net\_bs is superior to all previous methods which are dedicatedly trained for CFI.

## 4.3. Ablation Studies
To analyze the effectiveness of each component in our framework, we perform ablation experiments. Table 3 shows the results of ablation experiments for FAC and RB in Fig. 3 (b)) with \( N_{itr} = 1 \) and \( N_{stt} = 1 \) for a simplicity. **FAC.** By comparing the method (f) to (d) and (c) to (a) in Table 3, it is noticed that FAC can effectively improve the overall joint performances in the both cases without and with RB by taking little more runtime (+0.06s) and small number of additional parameters (+0.09M).

**Figure 5. Visual comparisons for MFI results on GoPro (HD) for our and joint SOTA methods. Best viewed in zoom.**

#P (+8.75M). It is worthwhile to note that our method also shows the best performances in terms of temporal consistency with tOF by help of self-induced flows in interpolating sharp frames at arbitrary time \( t \).

**Qualitative Comparison.** Figs. 4 and 5 show the visual comparisons of deblurring and VFI performances on YouTube240 and GoPro datasets, respectively. As shown, the blurriness is easily visible between \( B_0 \) and \( B_1 \), which is challenging for VFI. Our DeMFI-Nets show better generalized performances for the extreme scenes (Fig. 4) and larger-sized videos (Fig. 5), also in terms of temporal consistency. Due to page limits, more visual comparisons with larger sizes are provided in Appendices for all three test datasets. Also the results of deblurring and MFI (×8) of all the SOTA methods are publicly available at [https://github.com/JihyongOh/DeMFI](https://github.com/JihyongOh/DeMFI). Please note that it is laborious but worth to get results for the SOTA methods in terms of MFI (×8).
Table 3. Ablation experiments on RB and FAC ($F^b_0 = F_0$) in terms of total average of deblurring and MFI ($\times 8$).

Table 4. Ablation study on $N_{\text{trn}}$ and $N_{\text{tst}}$ of DeMFI-Net$_{rb}$.

Figure 6. Effect of FAC. The green boxes show blurrier patches that are more attentive in the counterpart feature based on flow-guidance to effectively bolster the source feature.

Conv$_1$. The green boxes show blurrier patches that are more attentive in the counterpart feature based on $f_{10}$ to reinforce the source feature $F_1$ complementally. On the other hand, the less focused regions such as backgrounds with less blurs are relatively have smaller $E$ after FAC. In summary, FAC bolsters the source feature by complementing the important regions with blurs in the counterpart feature pointed by flow-guidance. We also show the effectiveness of FAC without flow guidance when trained with $f = 0$. As shown in Table 3, we obtained the performance higher than without FAC but lower than with FAC by flow-guidance, as expected. Therefore, we conclude that FAC works very effectively under the self-induced flow guidance to bolster the center features to improve the performance of the joint task.

Recursive Boosting. By comparing the method (d) to (a), (e) to (b) and (f) to (c) in Table 3, it can be known that the RB consistently yields improved final joint results. Fig. 7 shows that $f_P$ and $f_P$ have a similar tendency in flow characteristics. Furthermore, the $f_P$ updated from $f_P$ seems sharper to perform PWB in pixel domain, which may help our divide-and-conquer approach effectively handles the joint task based on warping operation. It is noted that our weakest variant (a) (w/o both RB and FAC) even outperformed the second-best joint method (UTI-VFI*) as shown in Table 2, 3 on the both Adobe240 and YouTube240.

# of Recursive Boosting $N_t$. To inspect the relationship between $N_{\text{trn}}$ and $N_{\text{tst}}$ for RB, we train the three variants of DeMFI-Net$_{rb}$ for $N_{\text{trn}} = 1, 3, 5$ as shown in Table 4. Since the weight parameters in RB are shared for each recursive boosting, all the variants have same $#P=7.41M$ and each column in Table 4 has same runtime $R_t$. The performances are generally boosted by increasing $N_{\text{trn}}$, where each recursion is attributed to the recursive boosting loss that enforces the recursively updated flow $f_P$ to better focus on synthesis $S^{t-1}_F$ via the PWB. It should be noted that the overall performances are better when $N_{\text{tst}} \leq N_{\text{trn}}$, while they are dropped otherwise. So, we can properly regulate $N_{\text{tst}}$ by considering $R_t$ or computational constraints, even though the training with $N_{\text{trn}}$ is once over. That is, under the same runtime constraint of each $R_t$ as in the column when testing, we can also select the model trained with larger $N_{\text{trn}}$ to generate better results. On the other hand, we found out that further increasing $N_{\text{trn}}$ does not bring additional benefits due to saturated performance of DeMFI-Net$_{rb}$.

5. Conclusion

We propose a novel joint deblurring and multi-frame interpolation framework, called DeMFI-Net, based on our novel flow-guided attentive-correlation-based feature bolstering (FAC-FB) module and recursive boosting (RB), by learning the self-induced feature- and pixel-domain flows without any help of pretrained optical flow networks. FAC-FB module forcefully enriches the source feature by extracting attentive correlation from the counterpart feature at the position where self-induced flow points at, to finally improve results for the joint task. Our DeMFI-Net achieves state-of-the-art performances for diverse datasets with significant margins compared to the previous SOTA methods for both deblurring and multi-frame interpolation (MFI).

Limitations. Extreme conditions such as tiny objects, low-light condition and large motion would make the joint task very challenging. We also provide visual results of the failure cases in Appendices in detail.
Acknowledgement. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00419, Intelligent High Realistic Video Evaluation (IITP) grant funded by the Korea government of Information & communications Technology Planning & Evaluation).
[31] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In CVPR, pages 5437–5446, 2020.
[32] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In ICCV, pages 261–270, 2017.
[33] Jinsun Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In CVPR, pages 1628–1636, 2016.
[34] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In ECCV, pages 327–343. Springer, 2020.
[35] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In ICCV, 2020.
[36] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Dual-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
[38] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Xingjian Shi. Bilateral motion estimation with bilateral cost volume. In CVPR, pages 6114–6123, 2020.
[39] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkou Min, and Zhiyong Gao. Blurry video frame interpolation. In CVPR, pages 5114–5123, 2020.
[40] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkou Min, and Zhiyong Gao. Video frame interpolation and enhancement via pyramid recurrent framework. IEEE Transactions on Image Processing, 30:277–292, 2020.
[41] 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.
[42] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In NeurIPS, 2015.
[43] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In ICCV, 2021.
[44] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In CVPR, pages 1279–1288, 2017.
[45] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In CVPR, pages 8934–8943, 2018.
[46] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Ji-ayia Jia. Scale-recurrent network for deep image deblurring. In CVPR, pages 8174–8182, 2018.
[47] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In ECCV, pages 402–419. Springer, 2020.
[48] Jacob Telleen, Anne Sullivan, Jerry Yee, Oliver Wang, Prabath Gunawardane, Ian Collins, and James Davis. Synthetic shutter speed imaging. In Computer Graphics Forum, volume 26, pages 591–598. Wiley Online Library, 2007.
[49] Yafeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In CVPR, pages 3360–3369, 2020.
[50] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In CVPRW, pages 0–0, 2019.
[51] Jiyan Wu, Chau Yuen, Ngai-Man Cheung, Junliang Chen, and Chang Wen Chen. Modeling and optimization of high frame rate video transmission over wireless networks. IEEE Transactions on Wireless Communications, 15(4):2713–2726, 2015.
[52] Xiaoyu Yang, Yapon Tian, Yulun Zhang, Yun Fu, Jan P Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In CVPR, pages 3370–3379, 2020.
[53] Zeyu Xiao, Zhiwei Xiong, Xueyang Fu, Dong Liu, and Zheng-Jun Zha. Space-time video super-resolution using temporal profiles. In ACM MM, pages 664–672, 2020.
[54] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Ming-Ming Cheng. Temporal modulation network for controllable space-time video super-resolution. In CVPR, pages 6388–6397, 2021.
[55] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In NeurIPS, pages 1647–1656, 2019.
[56] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In CVPR, pages 5978–5986, 2019.
[57] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. IEEE Transactions on Image Processing, 28(1):291–301, 2018.
[58] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In CVPR, pages 2737–2746, 2020.
[59] Yulun Zhang, Yafeng Tian, Yu Kong, Bineng Zhong, and Chang Wen Chen. Modeling and optimization of high frame rate video transmission over wireless networks. IEEE Transactions on Wireless Communications, 15(4):2713–2726, 2015.
[60] Youjian Zhang, Chaoyue Wang, and Dacheng Tao. Video frame interpolation without temporal priors. In NeurIPS, 2021.
Appendices

A. Details of Architecture for DeMFI-Net

A.1. DeMFI-Net_{bs}

A.1.1 Feature Flow Residual Dense Backbone (FF-RDB) Module

The feature flow residual dense backbone (FF-RDB) module first takes four consecutive blurry input frames \((B_{-1}, B_0, B_1, B_2)\). It is similar to a backbone network of [40, 41] and the number of output channels is modified to 133 (= 64 \times 2 + 2 \times 2 + 1). As shown in Fig. 8 (a), it consists of one DownShuffle layer and one UpShuffle layer [42], six convolutional layers, and twelve residual dense blocks [60] that are each composed of four Conv3’s, one Conv1, and four ReLU functions as in Fig. 8 (b). All the hierarchical features obtained by the residual dense blocks are concatenated for successive network modules. The 133 output channels are composed of 64 \times 2 for two feature maps \((F_0', F_1')\) followed by tanh activation functions, 2 \times 2 bidirectional feature-domain flows \((f_{01}, f_{10})\) and 1 for an occlusion map logit \((o_{01})\).

A.1.2 U-Net-based Refine Module (RM)

The U-Net-based [39] Refine Module (RM) takes Agg^\text{2} as an input to refine \(F_0, F_1, f_{01}, f_{10}\) and \(o_{01}\) in a residual learning manner as \([F_0', F_1', f_{01}', f_{10}', o_{01}'] = \text{RM}(\text{Agg}^\text{1}) + [F_0, F_1, f_{01}, f_{10}, o_{01}]\) where Agg^\text{1} is the aggregation of \([F_0, F_1, f_{01}, f_{10}, o_{01}, f_{01}, f_{10}]\) in the concatenated form.

A.2. DeMFI-Net_{rb}

A.2.1 Booster Module

Booster Module iteratively updates \(f_P\) to perform PWB for \(S_{01}, S_{10}\) obtained from DeMFI-Net_{bs}. The Booster Module is composed of Mixer and GRU-based Booster (GB), and it first takes a recurrent hidden state \((F_{i-1}^{rec})\) and \(f_P^{i-1}\) as an input to yield two outputs of \(F_i^{rec}\) and \(\Delta_{i-1}\) that is added on \(f_P^{i-1}\). Note that \(f_P^0 = f_P^0\) and Agg^\text{2} is not related to \(i\)-th recursive boosting. The updating process is given as follows:

\[
M_{i-1} = \text{Mixer}(\text{Agg}^2, f_P^{i-1}) \tag{11}
\]
\[
[F_{i-1}^{rec}, \Delta_{i-1}] = \text{GB}(F_{i-1}^{rec}, M_{i-1}) \tag{12}
\]
\[
f_P^i = f_P^{i-1} + \Delta_{i-1}, \tag{13}
\]

where the initial feature \(F_{0}^{rec}\) is obtained as a 64-channel feature via channel reduction for Conv1([\(F_0, F_1, f_{01}, f_{10}\)]) of 192 channels. More details about both Mixer and the updating process of GB are described in the following subsections.

A.2.2 Mixer

The first component in Booster Module is called Mixer. As shown in Fig. 10, Mixer first passes Agg^\text{2} and \(f_P^{i-1}\)
through each independent set of convolution layers as Conv\(_7\)−ReLU−Conv\(_3\)−ReLU, respectively, then yields \(M_{i-1}\) via Conv\(_3\)−ReLU−Conv\(_3\)−ReLU by taking concatenated outputs of the sets. \(M_{i-1}\) is consecutively used in GRU-based Booster (GB) as described in the following subsection.

### A.2.3 GRU-based Booster (GB)

GRU-based Booster (GB) takes both \(M_{i-1}\) and \(F_{i-1}^{\text{rec}}\) as an input to finally produce an updated \(F_i^{\text{rec}}\) which is consecutively used to make \(\Delta_{i-1}\) that is added on \(F_{i-1}^{\text{fp}}\). GB adopts gated activation unit based on the GRU cell [5] by replacing fully connected layers with two separable convolutions of \(1 \times 5\) (Conv\(_{1x5}\)) and \(5 \times 1\) (Conv\(_{5x1}\)) as in [48] to efficiently increase a receptive field. The detailed process in GB is operated as follows:

\[
\begin{align*}
z_{i}^{1 \times 5} &= \sigma(\text{Conv}_{1 \times 5}([F_{i-1}^{\text{rec}}, M_{i-1}]))) \\
r_{i}^{1 \times 5} &= \sigma(\text{Conv}_{1 \times 5}([F_{i-1}^{\text{rec}}, M_{i-1}]))) \\
F_{i}^{\text{rec}, 1 \times 5} &= \text{tanh}(\text{Conv}_{1 \times 5}(z_{i}^{1 \times 5} \odot F_{i-1}^{\text{rec}} + r_{i}^{1 \times 5} \odot F_{i}^{\text{rec}, 1 \times 5}))) \\
z_{i}^{5 \times 1} &= \sigma(\text{Conv}_{5 \times 1}([F_{i}^{\text{rec}, 1 \times 5}, M_{i-1}]))) \\
r_{i}^{5 \times 1} &= \sigma(\text{Conv}_{5 \times 1}([F_{i}^{\text{rec}, 1 \times 5}, M_{i-1}]))) \\
F_{i}^{\text{rec}} &= \text{tanh}(\text{Conv}_{5 \times 1}(z_{i}^{5 \times 1} \odot F_{i}^{\text{rec}, 1 \times 5} + r_{i}^{5 \times 1} \odot F_{i}^{\text{rec}, 5 \times 1}))) \\
\Delta_{i-1} &= (\text{Conv}_3 \circ \text{ReLU} \circ \text{Conv}_3)(F_{i}^{\text{rec}}).
\end{align*}
\]

Please note that Eq. 21, 22 produce the final outputs \((F_{i}^{\text{rec}}, \Delta_{i-1})\) of the Booster Module as shown in Fig. 3 (c) in the main paper, indicated by blue arrows.

### B. Additional Qualitative Comparison Results

Figs. 11, 12, 13, 14, 15 show the abundant visual comparisons of deblurring and MFI \((\times 8)\) performances for all the three test datasets. To better show them, we generally show the cropped patches for each scene. Since the number of blurry input frames for each method is different, two blurry center-input frames \((B_0, B_1)\) are averagely shown in the figures. As shown, the severe blurriness can easily be shown between two center-input frames \((B_0, B_1)\), which is very challenging for VFI.

Our DeMFI-Nets, especially DeMFI-Net\(_{5b}\), better synthesize textures or patterns (1st/2nd scenes of Fig. 11, Fig. 12, 1st scene of Fig. 15), precisely generate thin poles (3rd scene of Fig. 11) or fast moving objects (2nd/3rd scenes of Fig. 14) and effectively capture letters (Fig. 12, Fig. 13, 1st scene of Fig. 14, 2nd/3rd/4th scenes of Fig. 15), which tend to be failed by all the previous methods.

Especially, CFI methods such as TNTT and PRF are more hard to interpolate sharp frames at the time index 2/8 or 6/8 than 4/8 (center time instance) within each scene because they can only produce intermediate frames of time at a power of 2 in a recursive manner. As a result, the prediction errors are accumulatively propagated to the later interpolated frames. On the other hand, our DeMFI-Net framework adopts self-induced flow-based warping methodology trained in an end-to-end manner, which finally leads to generate temporally consistent sharp intermediate frames from blurry input frames. Also the results of deblurring and MFI \((\times 8)\) of all the SOTA methods are publicly available at https://github.com/JihyongOh/DeMFI for easier comparison. Please note that it is laborious but worth to get results for the SOTA methods in terms of MFI \((\times 8)\).

### C. Limitations: Failure Cases

Fig. 16 shows the failure cases such as tiny objects (1st scene), low-light condition (2nd scene) and large motion (3rd scene), which would make the joint task very challenging. First, in the case of splashed tiny objects with blurriness, it is very hard to capture sophisticated motions from the afterimages of the objects so all the methods fail to delicately synthesize the frames as GT’s. Second, in the case of low-light condition, it is hard to distinguish the boundaries of the objects (green arrows) and to detect tiny objects such as fast falling coffee beans (dotted green line), which deteriorates the overall performances of all the methods. Lastly, large and complex motion with blurriness due to camera shaking also makes all the methods hard to precisely synthesize final frames as well. We hope these kinds of failure cases will motivate researchers for further challenging studies.

### D. Visual Comparison with Video

We provide a visual comparison video for TNTT [19], UTI-VFI* (retrained ver.) [61], PRF [41] (a larger-sized version of [40]) and DeMFI-Net\(_{5b}\) (5.3) (ours), which all have adopted joint learning for deblurring and VFI. The video
named https://www.youtube.com/will-be-updated shows several multi-frame interpolated (×8) results played as 30fps for a slow motion, synthesized from blurry input frames of 30fps. All the results of the methods are adequately resized to be simultaneously played at a single screen. Please take into account that YouTube240 test dataset contains extreme motion with blurriness.

TNTT generally synthesize blurry visual results and PRF tends to show temporal inconsistency for MFI (×8). These two joint methods simply do CFI, not for arbitrary time t. Therefore, their methods must be recursively applied after each center frame is interpolated for MFI, which causes error propagation into later-interpolated frames. Although UTI-VFI* shows better visual results than above two CFI joint methods, it tends to produce some artifacts especially on large motion with blurriness and tiny objects such as splash of water. This tendency is attributed to the error accumulation from the dependency on fP quality inevitably obtained by pretrained PWC-Net [46], where adoption of a pretrained net also brings a disadvantage in terms of both Rt and #P (+8.75M). On the other hand, our DeMFI-Net framework is based on the self-induced feature- and pixel-domain flows without any help of pretrained optical flow networks, to finally better interpolate the sharp frames.
Figure 11. Visual comparisons for MFI results on Adobe240. *Best viewed in zoom.*
Figure 12. Visual comparisons for MFI results on Adobe240. *Best viewed in zoom.*

Figure 13. Visual comparisons for MFI results on GoPro (HD). *Best viewed in zoom.*
Figure 14. Visual comparisons for MFI results on YouTube240. Best viewed in zoom.
Figure 15. Visual comparisons for MFI results on GoPro (HD). Best viewed in zoom.
Figure 16. Failure cases; tiny objects, low-light condition and large motion. *Best viewed in zoom.*