Long-term Video Frame Interpolation via Feature Propagation

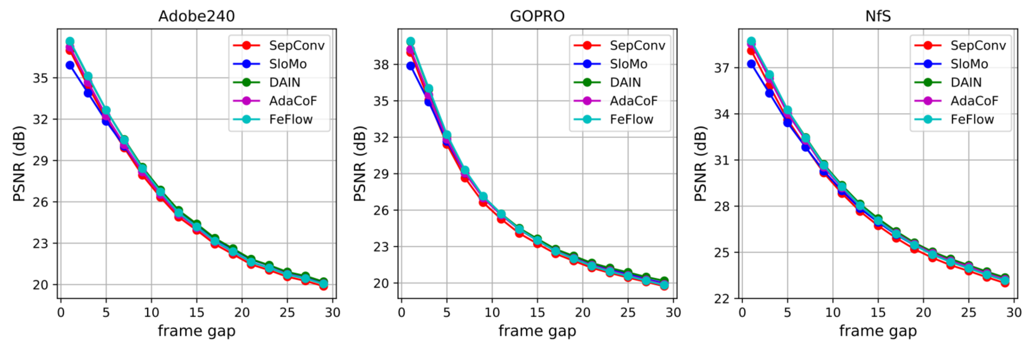
Dawit Mureja Argaw and In So Kweon

Introduction

- Video frame interpolation targets temporal super-resolution of an input video
  - Key premise: *the frame rate of the input sequence is already sufficiently high*

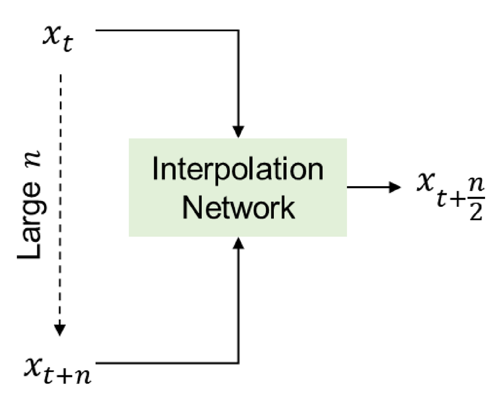
- How does VFI work?
  - Given input frames: $\langle x_t, x_{t+n} \rangle$
  1. Estimate motion between x_t and x_{t+n}
     - ✓ Optical flow
     - ✓ Motion kernel
  2. Interpolate the estimated motion to the target time
     - ✓ Directly warping input frames
     - ✓ Frame synthesis network

- What happens if we increase n?

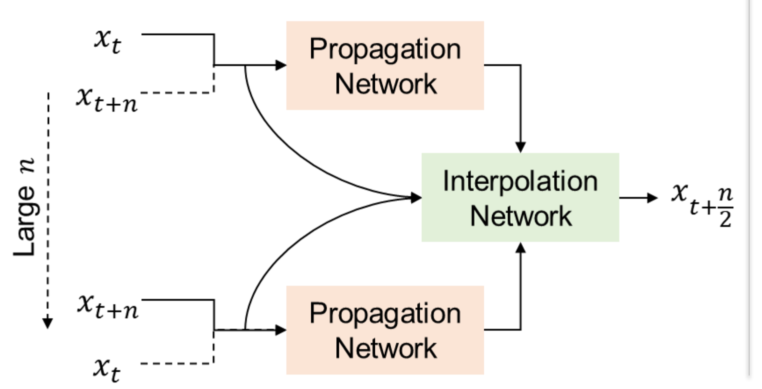


- *Incorrect motion estimation → incorrect interpolated frame*

- Previous works

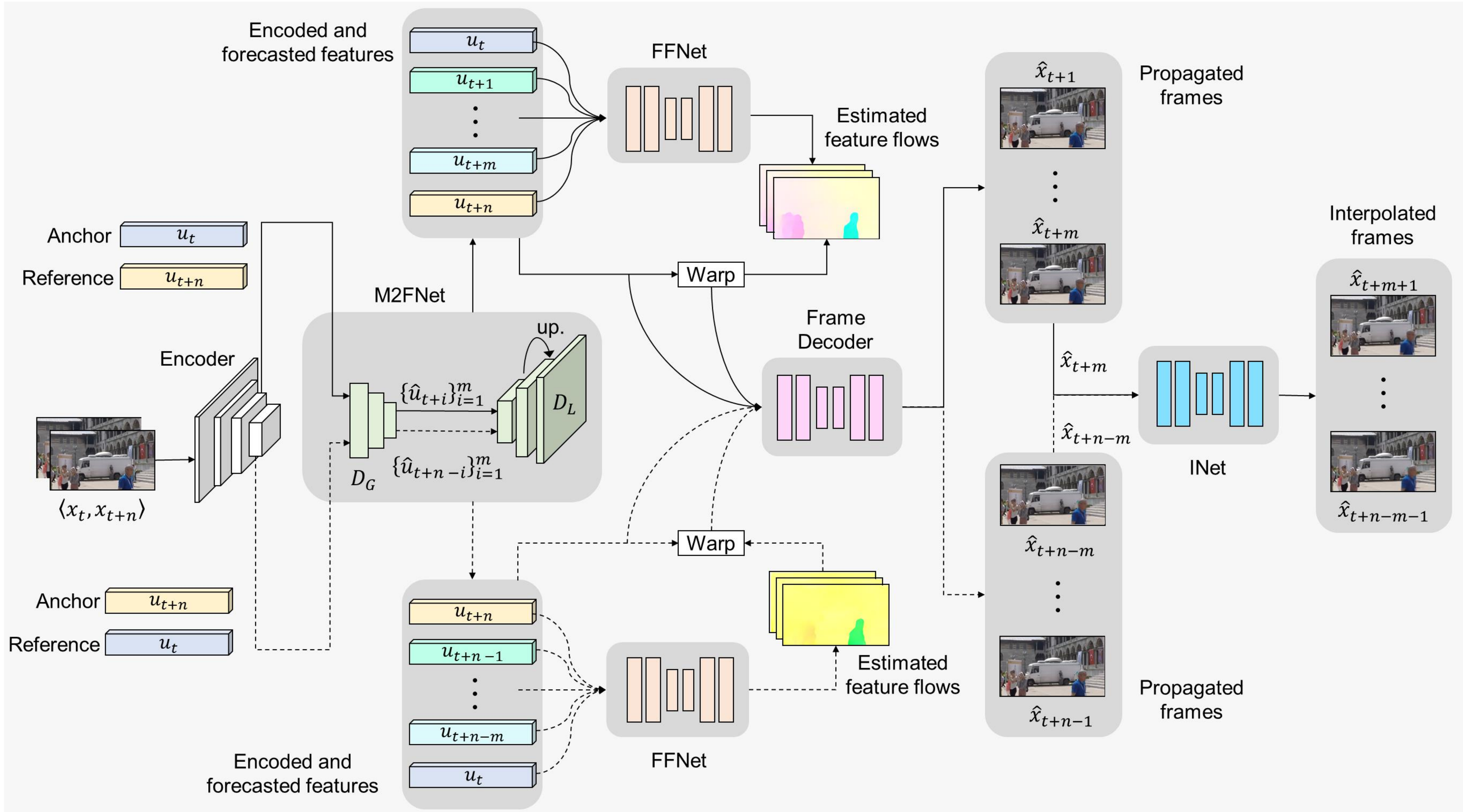


- Proposed formulation



- Contributions
  - General VFI framework relatively robust to low frame rate videos
  - Frame propagation network as a plug-in module
  - State-of-the-art performance on long term VFI

Propagation-Interpolation Network (P-INet)



Methodology

- Motion-to-feature forecasting
  - Encoder → top-down feature extraction
    - ✓ Anchor and reference features

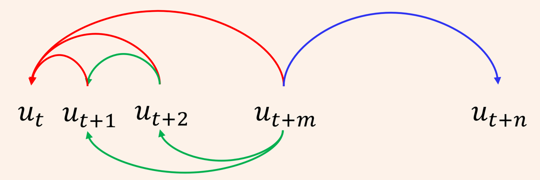
$$\{u_t^l\}_{l=1}^{k} = \text{Encoder}\,(x_t) \qquad \{u_{t+n}^l\}_{l=1}^{k} = \text{Encoder}\,(x_{t+n})$$

  - Motion decoders → to forecast features
    - ✓ Global (STN)
    - ✓ Local (Conv. decoder)

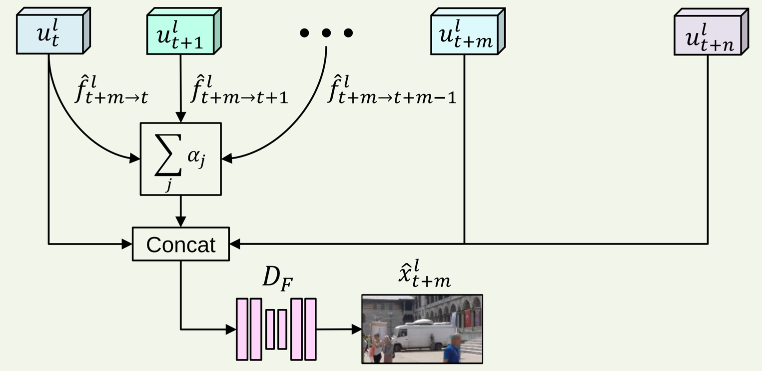
$$\left\{\theta^l_{\langle R_{t+i} | T_{t+i}\rangle}\right\}_{i=1}^{m} = D_G^l\left(u_t^l \,||\, u_{t+n}^l\right) \qquad u_{t+i}^l = D_L^l\left(\hat{u}_{t+i}^l \,||u_t^l\, ||u_{t+n}^l\, ||\text{up.}\left(u_{t+i}^{l+1}\right)\right)$$

$$\{\hat{u}_{t+i}^l\}_{i=1}^{m} = \text{trans.}\left(u_t^l, \left\{\theta^l_{\langle R_{t+i} | T_{t+i}\rangle}\right\}_{i=1}^{m}\right)$$

- Feature flow estimation
  - Optical flow estimator in PWC-Net



- Feature-to-frame decoding
  - Frame decoder → bottom-up frame synthesis



$$v_{t+i}^l = \sum_{j=t}^{t+i-1} \alpha_j.\text{warp}\left(u_j^l, \hat{f}^l_{t+i\to j}\right) \qquad \hat{x}_{t+i}^l = D_F^l\left(u_{t+i}^l \,||v_{t+i}^l\, ||u_t^l\, ||u_{t+n}^l\, ||\text{up.}\left(\hat{x}_{t+i}^{l+1}\right)\right)$$

- Training algorithm of P-INet

Input : $\langle x_t, x_{t+n} \rangle$ // n is the frame gap
Output : $\hat{x}_{t+i}$, where $1 < i < n$
Let N be the maximum frame gap in the dataset, M be the upper limit for small frame gap, and $\Delta t(n)$ be a reliable time frame of propagation which is dependent on n
foreach *input sample* **do**
  if $n \leq M$ **then** // small gap
    $\hat{x}_{t+i} = \text{INet}\,(x_t, x_{t+n})$ **for** all i
  else // large gap ($M < n \leq N$)
    if $i \leq \Delta t(n)$ **then** // propagate from x_t
      $\hat{x}_{t+i} = \text{PNet}\,(x_t, x_{t+n})$
    else if $\Delta t(n) < i < n - \Delta t(n)$ **then**
      // propagate and interpolate
      $\hat{x}_{t+\Delta t(n)} = \text{PNet}\,(x_t, x_{t+n})$
      $\hat{x}_{t+n-\Delta t(n)} = \text{PNet}\,(x_{t+n}, x_t)$
      $\hat{x}_{t+i} = \text{INet}\,(\hat{x}_{t+\Delta t(n)}, \hat{x}_{t+n-\Delta t(n)})$
    else // propagate from x_{t+n}
      $\hat{x}_{t+i} = \text{PNet}\,(x_{t+n}, x_t)$
  end
end

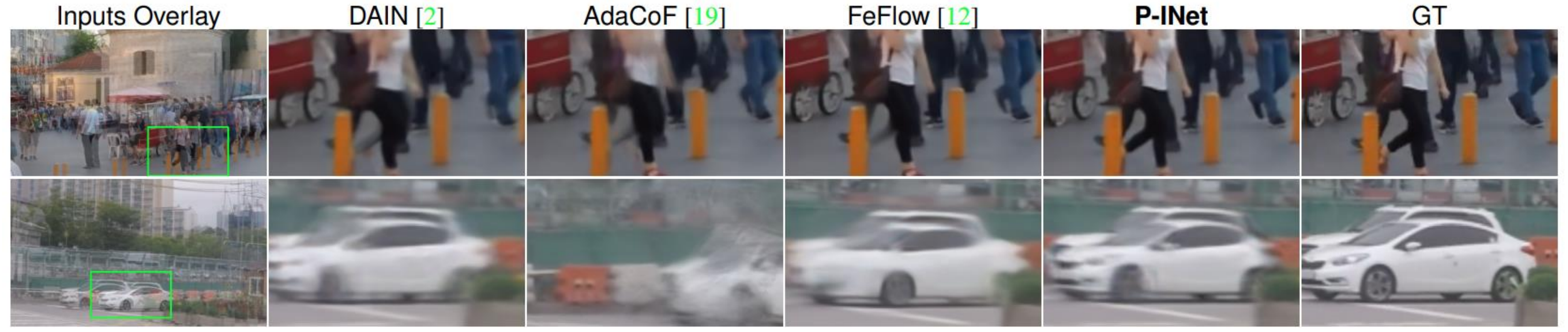
Experimental Results

- Quantitative comparison of our approach and SOTA methods at different fps

Method	Adobe240 [40]						GOPRO [27]						NfS [11]					
	30 fps		15 fps		8 fps		30 fps		15 fps		8 fps		30 fps		15 fps		8 fps	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SepConv [30]	29.91	0.915	23.94	0.811	19.88	0.707	28.64	0.871	23.23	0.694	19.74	0.560	31.84	0.915	26.73	0.811	23.00	0.707
SloMo [17]	30.03	0.917	24.30	0.818	20.17	0.717	29.03	0.917	23.58	0.818	19.99	0.718	31.83	0.917	26.95	0.818	23.19	0.717
DAIN [2]	30.53	0.924	24.39	0.824	20.21	0.721	29.25	0.924	23.63	0.824	20.18	0.721	32.46	0.924	27.19	0.824	23.36	0.720
AdaCoF [19]	30.14	0.896	24.11	0.741	20.07	0.567	29.05	0.876	23.49	0.701	19.89	0.571	32.28	0.919	27.05	0.819	23.23	0.719
FeFlow [12]	30.48	0.902	24.19	0.737	20.04	0.576	29.30	0.921	23.51	0.822	19.82	0.724	32.42	0.921	27.05	0.822	23.16	0.724
INet	30.30	0.920	24.21	0.819	20.12	0.718	29.17	0.919	23.59	0.821	20.04	0.722	32.03	0.920	26.99	0.822	23.27	0.721
P-INet	30.30	0.920	27.10	0.890	24.00	0.810	29.17	0.919	26.45	0.879	23.90	0.804	32.03	0.920	28.98	0.874	26.23	0.798

- ❑ Qualitative analysis on input samples with large temporal gap

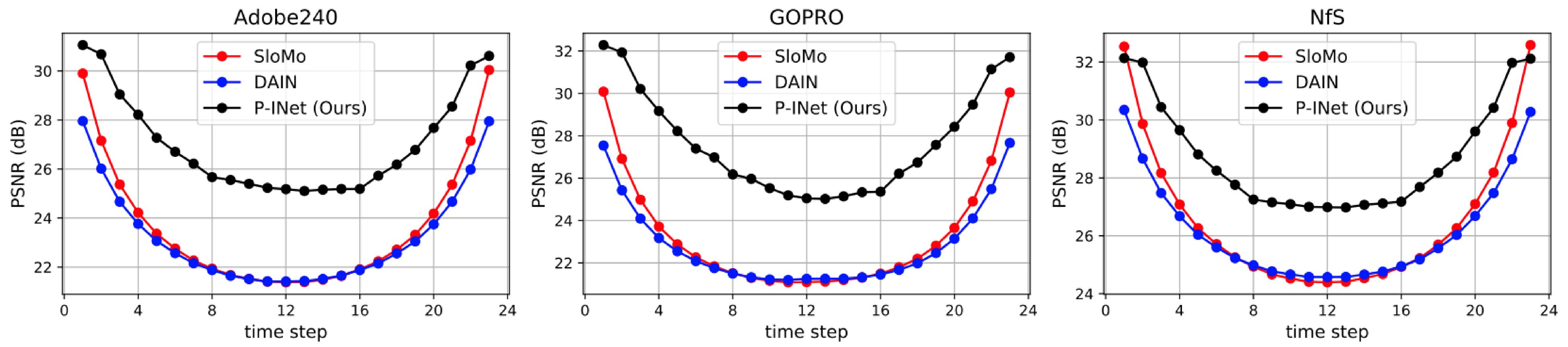
Inputs Overlay | DAIN [2] | AdaCoF [19] | FeFlow [12] | P-INet | GT



- Qualitative analysis of cascading PNet with state-of-the-art VFI approaches

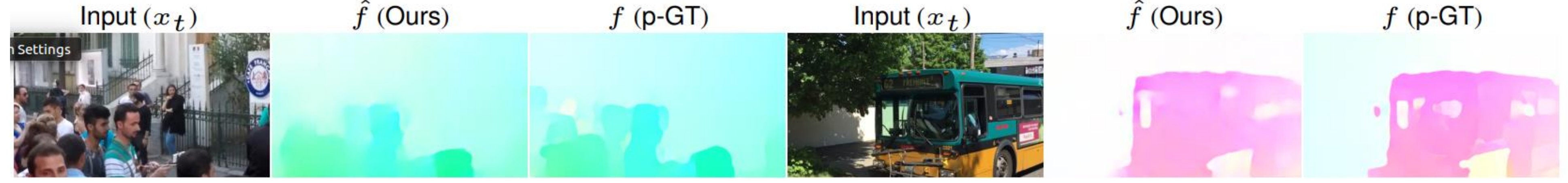
Inputs Overlay | FeFlow [12] | **PNet** + FeFlow [12] | DAIN [2] | **PNet** + DAIN [2] | GT

- Quantitative analysis of frames at different time steps (10 → 240 fps)



- Qualitative analysis of estimated feature flows and p-GT flows

Input (x_t) | $\hat{f}$ (Ours) | f (p-GT) | Input (x_t) | $\hat{f}$ (Ours) | f (p-GT)



Ablation Studies

- Loss functions
  - Intermediate flow supervision using p-GT flows is important (inter-frame motion and direction supervisions)
  - Gradient difference loss improves performance
- M2FNet
  - Conv. Nets are capable of decoding both global and local motions
  - Explicitly using global decoder improved performance
- Frame decoding
  - Incorporating features of past frames when decoding a current frame is important for long-term VFI

	Adobe240 [40]		GOPRO [27]	
Loss Functions	PSNR	SSIM	PSNR	SSIM
w/o $\mathcal{L}_{\text{M2FNet}}$	25.09	0.730	25.16	0.728
w/o inter-frame motion	25.81	0.776	26.11	0.776
w/o direction supervision	27.13	0.801	27.83	0.806
w/o $\mathcal{L}_{\text{GDL}}$	26.97	0.801	27.84	0.813
M2FNet				
w/o $\mathcal{D}_L$	25.13	0.734	25.71	0.760
w/o $\mathcal{D}_G$	26.96	0.801	27.34	0.811
Frame Decoding				
only warping u_t in Eq. (7)	26.82	0.793	27.41	0.812
excluding v_{t+i} from Eq. (8)	26.03	0.781	26.57	0.789
P-INet	**27.70**	**0.816**	**28.43**	**0.843**

References

[1]. Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation, CVPR'19 (**DAIN**)

[2]. Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation, CVPR'20 (**FeFlow**)

[3]. Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik G. Learned-Miller, and Jan Kautz. Super slomo: High-quality estimation of multiple intermediate frames for video interpolation, CVPR'18 (**SloMo**)

[4]. Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation, CVPR'20 (**AdaCoF**)

[5]. Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution, CVPR'17 (**SepConv**)

CVPR 2022 JUNE 19-24 NEW ORLEANS • LOUISIANA