IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation

Lingtong Kong[1]*, Boyuan Jiang[2]*, Donghao Luo[2], Wenqing Chu[2], Xiaoming Huang[2],
Ying Tai[2], Chengjie Wang[2], Jie Yang[1]†
[1]Shanghai Jiao Tong University, China, [2]Youtu Lab, Tencent
{ltkong, jieyang}@sjtu.edu.cn
{byronjiang, michaelluo, wenqingchu, skyhuang, yingtai, jasoncjwang}@tencent.com

Abstract

Prevailing video frame interpolation algorithms, that generate the intermediate frames from consecutive inputs, typically rely on complex model architectures with heavy parameters or large delay, hindering them from diverse real-time applications. In this work, we devise an efficient encoder-decoder based network, termed IFRNet, for fast intermediate frame synthesizing. It first extracts pyramid features from given inputs, and then refines the bilateral intermediate flow fields together with a powerful intermediate feature until generating the desired output. The gradually refined intermediate feature can not only facilitate intermediate flow estimation, but also compensate for contextual details, making IFRNet do not need additional synthesis or refinement module. To fully release its potential, we further propose a novel task-oriented optical flow distillation loss to focus on learning the useful teacher knowledge towards frame synthesizing. Meanwhile, a new geometry consistency regularization term is imposed on the gradually refined intermediate features to keep better structure layout. Experiments on various benchmarks demonstrate the excellent performance and fast inference speed of proposed approaches. Code is available at https://github.com/ltkong218/IFRNet.

1. Introduction

Video frame interpolation (VFI), that converts low frame rate (LFR) image sequences to high frame rate (HFR) videos is an important low-level computer vision task. Related techniques are widely applied to various practical applications, such as slow-motion generation [22], novel view synthesis [55] and cartoon creation [42]. Although it has been studied by a large number of researches, there are still great challenges when dealing with complicated dynamic scenes, which include large displacement, severe occlusion, motion blur and abrupt brightness change.

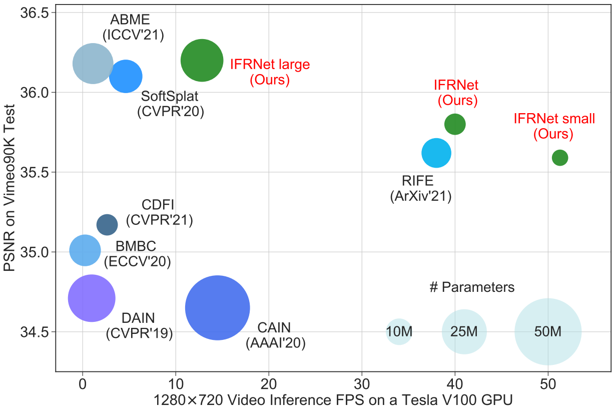


Figure 1. **Speed, accuracy and parameters comparison.** Proposed IFRNet achieves state-of-the-art frame interpolation accuracy with fast inference speed and lightweight model size.

Recently, with the development of optical flow networks [13, 24, 45, 46], significant progress has been made by flow-based VFI approaches [22, 33, 37, 49], since optical flow can provide an explicit correspondence to register frames in a video sequence. Successful flow-based approaches usually follow a three-step pipeline: **1)** Estimate optical flow between target frame and input frames. **2)** Warp input frames or context features by predicted flow fields for spatial alignment. **3)** Refine warped frames or features and generate the target frame by a synthesis network. Denoting input frames and target frame to be I_0, I_1 and I_t $(0 < t < 1)$, existing methods either first estimate optical flow $F_{0\to1}, F_{1\to0}$ [3,22,32,33,36], and then approximate or refine bilateral intermediate flow $F_{t\to0}, F_{t\to1}$ [9, 22, 40, 49] as shown in Figure 2 (a), or throw the intractable intermediate flow estimation sub-task to a learnable flow network for end-to-end training [20, 50, 54] as depicted in Figure 2 (b). Their common step is to further employ an image synthesis network to encode spatial aligned context feature [32] for target frame generation or refinement.

∗ Equal contribution. This work was done when Lingtong Kong was an intern at Tencent Youtu Lab.

† Corresponding author: Jie Yang (jieyang@sjtu.edu.cn). This research is partly supported by NSFC, China (No: 61876107, U1803261).

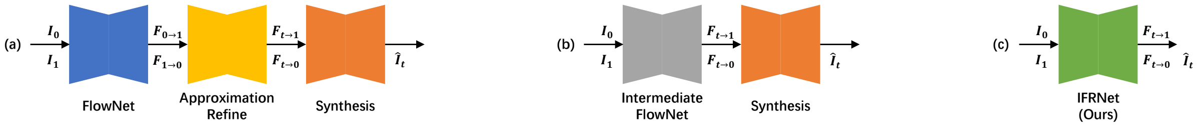


Figure 2. **Different flow-based VFI paradigms.** We roughly classify existing flow-based VFI methods based on encoder-decoders with specific function. In (a) [3,22,32,33,36,37,40,49], FlowNet estimates conventional optical flow $F_{0\to1}, F_{1\to0}$, the middle part approximates or further refines flow fields $F_{t\to0}, F_{t\to1}$. In (b) [20,50,54], the Intermediate FlowNet directly predicts intermediate flow of $F_{t\to0}, F_{t\to1}$. Both (a) and (b) contain a separate synthesis network for target frame generation. In (c), proposed IFRNet jointly refines the intermediate flow $F_{t\to0}, F_{t\to1}$ together with a powerful intermediate feature $\hat{\phi}_t$ to generate the target frame in a single encoder-decoder.

Although above pipeline that first estimates intermediate flow and then context feature has become the most popular paradigm for flow-based VFI approaches [9,32,33,37,40], it suffers from several defects. First, they divide intermediate flow and context feature refinement into separate encoder-decoders, which ignores the mutual promotion of these two crucial elements for frame interpolation. Second, their cascaded architecture based on above design concept can substantially increase the inference delay and model parameters, blocking them from mobile and real-time applications.

In this paper, we propose a novel Intermediate Feature Refine Network (IFRNet) for VFI to overcome the above limitations. For the first time, we merge above separated flow estimation and feature refinement into *a single encoder-decoder based model for compactness and fast inference*, abstracted in Figure 2 (c). It first extracts pyramid features from given inputs by the encoder, and then jointly refines the bilateral intermediate flow fields together with a powerful intermediate feature through coarse-to-fine decoders. The improved architecture can benefit intermediate flow and intermediate feature with each other, endowing our model with the ability to not only generate sharper moving objects but also capture better texture details.

For better supervision, we propose task-oriented flow distillation loss and feature space geometry consistency loss to effectively guide the multi-scale motion estimation and intermediate feature refinement. Specifically, our flow distillation approach adjusts the robustness of distillation loss adaptively in space and focuses on learning the useful teacher knowledge for frame synthesizing. Besides, proposed geometry consistency loss can employ the extracted intermediate features from ground truth to constrain the reconstructed intermediate features for keeping better structure layout. Figure 1 gives a speed, accuracy and parameters comparison among advanced VFI methods, demonstrating the state-of-the-art performance of our approaches. In summary, our main contributions are listed as follows:

- We devise a novel IFRNet to jointly perform intermediate flow estimation and intermediate feature refinement for efficient video frame interpolation.
- Task-oriented flow distillation loss and feature space geometry consistency loss are newly proposed to promote intermediate motion estimation and intermediate feature reconstruction of IFRNet, respectively.
- Benchmark results demonstrate that our IFRNet not only achieves state-of-the-art VFI accuracy, but also enjoys fast inference speed and lightweight model size.

2. Related Work

Video Frame Interpolation. The mainstream VFI methods can be classified into flow-based [3, 22, 29, 32, 33, 36, 37, 40, 49–51, 54], kernel-based [7, 8, 12, 25, 34, 35, 38] and hallucination-based approaches [10, 16, 23]. Different VFI paradigms have their own merits and flaws due to the substantial frame synthesizing manner. For example, kernel-based methods are good at handling motion blur by convolving over local patches [34, 35], successive works mainly extend it to deal with high resolution videos [38], increase the degrees of freedom for convolution kernel [7,8,25], or combine them with other paradigms for compensation [4, 12]. However, they are typically computationally expensive and short of dealing with occlusion. In another way, hallucination-based methods directly synthesize frames from the feature domain by blending field-of-view features generated by deformable convolution [11] or PixelShuffle operations [10]. They can naturally generate complex contextual details, while the predicted frames tend to be blurry when fast-moving objects exist.

Recently, significant progress has been made by flow-based VFI approaches, since optical flow can provide an explicit correspondence for frame registration. These solutions either employ an off-the-shelf flow model [32, 49] or estimate task-specific flow [22, 29, 37, 40, 50] as a guidance for pixel-level motion. Common subsequent step is to forward [14] or backward [48] warp input images to target frame, and finally refine warped frames by an image synthesis network [12, 32, 33, 37], often instantiated as a GridNet [15]. For achieving better image interpolation quality, more complicated deep models are devised to estimate intermediate flow fields [9, 49] and refine the generated target frame [22, 33, 36, 37]. However, the heavy computation cost and large inference delay make them unsuitable for resource limited devices. To take a breath from above module cascading competition, and reconsider the improvement of prior efficient flow-based VFI paradigm, *e.g.* DVF [29], we propose a novel single encoder-decoder based IFRNet, that can perform real-time inference with excellent accuracy.

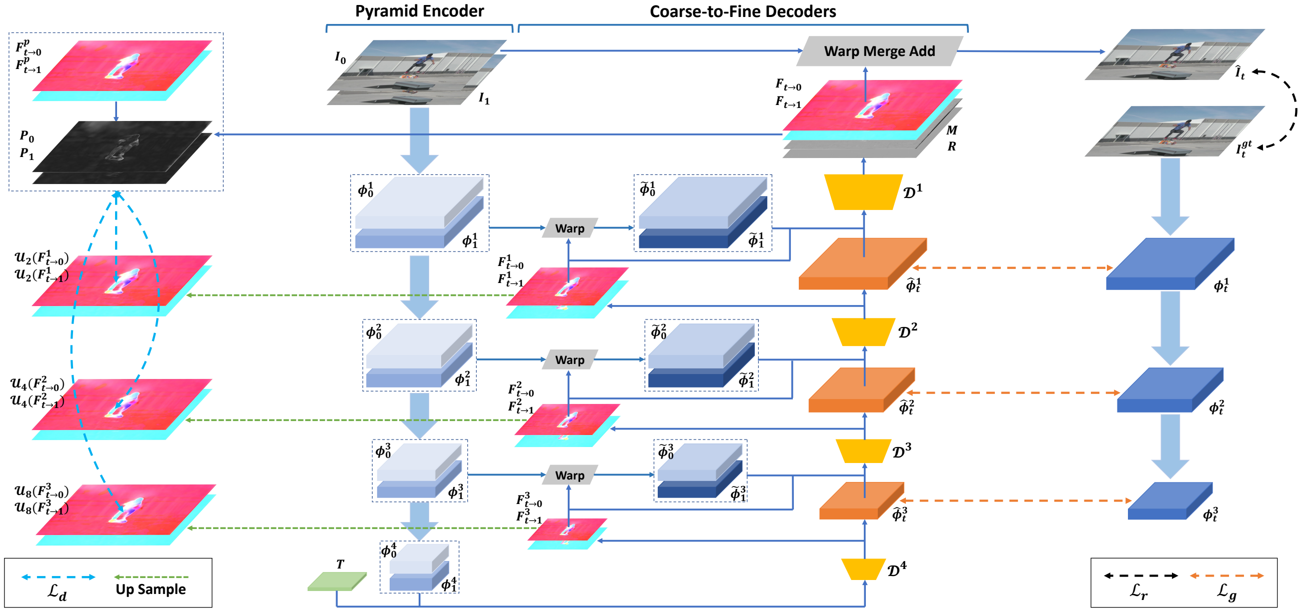


Figure 3. **Architecture overview and loss functions of IFRNet.** Our model is an efficient encoder-decoder based network, which first extracts pyramid context features from input frames with a shared encoder, and then gradually refines bilateral intermediate flow fields $F_{t\to0}, F_{t\to1}$ together with reconstructed intermediate feature $\hat{\phi}_t$ through coarse-to-fine decoders, until yielding the final output. Besides the common image reconstruction loss $\mathcal{L}_r$, task-oriented flow distillation loss $\mathcal{L}_d$ and feature space geometry consistency loss $\mathcal{L}_g$ are newly devised to guide the feature alignment procedure more effectively towards intermediate frame synthesizing.

Optical Flow Estimation. Finding dense correspondence between adjacent frames, namely optical flow estimation [19], has been studied for decades for its fundamental role in many downstream video processing tasks [5, 52]. FlowNet [13] is the first attempt to apply deep learning for optical flow estimation based on the encoder-decoder U-shape network. Inspired by traditional coarse-to-fine paradigm, SPyNet [39], PWC-Net [45] and FastFlowNet [24] integrate pyramid feature, backward warping and achieve impressive real-time performance. Knowledge distillation [18] also plays an important role in optical flow prediction, usually embodied as generating pseudo label in unsupervised optical flow learning [26, 27] or related tasks [1, 41]. A recent VFI method [20] also uses a distillation strategy to promote motion prediction. Beyond the difference of architecture design, our distillation approach can focus on the useful knowledge for intermediate frame synthesizing in a task adaptative manner.

3. Proposed Approach

In this section, we first introduce the IFRNet architecture built on the principle of joint refinement of intermediate flow and intermediate feature, to obtain an efficient encoder-decoder based framework for VFI. Then two novel objective functions, *i.e.*, task-oriented flow distillation loss and feature space geometry consistency loss are introduced to help our model achieve excellent performance.

3.1. IFRNet

Given two input frames I_0 and I_1 at adjacent time instances, video frame interpolation aims to synthesize an intermediate frame I_t, where $0 < t < 1$. To achieve this goal, proposed model performs a first extraction phase so as to retrieve a pyramid of features from each frame, then in a coarse-to-fine manner it progressively refines bilateral intermediate flow fields together with reconstructed intermediate feature until reaching the highest level of the pyramid to obtain the final output. Figure 3 sketches the overall architecture of proposed IFRNet.

Pyramid Encoder. To obtain contextual representation from each input frame, we design a compact encoder $\mathcal{E}$ to extract a pyramid of features. Purposely, the parameter shared encoder is built of a block of two 3×3 convolutions in each pyramid level, respectively with strides 2 and 1. As shown in Figure 3, IFRNet extracts 4 levels of pyramid features, counting 8 convolution layers, each followed by a PReLU activation [17]. By gradually decimating the spatial size, it increases the feature channels to 32, 48, 72 and 96, generating pyramid features ϕ_0^k, ϕ_1^k in level k ($k \in \{1, 2, 3, 4\}$) for frames I_0 and I_1, respectively.

Coarse-to-Fine Decoders. After extracting meaningful hierarchical representations, we then gradually refine intermediate flow fields through multiple decoders by backward warping pyramid features ϕ_0^k, ϕ_1^k to generate $\tilde{\phi}_0^k, \tilde{\phi}_1^k$ according to $F^k_{t\to0}$ and $F^k_{t\to1}$, respectively. The main advantage of

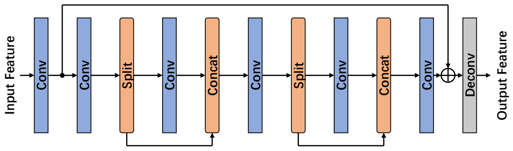


Figure 4. **Details of the decoder in each pyramid level.**

coarse-to-fine warping strategy consists of computing easier residual flow at each scale. Different from previous VFI approaches containing post-refinement [12, 20, 33, 37], we explore to improve the bilateral flow prediction during its coarse-to-fine procedure for higher efficiency. Specifically, we make each decoder $\mathcal{D}^{k+1}$ output a higher level reconstructed intermediate feature $\hat{\phi}_t^k$ besides bilateral flow fields $F_{t\rightarrow 0}^k, F_{t\rightarrow 1}^k$, which can fill up the missing reference information to facilitate motion estimation. On the other hand, better predicted flow fields $F_{t\rightarrow 0}^k, F_{t\rightarrow 1}^k$ will align source pyramid features to the target position more precisely, thus, generating better $\tilde{\phi}_0^k, \tilde{\phi}_1^k$, which can in turn improve higher level intermediate feature reconstruction. Therefore, decoders in proposed IFRNet can jointly refine bilateral intermediate flow fields together with reconstructed intermediate feature, benefitting each other until reaching desired output. Moreover, the gradually refined intermediate feature, containing bilateral occlusion and global context information, can finally generate fusion mask and compensate for motion details, that are often missing by flow-based methods, enabling IFRNet a powerful encoder-decoder VFI architecture without additional refinement [33, 37].

Concretely, in each pyramid level, we stack corresponding input features into a holistic volume that is forwarded by a compact decoder network $\mathcal{D}^k$, consisting of a block of six 3×3 convolutions and one 4×4 deconvolution, with strides 1 and $1/2$, respectively. A PReLU [17] follows each convolution layer. Details of each decoder is shown in Figure 4. In order to keep relative large receptive field and channel numbers for motion estimation and feature encoding while maintaining efficiency, we modify the third and the fifth convolution to update only partial channels of previous output tensor. Furthermore, residual connection and interlaced placement can promote information propagation and joint refinement. More details are shown in supplementary. Note that inputs of $\mathcal{D}^4$ and outputs of $\mathcal{D}^1$ are different from other decoders due to the task-related characteristics. In summary, features among decoders can be computed by

$$[F_{t\rightarrow 0}^3, F_{t\rightarrow 1}^3, \hat{\phi}_t^3] = \mathcal{D}^4([\phi_0^4, \phi_1^4, T]), \tag{1}$$

$$[F_{t\rightarrow 0}^{k-1}, F_{t\rightarrow 1}^{k-1}, \hat{\phi}_t^{k-1}] = \mathcal{D}^k([F_{t\rightarrow 0}^k, F_{t\rightarrow 1}^k, \hat{\phi}_t^k, \tilde{\phi}_0^k, \tilde{\phi}_1^k]), \tag{2}$$

$$[F_{t\rightarrow 0}, F_{t\rightarrow 1}, M, R] = \mathcal{D}^1([F_{t\rightarrow 0}^1, F_{t\rightarrow 1}^1, \hat{\phi}_t^1, \tilde{\phi}_0^1, \tilde{\phi}_1^1]), \tag{3}$$

where $\mathcal{D}^k (k = 2, 3)$ stand for decoders at middle pyramid levels, $[\cdot]$ denotes concatenation operation. T is a one-channel conditional input for arbitrary time interpolation, whose values are all the same and set to t. M is a one-channel merge mask exported by a sigmoid layer whose elements range from 0 to 1, and R is a three-channel image residual that can compensate for details. Finally, we can synthesize the desired frame $\hat{I}_t$ by following formulation

$$\hat{I}_t = M \odot \tilde{I}_0 + (1 - M) \odot \tilde{I}_1 + R, \tag{4}$$

$$\tilde{I}_0 = w(I_0, F_{t\rightarrow 0}),\ \tilde{I}_1 = w(I_1, F_{t\rightarrow 1}), \tag{5}$$

where w means backward warping, $\odot$ is element-wise multiplication. M adjusts the mixing ratio according to bidirectional occlusion information, while R compensates for some details when flow-based generation is unreliable, such as regions of target frame are occluded in both views.

Discussion with Optical Flow Networks. Different from the coarse-to-fine pipeline in real-time optical flow [24, 45] which mainly deals with large displacement matching challenge, in video interpolation, since the target frame is missing, its motion estimation becomes a "chicken-and-egg" problem. Therefore, decoders of IFRNet reconstruct intermediate feature besides intermediate flow fields, performing spatio-temporal feature aggregation and intermediate motion refinement jointly to benefit from each other.

Image Reconstruction Loss. According to above analysis, an efficient IFRNet has been designed for VFI, which is end-to-end trainable. For the purpose of generating intermediate frame, we employ the same image reconstruction loss $\mathcal{L}_r$ as [37] between network output $\hat{I}_t$ and ground truth frame I_t^{gt}, which is the sum of two terms and denoted by

$$\mathcal{L}_r = \rho(\hat{I}_t - I_t^{gt}) + \mathcal{L}_{cen}(\hat{I}_t, I_t^{gt}), \tag{6}$$

where $\rho(x) = (x^2 + \epsilon^2)^\alpha$ with $\alpha = 0.5, \epsilon = 10^{-3}$ is the Charbonnier loss [6] severing as a surrogate for the $\mathcal{L}_1$ loss. While $\mathcal{L}_{cen}$ is the census loss, which calculates the soft Hamming distance between census-transformed [31] image patches of size 7×7.

3.2. Task-Oriented Flow Distillation Loss

Training IFRNet with above reconstruction loss $\mathcal{L}_r$ can already perform intermediate frame synthesizing. However, the simple optimization target usually drops into local minimum, since illuminance cases are often challenging, *i.e.*, extreme brightness and repetitive texture regions. To deal with this problem, we try to adopt the knowledge distillation [18] strategy to guide multi-scale intermediate flow estimation of IFRNet by an off-the-shelf teacher flow network, that helps to align multi-scale pyramid features explicitly. In practice, the pre-trained teacher is only used during training, and we calculate its flow prediction as pseudo label $F_{t\rightarrow 0}^p, F_{t\rightarrow 1}^p$ in advance for efficiency. Note that RIFE [20] also uses flow distillation. However, their indiscriminate distillation manner usually learns undesired noise existed in pseudo label. Even if ground truth is available, optical flow itself is often a sub-optimal representation for specific video task [50]. To overcome above limitations, we propose task-oriented flow distillation loss that can decrease the adverse impacts while focusing on the useful knowledge for better VFI.

Observing that $F_{t\to0}, F_{t\to1}$ which directly control frame synthesis are sensitive to harmful information in pseudo label. Therefore, we impose multi-scale flow distillation except for the decoder $\mathcal{D}^1$, and leave its flow prediction totally constrained by the reconstruction loss $\mathcal{L}_r$ in a task-oriented manner [50]. Furthermore, we can compare above relaxed flow prediction $F_{t\to0}, F_{t\to1}$ with pseudo label $F^p_{t\to0}, F^p_{t\to1}$ to calculate robustness masks P_0, P_1, and use them to adjust the robustness of distillation loss spatially in lower multiple scales for better task-oriented flow distillation, whose procedure is depicted in Figure 3. Specifically, we can obtain $P_l (l \in \{0, 1\})$ by the following formulation

$$P_l = \exp(-\beta|F_{t\to l} - F^p_{t\to l}|_{epe}), \tag{7}$$

where $|\cdot|_{epe}$ calculates per-pixel end-point-error, the coefficient β controlling sensibility for robustness is set to 0.3 according to grid search. Foundation of above operations is based on the assumption that task-oriented flow generally agrees with true optical flow but differs in some details.

Following previous experience [21, 44], our task-oriented flow distillation employs the generalized Charbonnier loss $\rho(x) = (x^2 + \epsilon^2)^\alpha$ for better robust learning of intermediate flow, where parameters ϵ and α control the robustness of this loss. Formally, it can be written as

$$\mathcal{L}_d = \sum_{k=1}^{3}\sum_{l=0}^{1} \rho(\mathcal{U}_{2^k}(F^k_{t\to l}) - F^p_{t\to l}), \tag{8}$$

where $\mathcal{U}_s$ is the bilinear upsampling operation with scale factor s. However, different from the fixed format like previous methods [21, 44], we make it adjustable about VFI task by letting ϵ and α be functions of the robustness parameter p, where $p \in (0, 1]$ means the robustness value of any position in aforementioned robustness masks P_0, P_1. In general, we employ the linear and exponential linear functions to generate α and ϵ separately as follows

$$\alpha = p/2, \quad \epsilon = 10^{-(10p-1)/3}. \tag{9}$$

The coefficients are selected based on two typical cases. For example, when $p = 1.0$, $\rho(x)$ becomes the surrogate $\mathcal{L}_1$ loss in Eq. 6. And when $p = 0.4$, it turns to be the robust loss used in LiteFlowNet [21]. Figure 5 gives some intuitive examples of this adaptive robust loss. Comprehensively speaking, in each spatial location, if the task-oriented flow prediction of decoder $\mathcal{D}^1$ is consistent with that in pseudo label, the gradient of the adaptive distillation loss is relatively steep, which tends to distill this helpful information to the bottom three decoders by common gradient descent optimizer. On the other hand, the loss will become more robust to downgrade this relatively harmful flow knowledge.

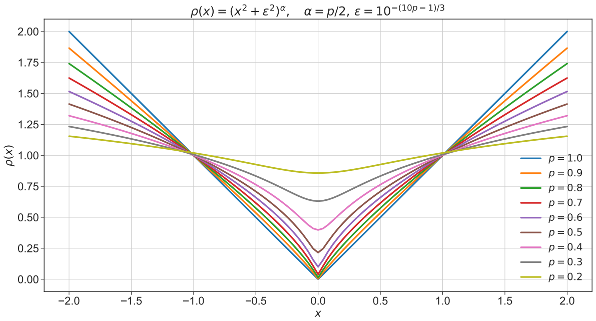


Figure 5. **Task-oriented flow distillation loss.** It takes the format of generalized Charbonnier loss, while the concrete form in each location is controlled by the corresponding robustness parameter p, which is determined by Eq. 7 to acquire task adaptive ability.

3.3. Feature Space Geometry Consistency Loss

Besides above task-oriented flow distillation loss for facilitating multi-scale intermediate flow estimation, better supervision of intermediate feature is preferred for further improvement. Observing that extracted pyramid features ϕ^k_0, ϕ^k_1 by the encoder $\mathcal{E}$, in a sense, play an equivalent role as the reconstructed intermediate feature $\hat{\phi}^k_t$ from the decoder $\mathcal{D}^{k+1}$, we try to employ the same parameter shared encoder $\mathcal{E}$ to extract a pyramid of features ϕ^k_t from ground truth frame I^{gt}_t, and use ϕ^k_t to regularize the reconstructed intermediate feature $\hat{\phi}^k_t$ in multi-scale feature domain.

Intuitively, we can adopt the commonly used $\mathcal{L}_1$ loss to restrict $\hat{\phi}^k_t$ to be close to ϕ^k_t. However, the overtighten constraint will harm the global context and occlusion information contained in reconstructed intermediate feature $\hat{\phi}^k_t$. To relax it and inspired by the local geometry alignment property of census transform [53], we extend the census loss $\mathcal{L}_{cen}$ [31] into multi-scale feature space for progressive supervision, where the soft Hamming distance is calculated between census-transformed corresponding feature maps with 3×3 patches in a channel-by-channel manner. Formally, this loss can be written as

$$\mathcal{L}_g = \sum_{k=1}^{3} \mathcal{L}_{cen}(\hat{\phi}^k_t, \phi^k_t). \tag{10}$$

Our motivation is that the extracted pyramid feature, containing useful low-level structure information for frame synthesizing, can regularize the reconstructed intermediate feature to keep better geometry layout. For each spatial location, $\mathcal{L}_g$ only constrain the geometry of its neighbor local patch in every feature map. Consequently, there is no restriction on the channel-wise representation for $\hat{\phi}^k_t$ to encode bilateral occlusion and residual information.

Based on above analysis, our final loss function, containing three parts for joint optimization, is formulated as

$$\mathcal{L} = \mathcal{L}_r + \lambda\mathcal{L}_d + \eta\mathcal{L}_g, \tag{11}$$

where weighting parameters are set to $\lambda = 0.01, \eta = 0.01$.

Method	Vimeo90K	UCF101	SNU-FILM				Time	Params	FLOPs
			Easy	Medium	Hard	Extreme	(s)	(M)	(T)
SepConv [35]	33.79/0.9702	34.78/0.9669	39.41/0.9900	34.97/0.9762	29.36/0.9253	24.31/0.8448	0.065	21.7	0.36
CAIN [10]	34.65/0.9730	34.91/0.9690	39.89/0.9900	35.61/0.9776	29.90/0.9292	24.78/0.8507	0.069	42.8	1.29
AdaCoF [25]	34.47/0.9730	34.90/0.9680	39.80/0.9900	35.05/0.9754	29.46/0.9244	24.31/0.8439	0.054	21.8	0.36
RIFE [20]	35.62/0.9780	35.28/0.9690	40.06/**0.9907**	35.75/0.9789	30.10/0.9330	24.84/0.8534	0.026	9.8	0.20
IFRNet	35.80/0.9794	35.29/0.9693	40.03/0.9905	35.94/0.9793	30.41/0.9358	25.05/0.8587	0.025	5.0	0.21
IFRNet small	35.59/0.9786	35.28/0.9691	39.96/0.9905	35.92/0.9792	30.36/0.9357	25.05/0.8582	**0.019**	2.8	**0.12**
ToFlow [50]	33.73/0.9682	34.58/0.9667	39.08/0.9890	34.39/0.9740	28.44/0.9180	23.39/0.8310	0.152	**1.4**	0.62
CyclicGen [28]	32.09/0.9490	35.11/0.9684	37.72/0.9840	32.47/0.9554	26.95/0.8871	22.70/0.8083	0.161	19.8	1.77
DAIN [3]	34.71/0.9756	34.99/0.9683	39.73/0.9902	35.46/0.9780	30.17/0.9335	25.09/0.8584	1.033	24.0	5.51
SoftSplat [33]	36.10/0.9700	35.39/0.9520	-	-	-	-	0.195	12.2	0.90
BMBC [36]	35.01/0.9764	35.15/0.9689	39.90/0.9902	35.31/0.9774	29.33/0.9270	23.92/0.8432	3.845	11.0	2.50
CDFI full [12]	35.17/0.9640	35.21/0.9500	**40.12**/0.9906	35.51/0.9778	29.73/0.9277	24.53/0.8476	0.380	5.0	0.82
ABME [37]	36.18/0.9805	35.38/**0.9698**	39.59/0.9901	35.77/0.9789	30.58/0.9364	**25.42**/**0.8639**	0.905	18.1	1.30
IFRNet large	**36.20**/**0.9808**	**35.42**/**0.9698**	40.10/0.9906	**36.12**/**0.9797**	**30.63**/**0.9368**	25.27/0.8609	0.079	19.7	0.79

Table 1. **Quantitative comparison (PSNR/SSIM) of VFI results on the Vimeo90K, UCF101 and SNU-FILM datasets.** For each item, the best result is **boldfaced**, and the second best is underlined. Top and bottom parts are divided by running time.

4. Experiments

In this section, we first introduce implementation details and datasets used in this paper. Then, we quantitatively and qualitatively compare IFRNet with recent state-of-the-arts on various benchmarks. Finally, ablation studies are carried out to analyze the contribution of proposed approaches. Experiments in the main paper follow a common practice of $t = 0.5$, that is synthesizing the single middle frame. IFRNet also supports multi-frame interpolation with temporal encoding T, whose results are presented in supplementary.

4.1. Implementation Details

We implement proposed algorithm in PyTorch, and use Vimeo90K [50] training set to train IFRNet from scratch. Our model is optimized by AdamW [30] algorithm for 300 epochs with total batch size 24 on four NVIDIA Tesla V100 GPUs. The learning rate is initially set to 1×10^{-4}, and gradually decays to 1×10^{-5} following a cosine attenuation schedule. During training, we augment the samples by random flipping, rotating, reversing sequence order and random cropping patches with size 224×224. For optical flow distillation, we extract pseudo label of bilateral intermediate flow fields with the pre-trained LiteFlowNet [21] in advance, and perform consistent augmentation operations with frame triplets during the whole training process.

4.2. Evaluation Metrics and Datasets

We evaluate our method on various datasets covering diverse motion scenes for comprehensive comparison. Common metrics, such as PSNR and SSIM [47] are adopted for quantitative evaluation. For Middlebury, we use the official IE and NIE indices. Now, we briefly introduce the used test datasets to assess our approaches.

Vimeo90K [50]: It contains frame triplets of 448×256 resolution. There are 3,782 triplets consisted in the test part.

UCF101 [43]: We adopt the test set selected in DVF [29], which includes 379 triplets of 256×256 frame size.

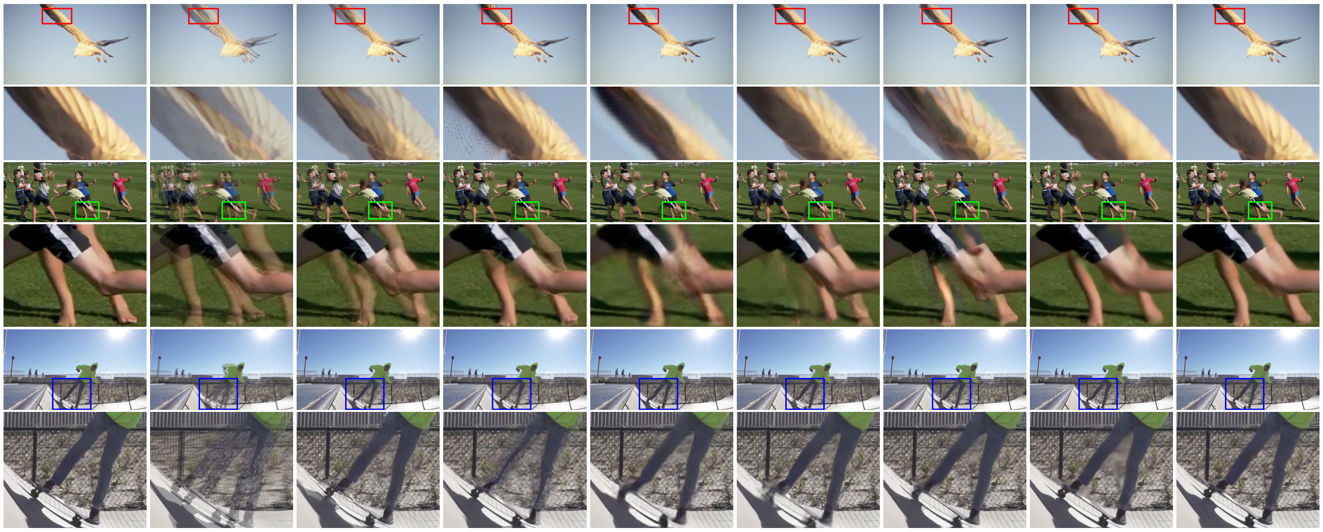
SNU-FILM [10]: SNU-FILM contains 1,240 frame triplets of approximate 1280×720 resolution. According to motion magnitude, it is divided into four different parts, namely, Easy, Medium, Hard, and Extreme for detailed comparison.

Middlebury [2]: The Middlebury benchmark is a widely used dataset to evaluate optical flow and VFI methods. Image resolution in this dataset is around 640×480. In this paper, we test on the Evaluation set without using Other set.

4.3. Comparison with the State-of-the-Arts

We compare IFRNet with state-of-the-art VFI methods, including kernel-based SepConv [35], AdaCoF [25] and CDFI [12], flow-based ToFlow [50], DAIN [3], SoftSplat [33], BMBC [36], RIFE [20] and ABME [37], and hallucination-based CAIN [10] and FeFlow [16]. For results on SNU-FILM, we execute the released codes of CDFI and RIFE and refer to the other results tested in ABME. For Middlebury, we directly test on the Evaluation part and submit interpolation results to the online benchmark. To measure the inference speed and computation complexity, we run all methods on one Tesla V100 GPU under 1280×720 resolution and average the running time with 100 iterations. For fair comparison, we further build a large and a small version of IFRNet by scaling feature channels with 2.0 and 0.75, respectively, and separate above methods into two classes, *i.e.*, *fast* and *slow*, according to their inference time.

Quantitative Evaluation. Table 1 and Table 2 summarize quantitative results on diverse benchmarks. On Vimeo90K and UCF101 test datasets, IFRNet large achieves the best results on both PSNR and SSIM metrics. A recent method ABME [37] also gets similar accuracy. However, our model runs **11.5** × faster with similar amount of parameters due to the efficiency of single encoder-decoder based architecture. Our large model also obtains leading results on the Easy,



(a) Ground Truth (b) Overlaid (c) SepConv [35] (d) DAIN [3] (e) CAIN [10] (f) AdaCoF [25] (g) CDFI [12] (h) ABME [37] (i) Ours

Figure 6. **Qualitative comparison of different VFI methods on SNU-FILM (Hard) dataset.** Proposed IFRNet algorithm can synthesize fast moving objects with sharp boundary while maintaining distinct contextual details. Zoom in for best view.

Method	Average		Mequon		Schefflera		Urban		Teddy		Backyard		Basketball		Dumptruck		Evergreen	
	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE	IE	NIE
SuperSlomo [22]	5.310	0.778	2.51	0.59	3.66	0.72	2.91	0.74	5.05	0.98	9.56	0.94	5.37	0.96	6.69	0.60	6.73	0.69
ToFlow [50]	5.490	0.840	2.54	0.55	3.70	0.72	3.43	0.92	5.05	0.96	9.84	0.97	5.34	0.98	6.88	0.72	7.14	0.90
DAIN [3]	4.856	0.713	2.38	0.58	3.28	0.60	3.32	0.69	4.65	0.86	7.88	0.87	4.73	0.85	6.36	0.59	6.25	0.66
FeFlow [16]	4.820	0.719	2.28	**0.51**	3.50	0.66	2.82	0.70	4.75	0.87	7.62	**0.84**	4.74	0.86	6.07	0.64	6.78	0.67
AdaCoF [25]	4.751	0.730	2.41	0.60	3.10	0.59	3.48	0.84	4.84	0.92	8.68	0.90	4.13	0.84	5.77	0.58	5.60	0.57
BMBC [36]	4.479	0.696	2.30	0.57	3.07	0.58	3.17	0.77	4.24	0.84	7.79	0.85	**4.08**	0.82	5.63	0.58	5.55	**0.56**
SoftSplat [33]	4.223	0.645	**2.06**	0.53	2.80	0.52	1.99	0.52	**3.84**	**0.80**	8.10	0.85	4.10	**0.81**	**5.49**	**0.56**	**5.40**	0.57
IFRNet large	**4.216**	**0.644**	2.08	0.53	**2.78**	**0.51**	**1.74**	**0.43**	3.96	0.83	**7.55**	0.87	4.42	0.84	5.56	**0.56**	5.64	0.58

Table 2. **Evaluation results on the Middlebury benchmark.** For each item, the best result is **boldfaced**, and the second best is underlined.

Medium and Hard parts of SNU-FILM datasets, while only falls behind ABME on the Extreme part. We attribute the reason to be that the bilateral cost volume constructed by ABME is good at estimating large displacement motion. In Table 2, IFRNet large achieves top-performing VFI accuracy in most of the eight Middlebury test sequences, and outperforms the previous state-of-the-art SoftSplat [33] on both average IE and NIE metrics. Although the improvement is limited, our approach runs **2.5** × faster than SoftSplat which takes cascaded VFI architecture. For FLOPs in convolution layers, IFRNet large also consumes significantly less computation than other VFI architectures.

In regard to real-time and lightweight VFI approaches, IFRNet yields about 0.2 dB better result than RIFE [20] on Vimeo90K, and the margin is more distinct on large motion cases in SNU-FILM dataset. It is worth noting that IFRNet only contains **half** parameters to achieve better results than RIFE thanks to the superiority of joint refinement of intermediate flow and context feature. Compared with CDFI full [12], IFRNet has the same 5M parameters, while achieving **0.63** dB higher PSNR on Vimeo90K with **15.2** × faster inference speed. Moreover, IFRNet small can further improve speed by **31**% and reduce parameters and computation complexity by **44**% than IFRNet while with only slight frame interpolation accuracy decrease.

Qualitative Evaluation. Figure 6 visually compares well-behaved VFI methods on SNU-FILM (Hard) dataset which contains large and complex motion scenes. It can be seen that kernel-based [12, 25, 35] and hallucination-based [10] methods fail to synthesize sharp motion boundary, containing ghost and blur artifacts. Compared with flow-based algorithms [3, 37], our approach can generate texture details faithfully thanks to the powerfulness of gradually refined intermediate feature. In short, IFRNet can synthesize pleasing target frame with more comfortable visual experience. More qualitative results can be found in our supplementary.

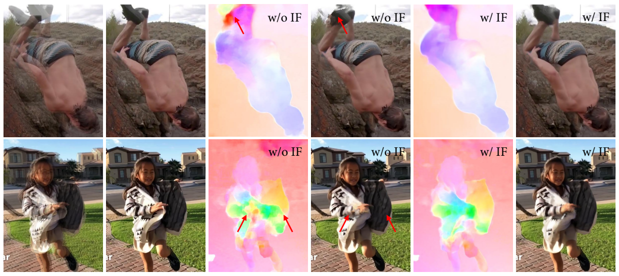
4.4. Ablation Study

To verify the effectiveness of proposed approaches, we carry out ablation study in terms of network architecture and loss function on Vimeo90K and SNU-FILM Hard datasets.

Intermediate Feature. To ablate the effectiveness of intermediate feature $\hat{\phi}_t^k$ in IFRNet, we build a model by removing $\hat{\phi}_t^k$ from the input and output of multiple decoders, while

Architecture		Vimeo90K	Hard
IF	R	PSNR	PSNR
✗	✗	34.83	29.96
✓	✗	35.22	30.22
✗	✓	35.11	30.06
✓	✓	**35.51**	**30.27**

Table 3. **Ablation study on different architecture variants.** 'IF' means intermediate feature $\hat{\phi}_t^k$ and 'R' stands for residual R.



(a) Overlaid (b) GT (c) $F_{t\to0}$ (d) Pred (e) $F_{t\to0}$ (f) Pred

Figure 7. **Visual comparison of intermediate flow and predicted frame of IFRNet w/o and w/ intermediate feature.**

keeping feature channels of middle parts of decoders unchanged. Also, we selectively remove residual R in Eq. 4 to isolate the improvement from intermediate flow and residual. We train them with only the reconstruction loss $\mathcal{L}_r$ under the same learning schedule as before. As listed in Table 3, from the first two rows, we can observe that intermediate feature can provide reference anchor information to promote intermediate flow estimation. Figure 7 also presents some visual examples to confirm the conclusion. Compared with the last and the second rows in Table 3, it demonstrates that gradually refined intermediate feature, containing global context information, can compensate better scene details. Conclusively, residual compensation from the intermediate context feature is necessary for IFRNet to achieve advanced VFI performance, since intermediate flow prediction is substantively unreliable. Overall, the two-fold benefits from intermediate feature greatly improves VFI accuracy of IFRNet with relatively small additional cost.

Task-Oriented Flow Distillation. Table 4 compares VFI accuracy under different combinations of proposed loss functions quantitatively. It can be seen that adding task-oriented flow distillation loss $\mathcal{L}_d$ consistently improves PSNR of 0.2 dB on Vimeo90K. To verify the superiority of its task adaptive ability, we also perform flow distillation with generalized Charbonnier loss under different robustness shown in Figure 5, whose results are summarized in Figure 8. It turns out that robustness parameter $p = 0.3$ achieves best VFI accuracy in the fixed robustness setting. On the other hand, flow distillation can damage frame quality when p approaches to 1.0 due to the harmful knowledge in pseudo label. In a word, proposed task-oriented approach achieves the best accuracy thanks to its spatial adaptive ability for adjusting robustness loss during flow distillation.

Loss Function			Vimeo90K	Hard
$\mathcal{L}_r$	$\mathcal{L}_d$	$\mathcal{L}_g$	PSNR	PSNR
✓	✗	✗	35.51	30.27
✓	✓	✗	35.72	30.38
✓	✗	✓	35.61	30.30
✓	✓	✓	**35.80**	**30.41**

Table 4. **Ablation study on different loss functions.**

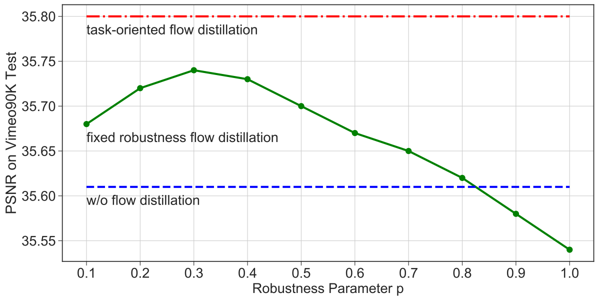


Figure 8. **Ablation study on different flow distillation losses.**

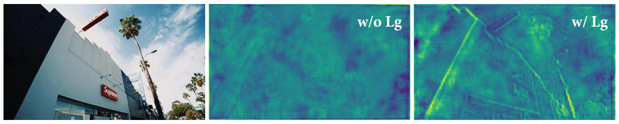


Figure 9. **Visual comparison of mean feature map of intermediate feature $\hat{\phi}_t^1$ w/o and w/ $\mathcal{L}_g$.** Leftmost is the ground truth.

Feature Space Geometry Consistency. As shown in Table 4, adding proposed feature space geometry consistency loss $\mathcal{L}_g$ based on above contributions, we can obtain a further improvement, that confirms the complementary effect of $\mathcal{L}_g$ in regard to $\mathcal{L}_d$. Figure 9 visually compares mean feature maps of intermediate feature $\hat{\phi}_t^1$ w/o and w/ $\mathcal{L}_g$. It shows that $\mathcal{L}_g$ can regularize the reconstructed intermediate feature to keep better geometry layout in multi-scale feature space, resulting in better VFI performance.

5. Conclusion

In this paper, we have devised an efficient deep architecture, termed IFRNet, for video frame interpolation, without any cascaded synthesis or refinement module. It gradually refines intermediate flow together with a powerful intermediate feature, that can not only boost intermediate flow estimation to synthesize sharp motion boundary but also provide global context representation to generate vivid motion details. Moreover, we have presented task-oriented flow distillation loss and feature space geometry consistency loss to fully release its potential. Experiments on various benchmarks demonstrate the state-of-the-art performance and fast inference speed of proposed approaches. We expect proposed single encoder-decoder joint refinement based IFRNet to be a useful component for many frame rate upconversion and intermediate view synthesis systems.

References

[1] Filippo Aleotti, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning end-to-end scene flow by distilling single tasks knowledge. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 3

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 2011. 6

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 6, 7

[4] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[5] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[6] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, 1994. 4

[7] Xianhang Cheng and Zhenzhong Chen. Video frame interpolation via deformable separable convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[8] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[9] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N. Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *Computer Vision – ECCV 2020*, 2020. 1, 2

[10] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2, 6, 7

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[12] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. Cdfi: Compression-driven network design for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 4, 6, 7

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 3

[14] Karl M. Fant. A nonaliasing, real-time spatial transform technique. *IEEE Computer Graphics and Applications*, 1986. 2

[15] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. In *Proceedings of the British Machine Vision Conference, 2017*, 2017. 2

[16] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6, 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 3, 4

[18] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 3, 4

[19] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981. 3

[20] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation. *CoRR*, 2021. 1, 2, 3, 4, 6, 7

[21] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 5, 6

[22] Huaizu Jiang, Deqing Sun, Varan Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 7

[23] Soo Kim, Jihyong Oh, and Munchurl Kim. Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[24] Lingtong Kong, Chunhua Shen, and Jie Yang. Fastflownet: A lightweight network for fast optical flow estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 1, 3, 4

[25] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6, 7

[26] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. 3

[27] Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[28] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. 6

[29] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 6

[30] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*, 2019. 6

[31] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 4, 5

[32] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[33] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4, 6, 7

[34] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[35] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 6, 7

[36] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *European Conference on Computer Vision*, 2020. 1, 2, 6, 7

[37] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 4, 6, 7

[38] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[39] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[40] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[41] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. 3

[42] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[43] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012. 6

[44] Deqing Sun, Stefan Roth, and Michael J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 2014. 5

[45] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3, 4

[46] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision – ECCV 2020*, 2020. 1

[47] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 6

[48] George Wolberg, H. M. Sueyllam, M. A. Ismail, and K. M. Ahmed. One-dimensional resampling with inverse and forward mapping functions. *Journal of Graphics Tools*, 2000. 2

[49] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems*, 2019. 1, 2

[50] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 2019. 1, 2, 5, 6, 7

[51] Liangzhe Yuan, Yibo Chen, Hantian Liu, Tao Kong, and Jianbo Shi. Zoom-in-to-check: Boosting video interpolation via instance-level discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[52] Yuan Yuan, Wei Su, and Dandan Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[53] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision — ECCV '94*, 1994. 5

[54] Haoxian Zhang, Yang Zhao, and Ronggang Wang. A flexible recurrent residual pyramid network for video frame interpolation. In *Computer Vision – ECCV 2020*, 2020. 1, 2

[55] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In *Computer Vision – ECCV 2016*, 2016. 1

IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation Supplementary Material

Lingtong Kong[1]*, Boyuan Jiang[2]*, Donghao Luo[2], Wenqing Chu[2], Xiaoming Huang[2],
Ying Tai[2], Chengjie Wang[2], Jie Yang[1]†
[1]Shanghai Jiao Tong University, China, [2]Youtu Lab, Tencent
{ltkong, jieyang}@sjtu.edu.cn
{byronjiang, michaelluo, wenqingchu, skyhuang, yingtai, jasoncjwang}@tencent.com

Figure 10. **Qualitative results of IFRNet for 8× interpolation on GoPro [9] and Adobe240 [13] test datasets.** Please watch the video with Adobe Reader. Each video has 9 frames where the first and the last frames are input, and the middle 7 frames are predicted by IFRNet.

In the supplementary, we first present multi-frame interpolation experiments of IFRNet. Second, qualitative video comparisions with other advanced VFI approaches are displayed. Third, we depict structure details of IFRNet and its variants. Fourth, we provide more visual examples and analysis of middle components for better understanding the workflow of IFRNet. Finally, we show the screenshot of VFI results on the Middlebury benchmark. Please note that the numbering within this supplementary has manually been adjusted to continue the ones in our main paper.

6. Multi-Frame Interpolation

Different from other multi-frame interpolation methods which scales optical flow [1, 5] or interpolates middle frames recursively [2, 7], IFRNet can predict multiple intermediate frames by proposed one-channel temporal encoding mask T, which is one of the input of the coarsest decoder $\mathcal{D}^4$. The temporal encoding is a conditional input signal whose values are all the same and set to t, where $t \in \{1/8, 2/8, \ldots, 7/8\}$ in 8× interpolation setting. Also, proposed task-oriented flow distillation loss and feature space geometry consistency loss still work for any intermediate time instance t. To evaluate IFRNet for 8× interpolation, we use the train/test split of FLAVR [6], where we train IFRNet on GoPro [9] training set with the same learning schedule and loss functions as our main paper. Then we test the pre-trained model on GoPro testing and Adobe240 [13] datasets whose results are listed in Table 5.

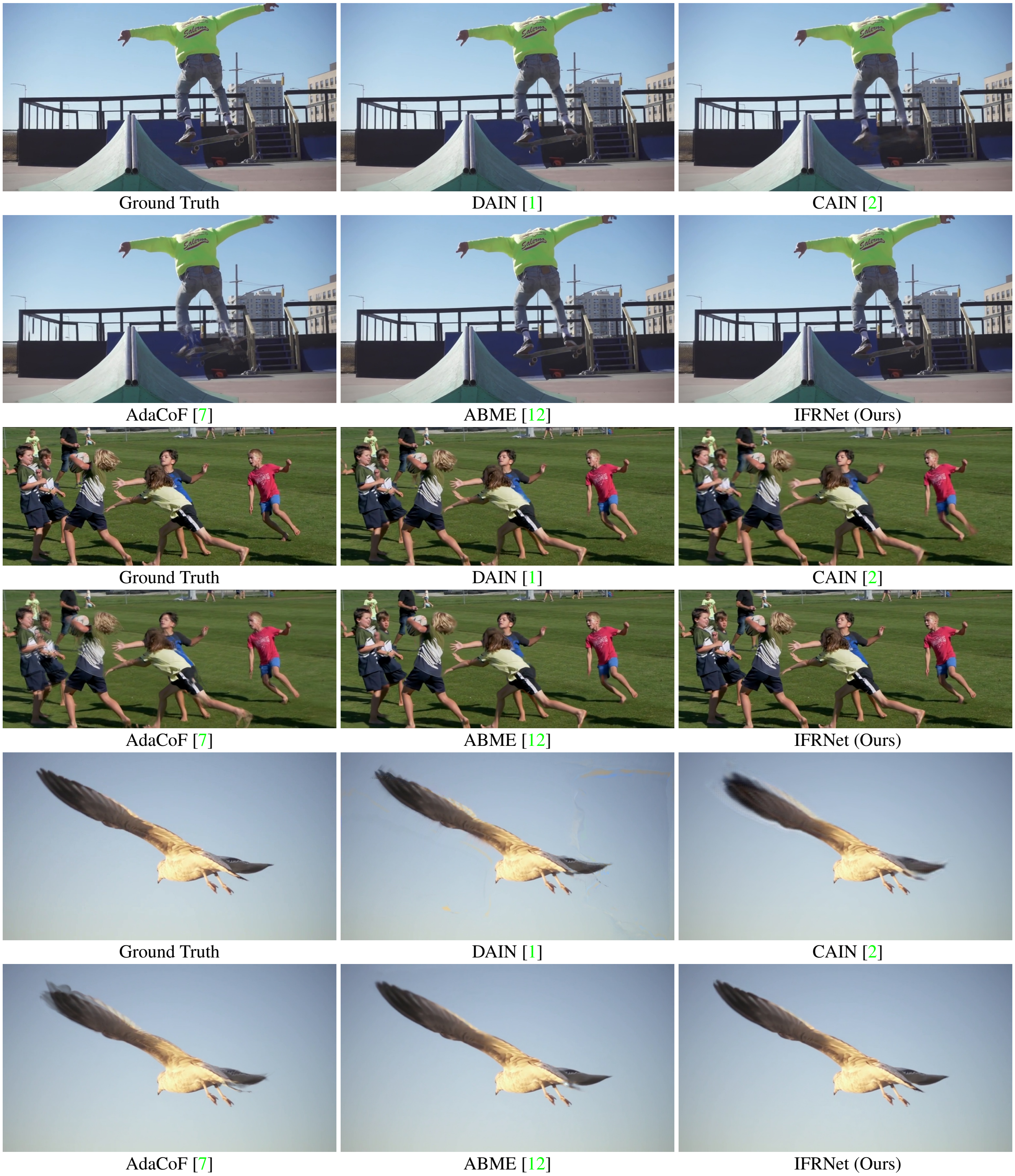
Method	GoPro [9]		Adobe240 [13]		Time
	PSNR	SSIM	PSNR	SSIM	(s)
DVF [8]	21.94	0.776	28.23	0.896	0.87
SuperSloMo [5]	28.52	0.891	30.66	0.931	0.44
DAIN [1]	29.00	0.910	29.50	0.910	4.10
IFRNet (Ours)	**29.84**	**0.920**	**31.93**	**0.943**	**0.16**

Table 5. **Quantitative comparison for 8× interpolation.**

IFRNet outperforms all of the other SOTA methods

* Equal contribution. This work was done when Lingtong Kong was an intern at Tencent Youtu Lab. Code is available at https://github.com/ltkong218/IFRNet.

† Corresponding author: Jie Yang (jieyang@sjtu.edu.cn). This research is partly supported by NSFC, China (No: 61876107, U1803261).



Ground Truth DAIN [1] CAIN [2]

AdaCoF [7] ABME [12] IFRNet (Ours)

Ground Truth DAIN [1] CAIN [2]

AdaCoF [7] ABME [12] IFRNet (Ours)

Ground Truth DAIN [1] CAIN [2]

AdaCoF [7] ABME [12] IFRNet (Ours)

Figure 11. **Video comparison on SNU-FILM [2] dataset**. Please watch the video with Adobe Reader and zoom in for best view.

with 2 input frames on both GoPro and Adobe240 datasets in both PSNR and SSIM metrics. For example, IFRNet achieves **0.84** dB better results than DAIN [1] on GoPro and exceeds SuperSloMo [5] by **1.27** dB on Adobe240. Thanks to the modularity character of IFRNet, the encoder only needs a single forward pass, while the decoders infer 7 times with different temporal embedding to convert videos from 30 fps into 240 fps. Therefore, the speed advantage of IFRNet is still or even more obvious than other approaches. Figure 10 gives some qualitative results of IFRNet for 8× interpolation, demonstrating its superior ability for frame rate up-conversion and slow motion generation.

7. Video Comparison

In this part, we qualitatively compare interpolated videos by proposed IFRNet against other open source VFI methods on SNU-FILM [2] dataset, whose results are shown in Figure 11. As can be seen, our approach can generate motion boundary and texture details faithfully thanks to the powerfulness of gradually refined intermediate feature.

8. Network Architecture

In this section, we present the structure details of five sub-networks of IFRNet, *i.e.*, pyramid encoder $\mathcal{E}$ and coarse-to-fine decoders $\mathcal{D}^4, \mathcal{D}^3, \mathcal{D}^2, \mathcal{D}^1$. In each following figure, arguments of 'Conv' and 'Deconv' from left to right are input channels, output channels, kernel size, stride and padding, respectively. Dimensions of input and output tensors from left to right stand for feature channels, height and width, separately. A PReLU [4] follows each 'Conv' layer, while there is no activation after each 'Deconv' layer. In practice, the intermediate flow fields are estimated in a residual manner, which is not reflected in the figures to emphasize the primary network structure. We take input frames with spatial size of 640×480 as example.

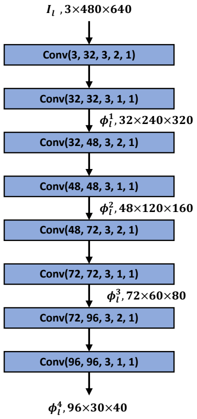


Figure 12. **Details of the pyramid encoder $\mathcal{E}$.** The two input frames $I_l, l \in \{0, 1\}$ are encoded by the same Siamese network.

As for IFRNet large and IFRNet small, feature channels from the first to the fourth pyramid levels are set to 64, 96, 144, 192 and 24, 36, 54, 72, respectively. Correspondingly, channel numbers in multiple decoders are adjusted. Also, feature channels of the third and the fifth convolution layers in coarse-to-fine decoders of IFRNet large and IFRNet small are set to 64 and 24, separately.

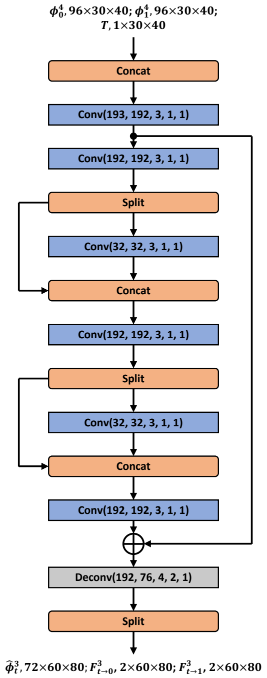


Figure 13. **Details of the bottom decoder $\mathcal{D}^4$.**

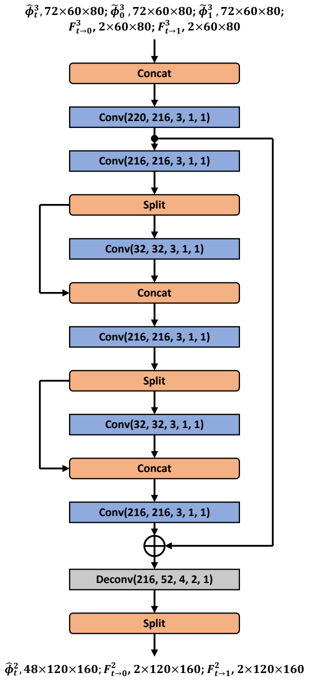

$\widehat{\phi}_t^3$, 72×60×80; $\widetilde{\phi}_0^3$, 72×60×80; $\widetilde{\phi}_1^3$, 72×60×80;
$F_{t\to0}^3$, 2×60×80; $F_{t\to1}^3$, 2×60×80
Concat
Conv(220, 216, 3, 1, 1)
Conv(216, 216, 3, 1, 1)
Split
Conv(32, 32, 3, 1, 1)
Concat
Conv(216, 216, 3, 1, 1)
Split
Conv(32, 32, 3, 1, 1)
Concat
Conv(216, 216, 3, 1, 1)
Deconv(216, 52, 4, 2, 1)
Split
$\widehat{\phi}_t^2$, 48×120×160; $F_{t\to0}^2$, 2×120×160; $F_{t\to1}^2$, 2×120×160

Figure 14. **Details of the middle decoder $\mathcal{D}^3$.**

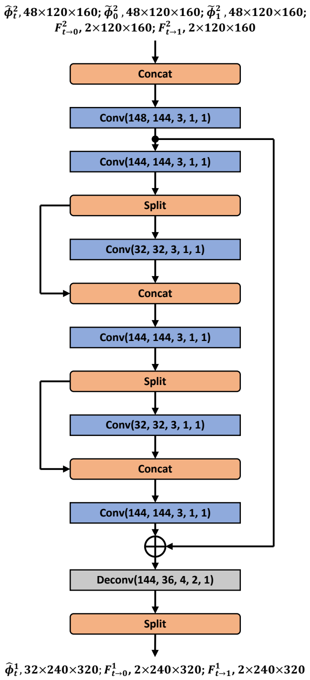

$\widehat{\phi}_t^2$, 48×120×160; $\widetilde{\phi}_0^2$, 48×120×160; $\widetilde{\phi}_1^2$, 48×120×160;
$F_{t\to 0}^2$, 2×120×160; $F_{t\to 1}^2$, 2×120×160
Concat
Conv(148, 144, 3, 1, 1)
Conv(144, 144, 3, 1, 1)
Split
Conv(32, 32, 3, 1, 1)
Concat
Conv(144, 144, 3, 1, 1)
Split
Conv(32, 32, 3, 1, 1)
Concat
Conv(144, 144, 3, 1, 1)
Deconv(144, 36, 4, 2, 1)
Split
$\widehat{\phi}_t^1$, 32×240×320; $F_{t\to 0}^1$, 2×240×320; $F_{t\to 1}^1$, 2×240×320

Figure 15. **Details of the middle decoder $\mathcal{D}^2$.**

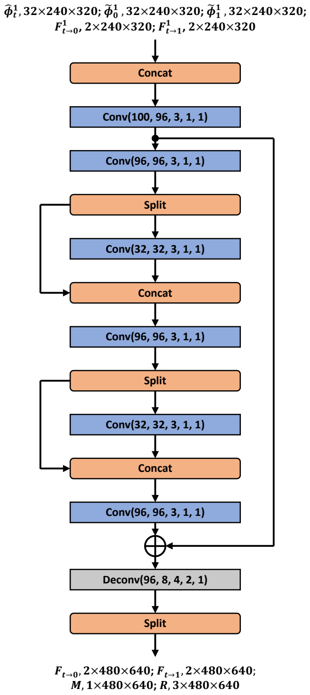

$\widehat{\phi}_t^1$, 32×240×320; $\widetilde{\phi}_0^1$, 32×240×320; $\widetilde{\phi}_1^1$, 32×240×320;
$F_{t\to 0}^1$, 2×240×320; $F_{t\to 1}^1$, 2×240×320
Concat
Conv(100, 96, 3, 1, 1)
Conv(96, 96, 3, 1, 1)
Split
Conv(32, 32, 3, 1, 1)
Concat
Conv(96, 96, 3, 1, 1)
Split
Conv(32, 32, 3, 1, 1)
Concat
Conv(96, 96, 3, 1, 1)
Deconv(96, 8, 4, 2, 1)
Split
$F_{t\to 0}$, 2×480×640; $F_{t\to 1}$, 2×480×640;
M, 1×480×640; R, 3×480×640

Figure 16. **Details of the top decoder $\mathcal{D}^1$.**

9. Visualization and Discussion

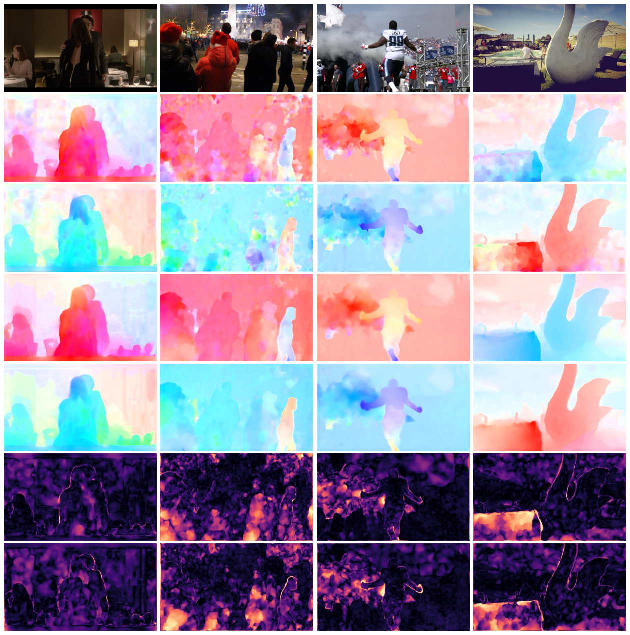


Figure 17. **Illustration of task-oriented flow distillation.** From top to bottom rows are ground truth frame I_t^{gt}, pseudo label of intermediate flow fields $F_{t\to 0}^p, F_{t\to 1}^p$, predicted intermediate flow fields $F_{t\to 0}, F_{t\to 1}$, task-oriented robustness masks P_0, P_1. Darker color in P_0, P_1 approaches to 1, while brighter color tends to 0. Each column represents a separate example on Vimeo90K [15] dataset. Zoom in for best view.

Figure 17 presents some visual examples to show the robustness masks in proposed task-oriented flow distillation loss, which can decrease the adverse impacts while focusing on the useful knowledge for better frame interpolation. It seems that intermediate flow prediction of IFRNet behaves smoother and contains less artifacts than flow prediction of pseudo label, that helps to achieve better VFI accuracy.

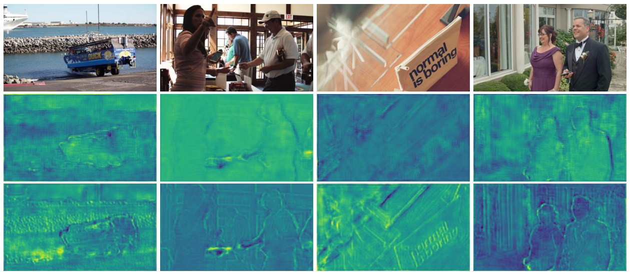


Figure 18. **Illustration of mean feature map of intermediate feature $\hat{\phi}_t^1$ w/o and w/ $\mathcal{L}_g$.** From top to bottom rows are ground truth frame I_t^{gt}, mean feature map of $\hat{\phi}_t^1$ w/o $\mathcal{L}_g$, mean feature map of $\hat{\phi}_t^1$ w/ $\mathcal{L}_g$. Each column represents a separate example on Vimeo90K [15] dataset. Zoom in for best view.

Figure 18 depicts more visual results of mean feature maps of intermediate feature w/o and w/ proposed geometry consistency loss, demonstrating its effect on regularizing refined intermediate feature to keep better structure layout.

Figure 19 gives visual understanding of frame interpola-

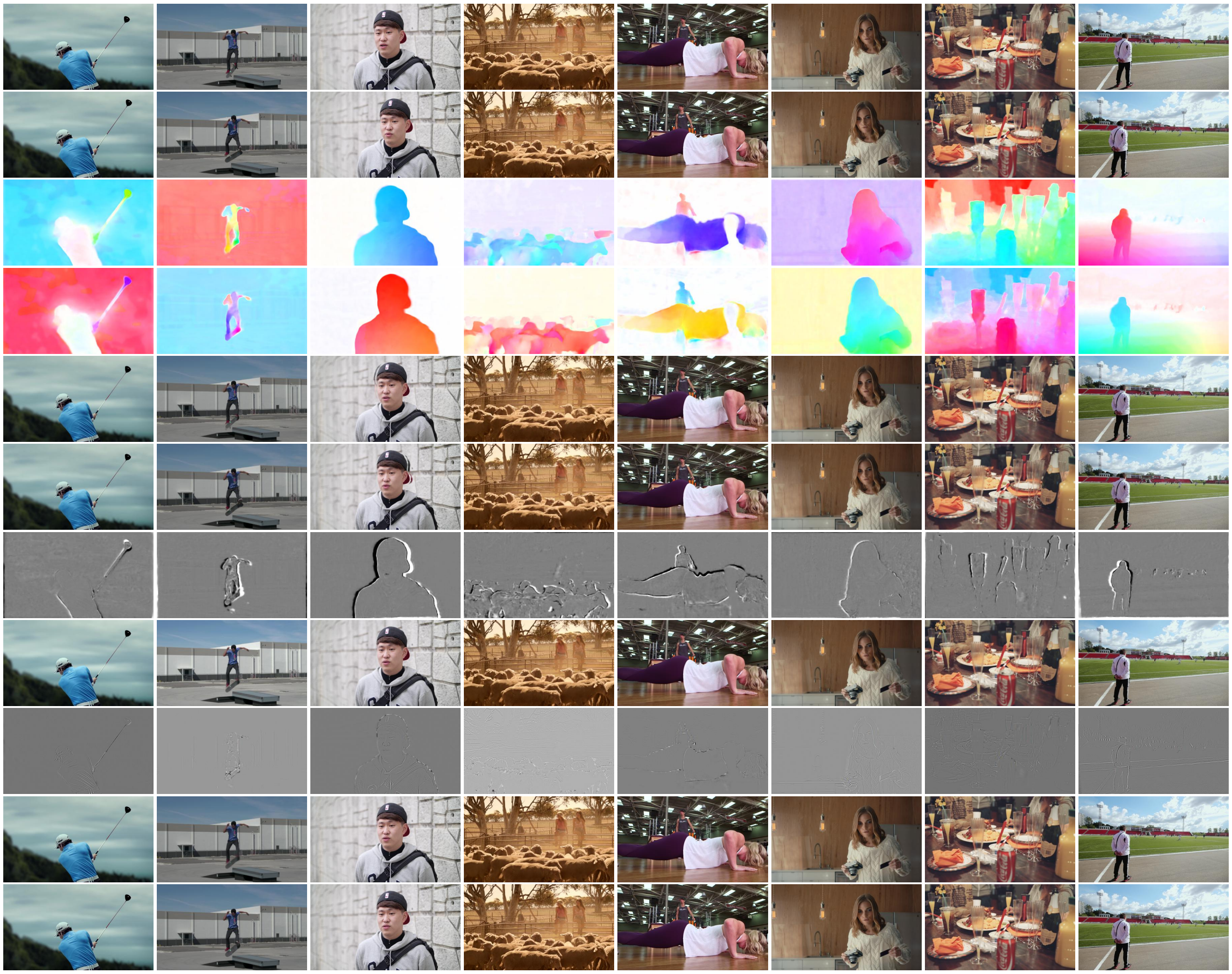
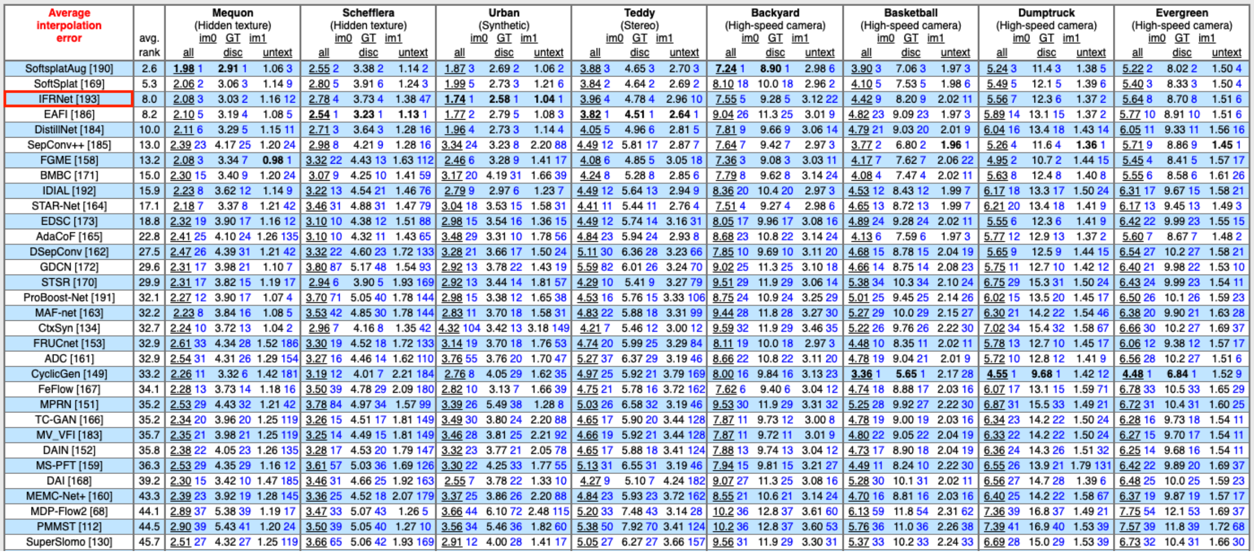


Figure 19. **Illustration of intermediate components of IFRNet.** From top to bottom rows are input frames I_0, I_1, predicted intermediate flow fields $F_{t\rightarrow 0}, F_{t\rightarrow 1}$, warped input frames $\tilde{I}_0, \tilde{I}_1$, merge mask M, merged frame $\hat{I}_t^{'}$, residual R, final prediction $\hat{I}_t$ and ground truth I_t^{gt}, where merged frame is calculated by $\hat{I}_t^{'} = M \odot \tilde{I}_0 + (1 - M) \odot \tilde{I}_1$. For better visualization of residual R, we multiply it by 10 and add a bias of 0.5. Each column represents a separate example on Vimeo90K [15] dataset. Zoom in for best view.

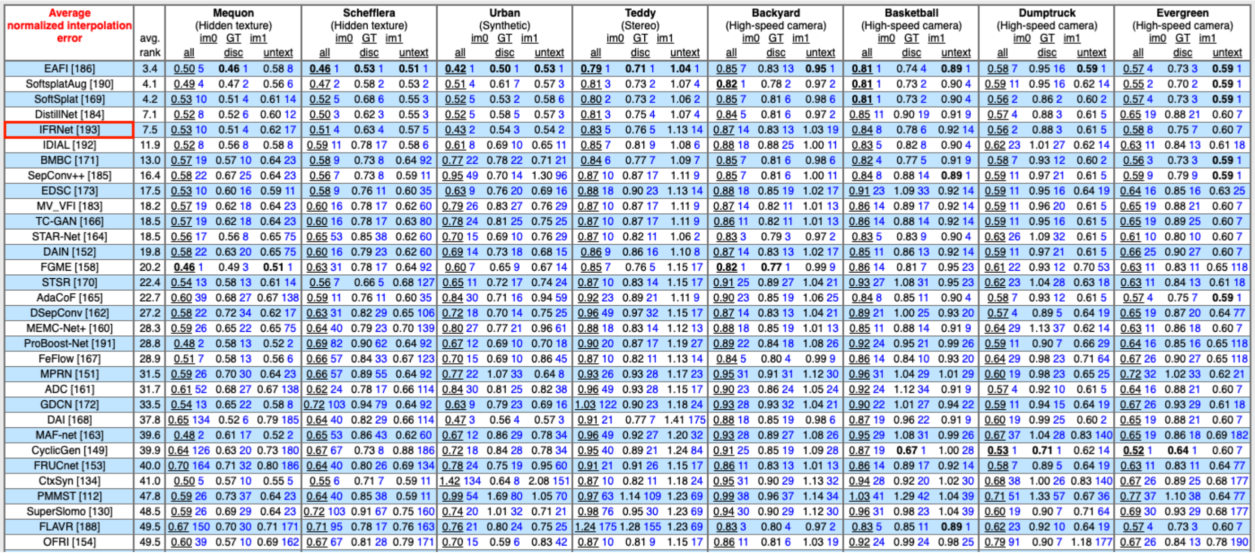
tion process of IFRNet. Thanks to the reference anchor information offered by intermediate feature together with effective supervision provided by geometry consistency loss and task-oriented flow distillation loss, IFRNet can estimate relatively good intermediate flow with clear motion boundary. Further, we can see that merge mask M can identify occluded regions of warped frames by adjusting the mixing weight, where it tends to average the candidate regions when both views are visible. Finally, residual R can compensate for some contextual details, which usually response at motion boundary and image edges. Different from other flow-based VFI methods that take cascaded structure design, merge mask M and residual R in IFRNet share the same encoder-decoder with intermediate optical flow, making proposed architecture achieve better VFI accuracy while being more lightweight and fast.

Readers may think our IFRNet is similar with PWC-Net [14] which is designed for optical flow. However, It is non-trivial to adapt PWC-Net for frame interpolation, since previous related works employ it as one of many components. We summarize their difference in several aspects: **1)** Anchor feature in PWC-Net is extracted by the encoder, while in IFRNet, it is reconstructed by the decoder. **2)** Besides motion information in intermediate feature, there are occlusion, texture and temporal information in it. **3)** PWC-Net designed for motion estimation, is optimized only by flow regression loss with strong augmentation. However, IFRNet designed for frame synthesizing, is optimized in a multi-target manner with weak data augmentation.



Average interpolation error	avg. rank	Mequon (Hidden texture) im0 GT im1			Schefflera (Hidden texture) im0 GT im1			Urban (Synthetic) im0 GT im1			Teddy (Stereo) im0 GT im1			Backyard (High-speed camera) im0 GT im1			Basketball (High-speed camera) im0 GT im1			Dumptruck (High-speed camera) im0 GT im1			Evergreen (High-speed camera) im0 GT im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
SoftsplatAug [190]	2.6	1.98 1	2.91 1	1.06 3	2.55 2	3.38 2	1.14 2	1.87 3	2.69 2	1.06 2	3.88 3	4.65 3	2.70 3	7.24 1	8.90 1	2.98 6	3.90 3	7.06 3	1.97 3	5.24 3	11.4 3	1.38 5	5.22 2	8.02 2	1.50 4
SoftSplat [169]	5.3	2.06 2	3.06 3	1.14 9	2.80 5	3.91 6	1.24 3	1.99 5	2.73 3	1.21 6	3.84 2	4.64 2	2.69 2	8.10 18	10.0 18	2.96 2	4.10 5	7.53 5	1.98 6	5.49 5	12.1 5	1.39 6	5.40 3	8.33 3	1.50 4
IFRNet [193]	8.0	2.08 3	3.03 2	1.16 12	2.78 4	3.73 4	1.38 47	1.74 1	2.58 1	1.04 1	3.96 4	4.78 4	2.96 10	7.55 5	9.28 5	3.12 22	4.42 9	8.20 9	2.02 11	5.56 7	12.3 6	1.37 2	5.64 8	8.70 8	1.51 6
EAFI [186]	8.2	2.10 5	3.19 4	1.08 5	2.54 1	3.23 1	1.13 1	1.77 2	2.79 5	1.08 3	3.82 1	4.51 1	2.64 1	9.04 26	11.3 25	3.01 9	4.82 23	9.09 23	1.97 3	5.89 14	13.1 15	1.37 2	5.77 10	8.91 10	1.51 6
DistillNet [184]	10.0	2.11 6	3.29 5	1.15 11	2.71 3	3.64 3	1.28 16	1.96 4	2.73 3	1.14 4	4.05 5	4.96 6	2.81 5	7.81 9	9.66 9	3.06 14	4.79 21	9.03 20	2.01 9	6.04 16	13.4 18	1.43 14	6.05 11	9.33 11	1.56 16
SepConv++ [185]	13.0	2.39 23	4.17 25	1.20 24	2.98 8	4.21 9	1.28 16	3.34 24	3.23 8	2.20 88	4.49 12	5.81 17	2.87 7	7.64 7	9.42 7	2.97 3	3.77 2	6.80 2	1.96 1	5.26 4	11.6 4	1.36 1	5.71 9	8.86 9	1.45 1
FGME [158]	13.2	2.08 3	3.34 7	0.98 1	3.32 22	4.43 13	1.63 112	2.46 6	3.28 9	1.41 17	4.08 6	4.85 5	3.05 18	7.36 3	9.08 3	3.03 11	4.17 7	7.62 7	2.06 22	4.95 2	10.7 2	1.44 15	5.45 4	8.41 5	1.57 17
BMBC [171]	15.0	2.30 15	3.40 9	1.20 24	3.07 9	4.25 10	1.41 59	3.17 20	4.19 31	1.66 39	4.24 8	5.28 8	2.85 6	7.79 8	9.62 8	3.14 24	4.08 4	7.47 4	2.02 11	5.63 8	12.4 8	1.40 8	5.55 6	8.58 6	1.61 26
IDIAL [192]	15.9	2.23 8	3.62 12	1.14 9	3.22 13	4.54 21	1.46 76	2.79 9	2.97 6	1.23 7	4.49 12	5.64 13	2.94 9	8.36 20	10.4 20	2.97 3	4.53 12	8.43 12	1.99 7	6.17 18	13.3 17	1.50 24	6.31 17	9.67 15	1.58 21
STAR-Net [164]	17.1	2.18 7	3.37 8	1.21 42	3.46 31	4.88 31	1.47 79	3.04 18	3.53 15	1.58 31	4.41 11	5.44 11	2.76 4	7.51 4	9.27 4	2.98 6	4.65 13	8.72 13	1.99 7	6.21 20	13.4 18	1.41 9	6.17 13	9.45 13	1.49 3
EDSC [173]	18.8	2.32 19	3.90 17	1.16 12	3.10 10	4.38 12	1.51 88	2.98 15	3.54 16	1.36 15	4.49 12	5.74 14	3.16 31	8.05 17	9.96 17	3.08 16	4.89 24	9.28 24	2.02 11	5.55 6	12.3 6	1.41 9	6.42 22	9.99 23	1.55 15
AdaCoF [165]	22.8	2.41 25	4.10 24	1.26 135	3.10 10	4.32 11	1.43 65	3.48 29	3.31 10	1.78 56	4.84 23	5.94 24	2.93 8	8.68 23	10.8 22	3.14 24	4.13 6	7.59 6	1.97 3	5.77 12	12.9 13	1.37 2	5.60 7	8.67 7	1.48 2
DSepConv [162]	27.5	2.47 26	4.39 31	1.21 42	3.32 22	4.60 23	1.72 133	3.28 21	3.66 17	1.50 24	5.11 30	6.36 28	3.23 66	7.85 10	9.69 10	3.11 20	4.68 15	8.78 15	2.04 19	5.65 9	12.5 9	1.44 15	6.54 27	10.2 27	1.58 21
GDCN [172]	29.6	2.31 17	3.98 21	1.10 7	3.80 87	5.17 48	1.54 93	2.92 13	3.78 22	1.43 19	5.59 82	6.01 26	3.24 70	9.02 25	11.3 25	3.10 18	4.66 14	8.75 14	2.08 23	5.75 11	12.7 10	1.42 12	6.40 21	9.98 22	1.53 10
STSR [170]	29.9	2.31 17	3.82 15	1.19 17	2.94 6	3.90 5	1.93 169	2.92 13	3.44 14	1.81 57	4.29 10	5.41 9	3.27 79	9.51 29	11.9 29	3.06 14	5.38 34	10.3 34	2.10 24	6.75 29	15.3 31	1.50 24	6.43 24	9.99 23	1.54 11
ProBoost-Net [191]	32.1	2.27 12	3.90 17	1.07 4	3.70 71	5.05 40	1.78 144	2.98 15	3.38 12	1.65 38	4.53 16	5.76 15	3.33 106	8.75 24	10.9 24	3.25 29	5.01 25	9.45 25	2.14 26	6.02 15	13.5 20	1.45 17	6.50 26	10.1 26	1.59 23
MAF-net [163]	32.2	2.23 8	3.84 16	1.08 5	3.53 42	4.85 30	1.78 144	2.83 11	3.70 18	1.58 31	4.83 22	5.88 18	3.31 99	9.44 28	11.8 28	3.27 30	5.27 29	10.0 29	2.15 27	6.30 21	14.2 22	1.54 46	6.38 20	9.90 21	1.63 28
CtxSyn [134]	32.7	2.24 10	3.72 13	1.04 2	2.96 7	4.16 8	1.35 42	4.32 104	3.42 13	3.18 149	4.21 7	5.46 12	3.00 12	9.59 32	11.9 29	3.46 35	5.22 26	9.76 26	2.22 30	7.02 34	15.4 32	1.58 67	6.66 30	10.2 27	1.69 37
FRUCnet [153]	32.9	2.61 33	4.34 28	1.52 186	3.30 19	4.52 18	1.72 133	3.14 19	3.70 18	1.76 53	4.74 20	5.99 25	3.29 84	8.11 19	10.0 18	2.97 3	4.48 10	8.35 11	2.02 11	5.78 13	12.7 10	1.45 17	6.06 12	9.38 12	1.57 17
ADC [161]	32.9	2.54 31	4.31 26	1.29 154	3.27 16	4.46 14	1.62 110	3.76 55	3.76 20	1.70 47	5.27 37	6.37 29	3.19 46	8.66 22	10.8 22	3.11 20	4.78 19	9.04 21	2.01 9	5.72 10	12.8 12	1.41 9	6.56 28	10.2 27	1.51 6
CyclicGen [149]	33.2	2.26 11	3.32 6	1.42 181	3.19 12	4.01 7	2.21 184	2.76 8	4.05 29	1.62 35	4.97 25	5.92 21	3.79 169	8.00 16	9.84 16	3.13 23	3.36 1	5.65 1	2.17 28	4.55 1	9.68 1	1.42 12	4.48 1	6.84 1	1.52 9
FeFlow [167]	34.1	2.28 13	3.73 14	1.18 16	3.50 39	4.78 29	2.09 180	2.82 10	3.13 7	1.66 39	4.75 21	5.78 16	3.72 162	7.62 6	9.40 6	3.04 12	4.74 18	8.88 17	2.03 16	6.07 17	13.1 15	1.59 71	6.78 33	10.5 33	1.65 29
MPRN [151]	35.2	2.53 29	4.43 32	1.21 42	3.78 84	4.97 34	1.57 99	3.39 26	5.49 38	1.28 8	5.03 26	6.58 32	3.19 46	9.53 30	11.9 29	3.31 32	5.25 28	9.92 27	2.22 30	6.87 31	15.5 33	1.49 21	6.72 31	10.4 31	1.60 25
TC-GAN [166]	35.2	2.34 20	3.96 20	1.25 119	3.26 15	4.51 17	1.81 149	3.49 30	3.80 24	2.20 88	4.65 17	5.90 20	3.44 128	7.87 11	9.73 12	3.00 8	4.78 19	9.00 19	2.03 16	6.34 23	14.2 22	1.50 24	6.28 16	9.73 18	1.54 11
MV_VFI [183]	35.7	2.35 21	3.98 21	1.25 119	3.25 14	4.49 15	1.81 149	3.46 28	3.81 25	2.21 92	4.66 19	5.92 21	3.44 128	7.87 11	9.72 11	3.01 9	4.80 22	9.05 22	2.04 19	6.33 22	14.2 22	1.50 24	6.27 15	9.70 17	1.54 11
DAIN [152]	35.8	2.38 22	4.05 23	1.26 135	3.28 17	4.53 20	1.79 147	3.32 23	3.77 21	2.05 78	4.65 17	5.88 18	3.41 124	7.88 13	9.74 13	3.04 12	4.73 17	8.90 18	2.04 19	6.36 24	14.3 26	1.51 32	6.25 14	9.68 16	1.54 11
MS-PFT [159]	36.3	2.53 29	4.35 29	1.16 12	3.61 57	5.03 36	1.69 126	3.30 22	4.25 33	1.77 55	5.13 31	6.55 31	3.19 46	7.94 15	9.81 15	3.21 27	4.49 11	8.24 10	2.22 30	6.55 26	13.9 21	1.79 131	6.42 22	9.89 20	1.69 37
DAI [168]	39.2	2.30 15	3.42 10	1.47 185	3.46 31	4.66 25	1.92 163	2.55 7	3.78 22	1.33 10	4.27 9	5.10 7	4.24 182	9.07 27	11.3 25	3.08 16	5.28 30	10.1 31	2.02 11	6.56 27	14.7 28	1.39 6	6.48 25	10.0 25	1.59 23
MEMC-Net+ [160]	43.3	2.39 23	3.92 19	1.28 145	3.36 25	4.52 18	2.07 179	3.37 25	3.86 26	2.20 88	4.84 23	5.93 23	3.72 162	8.55 21	10.6 21	3.14 24	4.70 16	8.81 16	2.03 16	6.40 25	14.2 22	1.58 67	6.37 19	9.87 19	1.57 17
MDP-Flow2 [68]	44.1	2.89 37	5.38 39	1.19 17	3.47 33	5.07 43	1.26 5	3.66 44	6.10 72	2.48 115	5.20 33	7.48 43	3.14 28	10.2 36	12.8 37	3.61 60	6.13 59	11.8 54	2.31 62	7.36 39	16.8 37	1.49 21	7.75 54	12.1 53	1.69 37
PMMST [112]	44.5	2.90 39	5.43 41	1.20 24	3.50 39	5.05 40	1.27 10	3.56 34	5.46 36	1.82 60	5.38 50	7.92 70	3.41 124	10.2 36	12.8 37	3.60 53	5.76 36	11.0 36	2.26 38	7.39 41	16.9 40	1.53 39	7.57 39	11.8 39	1.72 68
SuperSlomo [130]	45.7	2.51 27	4.32 27	1.25 119	3.66 65	5.06 42	1.93 169	2.91 12	4.00 28	1.41 17	5.05 27	6.27 27	3.66 157	9.56 31	11.9 29	3.30 31	5.37 33	10.2 33	2.24 33	6.69 28	15.0 29	1.53 39	6.73 32	10.4 31	1.66 30

Figure 20. **Screenshot of our IE-ranking on the Middlebury benchmark (taken on the November 16th, 2021).**



Average normalized interpolation error	avg. rank	Mequon (Hidden texture) im0 GT im1			Schefflera (Hidden texture) im0 GT im1			Urban (Synthetic) im0 GT im1			Teddy (Stereo) im0 GT im1			Backyard (High-speed camera) im0 GT im1			Basketball (High-speed camera) im0 GT im1			Dumptruck (High-speed camera) im0 GT im1			Evergreen (High-speed camera) im0 GT im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
EAFI [186]	3.4	0.50 5	0.46 1	0.58 8	0.46 1	0.53 1	0.51 1	0.42 1	0.50 1	0.53 1	0.79 1	0.71 1	1.04 1	0.85 7	0.83 13	0.95 1	0.81 1	0.74 4	0.89 1	0.58 7	0.95 16	0.59 1	0.57 4	0.73 3	0.59 1
SoftsplatAug [190]	4.1	0.49 4	0.47 2	0.56 6	0.47 2	0.58 2	0.53 2	0.51 4	0.61 7	0.57 3	0.81 3	0.73 2	1.07 4	0.82 1	0.78 2	0.97 2	0.81 1	0.73 2	0.90 4	0.59 11	0.95 16	0.62 14	0.55 2	0.70 2	0.59 1
SoftSplat [169]	4.2	0.53 10	0.51 4	0.61 14	0.52 5	0.68 6	0.55 3	0.52 5	0.53 2	0.58 6	0.80 2	0.73 2	1.06 2	0.85 7	0.81 6	0.98 6	0.81 1	0.73 2	0.90 4	0.56 2	0.86 2	0.60 2	0.57 4	0.73 3	0.59 1
DistillNet [184]	7.1	0.52 8	0.52 6	0.60 12	0.50 3	0.62 3	0.55 3	0.52 5	0.58 5	0.57 3	0.81 3	0.75 4	1.07 4	0.84 5	0.81 6	0.97 2	0.85 11	0.90 19	0.91 9	0.57 4	0.88 3	0.61 5	0.65 19	0.88 21	0.60 7
IFRNet [193]	7.5	0.53 10	0.51 4	0.62 17	0.51 4	0.63 4	0.57 5	0.43 2	0.54 3	0.54 2	0.83 5	0.76 5	1.13 14	0.87 14	0.83 13	1.03 19	0.84 8	0.78 6	0.92 14	0.56 2	0.88 3	0.61 5	0.58 8	0.75 7	0.60 7
IDIAL [192]	11.9	0.52 8	0.56 8	0.58 8	0.59 11	0.78 17	0.58 6	0.61 8	0.69 10	0.65 11	0.85 7	0.81 9	1.08 6	0.88 18	0.88 25	1.00 11	0.83 5	0.82 8	0.90 4	0.62 23	1.01 27	0.62 14	0.63 11	0.84 13	0.61 18
BMBC [171]	13.0	0.57 19	0.57 10	0.64 23	0.58 9	0.73 8	0.64 92	0.77 22	0.78 22	0.71 21	0.84 6	0.77 7	1.09 7	0.85 7	0.81 6	0.98 6	0.82 4	0.77 5	0.91 9	0.58 7	0.93 12	0.60 2	0.56 3	0.73 3	0.59 1
SepConv++ [185]	16.4	0.58 22	0.67 25	0.64 23	0.56 7	0.73 8	0.59 11	0.95 49	0.70 14	1.30 96	0.87 10	0.87 17	1.11 9	0.85 7	0.81 6	1.00 11	0.84 8	0.88 14	0.89 1	0.59 11	0.97 21	0.61 5	0.59 9	0.79 9	0.59 1
EDSC [173]	17.5	0.53 10	0.60 16	0.59 11	0.58 9	0.76 11	0.60 35	0.63 9	0.76 20	0.69 16	0.88 18	0.90 23	1.13 14	0.88 18	0.85 19	1.02 17	0.91 23	1.09 33	0.92 14	0.59 11	0.95 16	0.64 19	0.64 16	0.85 16	0.63 25
MV_VFI [183]	18.2	0.57 19	0.62 18	0.64 23	0.60 16	0.78 17	0.62 60	0.79 26	0.83 27	0.76 29	0.87 10	0.87 17	1.11 9	0.87 14	0.82 11	1.01 13	0.86 14	0.89 17	0.92 14	0.59 11	0.96 20	0.61 5	0.65 19	0.88 21	0.60 7
TC-GAN [166]	18.5	0.57 19	0.62 18	0.64 23	0.60 16	0.78 17	0.63 80	0.78 24	0.81 25	0.75 25	0.87 10	0.87 17	1.11 9	0.86 11	0.82 11	1.01 13	0.86 14	0.88 14	0.92 14	0.59 11	0.95 16	0.61 5	0.65 19	0.89 25	0.60 7
STAR-Net [164]	18.5	0.56 17	0.56 8	0.65 75	0.65 53	0.85 38	0.62 60	0.70 15	0.69 10	0.76 29	0.87 10	0.82 11	1.06 2	0.83 3	0.79 3	0.97 2	0.83 5	0.83 9	0.90 4	0.63 26	1.09 32	0.61 5	0.61 10	0.80 10	0.60 7
DAIN [152]	19.8	0.58 22	0.63 20	0.65 75	0.60 16	0.79 23	0.62 60	0.69 14	0.73 18	0.68 15	0.86 9	0.86 16	1.10 8	0.87 14	0.83 13	1.02 17	0.85 11	0.86 13	0.92 14	0.59 11	0.97 21	0.61 5	0.66 25	0.90 27	0.60 7
FGME [158]	20.2	0.46 1	0.49 3	0.51 1	0.63 31	0.78 17	0.64 92	0.60 7	0.65 9	0.67 14	0.85 7	0.76 5	1.15 17	0.82 1	0.77 1	0.99 9	0.86 14	0.81 7	0.95 23	0.61 22	0.93 12	0.70 53	0.63 11	0.83 11	0.65 118
STSR [170]	22.4	0.54 13	0.58 13	0.61 14	0.56 7	0.66 5	0.68 127	0.65 11	0.72 17	0.74 24	0.87 10	0.83 14	1.15 17	0.91 25	0.89 27	1.04 21	0.93 27	1.08 31	0.95 23	0.62 23	1.04 28	0.63 18	0.63 11	0.84 13	0.61 18
AdaCoF [165]	22.7	0.60 39	0.68 27	0.67 138	0.59 11	0.76 11	0.60 35	0.84 30	0.71 16	0.94 59	0.92 23	0.89 21	1.11 9	0.90 23	0.85 19	1.06 25	0.84 8	0.85 11	0.90 4	0.58 7	0.93 12	0.61 5	0.57 4	0.75 7	0.59 1
DSepConv [162]	27.2	0.58 22	0.72 34	0.62 17	0.63 31	0.82 29	0.65 106	0.72 18	0.70 14	0.75 25	0.96 49	0.97 32	1.15 17	0.87 14	0.83 13	1.04 21	0.89 21	1.00 25	0.93 20	0.57 4	0.89 5	0.64 19	0.65 19	0.87 20	0.64 77
MEMC-Net+ [160]	28.3	0.59 26	0.65 22	0.65 75	0.64 40	0.79 23	0.70 139	0.80 27	0.77 21	0.96 61	0.88 18	0.83 14	1.12 13	0.88 18	0.85 19	1.01 13	0.85 11	0.88 14	0.91 9	0.64 29	1.13 37	0.62 14	0.63 11	0.86 18	0.60 7
ProBoost-Net [191]	28.8	0.48 2	0.58 13	0.52 2	0.69 82	0.90 62	0.64 92	0.67 12	0.69 10	0.70 18	0.90 20	0.87 17	1.19 27	0.89 22	0.84 18	1.08 26	0.92 24	0.95 21	0.99 26	0.59 11	0.90 7	0.66 29	0.64 16	0.85 16	0.65 118
FeFlow [167]	28.9	0.51 7	0.58 13	0.56 6	0.66 57	0.84 33	0.67 123	0.70 15	0.69 10	0.86 45	0.87 10	0.82 11	1.13 14	0.84 5	0.80 4	0.99 9	0.86 14	0.84 10	0.93 20	0.64 29	0.98 23	0.71 64	0.67 26	0.90 27	0.65 118
MPRN [151]	31.5	0.59 26	0.70 30	0.64 23	0.66 57	0.89 55	0.64 92	0.77 22	1.07 33	0.64 8	0.93 26	0.93 28	1.17 23	0.95 31	0.91 31	1.12 30	0.96 31	1.04 29	1.01 29	0.60 19	0.98 23	0.65 25	0.72 32	1.02 33	0.62 21
ADC [161]	31.7	0.61 52	0.68 27	0.67 138	0.62 24	0.78 17	0.66 114	0.84 30	0.81 25	0.82 38	0.96 49	0.93 28	1.15 17	0.90 23	0.86 24	1.05 24	0.92 24	1.12 34	0.91 9	0.57 4	0.92 10	0.61 5	0.64 16	0.88 21	0.60 7
GDCN [172]	33.5	0.54 13	0.65 22	0.58 8	0.72 103	0.94 79	0.64 92	0.63 9	0.79 23	0.69 16	1.03 122	0.90 23	1.18 24	0.93 28	0.93 32	1.04 21	0.90 22	1.01 27	0.94 22	0.59 11	0.94 15	0.64 19	0.67 26	0.93 29	0.61 18
DAI [168]	37.8	0.65 134	0.52 6	0.79 185	0.64 40	0.82 29	0.66 114	0.47 3	0.56 4	0.57 3	0.91 21	0.77 7	1.41 175	0.88 18	0.85 19	0.98 6	0.87 19	0.96 22	0.91 9	0.60 19	0.99 25	0.60 2	0.65 19	0.88 21	0.60 7
MAF-net [163]	39.6	0.48 2	0.61 17	0.52 2	0.65 53	0.86 43	0.62 60	0.67 12	0.86 29	0.78 34	0.96 49	0.92 27	1.20 32	0.93 28	0.89 27	1.08 26	0.95 29	1.08 31	0.99 26	0.67 37	1.04 28	0.83 140	0.65 19	0.86 18	0.69 182
CyclicGen [149]	39.9	0.64 126	0.63 20	0.73 180	0.67 67	0.73 8	0.88 186	0.72 18	0.84 28	0.78 34	0.95 40	0.89 21	1.24 84	0.91 25	0.85 19	1.09 28	0.87 19	0.67 1	1.00 28	0.53 1	0.71 1	0.62 14	0.52 1	0.64 1	0.60 7
FRUCnet [153]	40.0	0.70 164	0.71 32	0.80 186	0.64 40	0.80 26	0.69 134	0.78 24	0.75 19	0.95 60	0.91 21	0.91 26	1.15 17	0.86 11	0.83 13	1.01 13	0.86 14	0.89 17	0.92 14	0.58 7	0.89 5	0.64 19	0.63 11	0.83 11	0.64 77
CtxSyn [134]	41.0	0.50 5	0.57 10	0.55 5	0.55 6	0.71 7	0.59 11	1.42 134	0.64 8	2.08 151	0.87 10	0.82 11	1.18 24	0.95 31	0.90 29	1.13 32	0.94 28	0.92 20	1.02 30	0.68 38	1.00 26	0.83 140	0.67 26	0.89 25	0.68 177
PMMST [112]	47.8	0.59 26	0.73 37	0.64 23	0.64 40	0.85 38	0.59 11	0.99 54	1.69 80	1.05 70	0.97 63	1.14 109	1.23 69	0.99 38	0.96 37	1.14 34	1.03 41	1.29 42	1.04 39	0.71 51	1.33 57	0.67 36	0.77 37	1.10 38	0.64 77
SuperSlomo [130]	48.5	0.59 26	0.69 29	0.64 23	0.72 103	0.91 67	0.75 160	0.74 20	1.01 32	0.71 21	0.98 76	0.95 30	1.23 69	0.94 30	0.90 29	1.12 30	0.96 31	0.98 23	1.04 39	0.60 19	0.90 7	0.71 64	0.69 30	0.93 29	0.68 177
FLAVR [188]	49.5	0.67 150	0.70 30	0.71 171	0.71 95	0.78 17	0.76 163	0.76 21	0.80 24	0.75 25	1.24 175	1.28 155	1.23 69	0.83 3	0.80 4	0.97 2	0.83 5	0.85 11	0.89 1	0.62 23	0.92 10	0.64 19	0.57 4	0.73 3	0.60 7
OFRI [154]	49.5	0.60 39	0.57 10	0.69 162	0.67 67	0.81 28	0.79 171	0.70 15	0.59 6	0.83 42	0.87 10	0.81 9	1.15 17	0.86 11	0.81 6	1.03 19	0.92 24	0.99 24	0.98 25	0.79 91	0.90 7	1.18 177	0.67 26	0.84 13	0.78 190

Figure 21. **Screenshot of our NIE-ranking on the Middlebury benchmark (taken on the November 16th, 2021).**

10. Screenshots of the Middlebury Benchmark

We take screenshots of the online Middlebury benchmark for VFI on the November 16th, 2021, whose results are shown in Figure 20 and Figure 21. Since the average rank is a relative indicator, previous methods [1, 3, 10, 11] usually report average IE (interpolation error) and average NIE (normalized interpolation error) for comparison. As summarized in Table 2 in our main paper, proposed IFRNet large model achieves best results on both IE and NIE metrics among all published VFI methods that are trained on Vimeo90K [15] dataset. Moreover, IFRNet large runs several times faster than previous state-of-the-art algorithms [10, 12], demonstrating the superior VFI accuracy and fast inference speed of proposed approaches.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 6

[2] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 1, 2, 3

[3] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.

Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[5] Huaizu Jiang, Deqing Sun, Varan Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[6] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. In *Arxiv*, 2021. 1

[7] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[8] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[9] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1

[10] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[11] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *European Conference on Computer Vision*, 2020. 6

[12] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 6

[13] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[14] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 5

[15] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 2019. 4, 5, 6