Supplementary Material: Range-nullspace Video Frame Interpolation with Focalized Motion Estimation

Zhiyang Yu, Yu Zhang, Dongqing Zou, Xijun Chen, Jimmy S. Ren, Shunqing Ren

A. Introduction

In this supplementary material, we elaborate the details on the following aspects:

1. **Additional derivations and analyses (Sect. B)**, providing additional remarks for Eq.(14),(16), and (19) in the submitted paper.
2. **Further analyses (Sect. C)**, including the connection of the proposed FME with previous methods and the influence of optical flow extractor.
3. **Benchmarking reproducibility (Sect. D)**, where we provide evidence for guaranteeing the reproducibility of all the methods in our benchmarks.
4. **Architecture details (Sect. E)**, providing detailed architecture of the proposed framework.
5. **More visual results (Sect. F)**, providing more visual comparisons (figures and videos) between our approach and the state-of-the-art methods on test datasets. Videos can be found at https://youtu.be/Xyn1a2wRQJ8

The equations, theorems, tables, and figures are all numbered consecutively to those in the submitted paper.

B. Additional derivations and analyses

B.1. The forward operator has no right inverse

For any forward operator $\hat{\mathbf{H}}_t = [\mathbf{H}_{0\to t}^{\mathrm{T}}, \mathbf{H}_{1\to t}^{\mathrm{T}}]^{\mathrm{T}}$, a necessary condition for the existence of its right inverse is that $\hat{\mathbf{H}}_t$ must have full row rank. However, due to the particular physical meaning defined in Eq.(11), if the dimension of a vectorized frame is N, $\hat{\mathbf{H}}_t$ should satisfy $\hat{\mathbf{H}}_t \in \mathbb{R}^{2N\times N}$, then $rank(\hat{\mathbf{H}}_t) \leq min\{2N, N\} < 2N$. So $\hat{\mathbf{H}}_t$ does not have full row rank or right inverse.

B.2. Additional remarks of Eq.(16)

The pseudo inverse of $\mathbf{C} = [\mathbf{I}_d, -\mathbf{I}_d]$ is easy to be achieved:$\mathbf{C}^{\pm} = \frac{1}{2}[\mathbf{I}_d^{\mathrm{T}}, \ -\mathbf{I}_d^{\mathrm{T}}]^{\mathrm{T}}$. Then, the nullspace projection of $\mathcal{C}$ can be formulated as:

$$\mathcal{N}_{\mathcal{C}}(\boldsymbol{y}) = (\mathbf{I}_d - \mathbf{C}^{\pm}\mathbf{C})\boldsymbol{y} = \frac{1}{2}\begin{bmatrix}\mathbf{I}_d & \mathbf{I}_d\\ \mathbf{I}_d & \mathbf{I}_d\end{bmatrix}\boldsymbol{y}. \tag{20}$$

Donate $\boldsymbol{y} = [\boldsymbol{y}_1^{\mathrm{T}}, \ \boldsymbol{y}_2^{\mathrm{T}}]^{\mathrm{T}}$, if we suppose the distance metric $\Sigma = \mathbf{I}_d$ and $\tilde{\mathbf{I}}_t^* = [\tilde{\mathbf{I}}_{t,1}^{*\mathrm{T}}, \ \tilde{\mathbf{I}}_{t,2}^{*\mathrm{T}}]^{\mathrm{T}}$, Eq.(16) can be reformulated as:

$$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}\in\mathbb{R}^{2N}} (\frac{\boldsymbol{y}_1 + \boldsymbol{y}_2}{2} - \tilde{\mathbf{I}}_{t,1}^*)^2 + (\frac{\boldsymbol{y}_1 + \boldsymbol{y}_2}{2} - \tilde{\mathbf{I}}_{t,2}^*)^2 \tag{21}$$

whose minimum can be achieved when $(\boldsymbol{y}_1 + \boldsymbol{y}_2)/2 = (\tilde{\mathbf{I}}_{t,1}^* + \tilde{\mathbf{I}}_{t,2}^*)/2$. So the projected result is

$$\tilde{\boldsymbol{I}}_t^{*\mathcal{N}_{\mathcal{C}}} = \mathcal{N}_{\mathcal{C}}(\boldsymbol{y}^*) = \begin{bmatrix}(\tilde{\mathbf{I}}_{t,1}^* + \tilde{\mathbf{I}}_{t,2}^*)/2\\ (\tilde{\mathbf{I}}_{t,1}^* + \tilde{\mathbf{I}}_{t,2}^*)/2,\end{bmatrix} \tag{22}$$

and the corresponding final solution is $\mathbf{I}_t^* = (\tilde{\mathbf{I}}_{t,1}^* + \tilde{\mathbf{I}}_{t,2}^*)/2$. However, supposing $\Sigma = \mathbf{I}_d$ may not be the best choice, so we benefit from end-to-end training and predict a dynamic blending mask as a diagonal matrix, yielding Eq.(17).

B.3. Gradient of the modified ℓ_2 error

We use the variant of standard ℓ_2 error: $\log(\|\mathbf{I}_t^* - \mathbf{I}_t^{gt}\|_2^2 + \epsilon)$ whose gradient is $\frac{2\times(\mathbf{I}_t^* - \mathbf{I}_t^{gt})}{\|\mathbf{I}_t^* - \mathbf{I}_t^{gt}\|_2^2 + \epsilon}$. Compared with the gradient of the standard ℓ_2 error:$2 \times (\mathbf{I}_t^* - \mathbf{I}_t^{gt})$, the denominator can mitigate the diminishing gradient issue when the prediction is close to the ground truth.

C. Further analyses

C.1. Connection of FME with previous methods

To understand the connection with previous works, it is instructive to consider some special cases of Eq. (7) and Eq. (8).

1) Connection with Quadratic [11]: In Eq. (8), if $\mathbf{W}_{-1} \equiv \mathbf{W}_1 \equiv \mathbf{W}_2 \equiv 1/3$, then the proposed method degenerates into [11], where the confidence no longer works.

2) Connection with Quadratic [18]: In Eq. (8), if $\mathbf{W}_{-1} \equiv \mathbf{W}_1 \equiv 1/2$ and $\mathbf{W}_2 \equiv 0$, then the proposed method degenerates into [18], where the approximated trajectory will faithfully pass through measurements at t_{-1} and t_1.

3) Connection with Linear: In Eq. (7), if $\mathbf{W}_{-1} \equiv \mathbf{W}_2 \equiv 0$ and $\mathbf{W}_1 \equiv 1$, then we can get $\mathbf{F}_{0\to 1} \equiv \mathbf{A} + \mathbf{B}$,

Table 5. Comparing our methods with the control groups on the dataset of GoPro in terms of PSNR and SSIM.

	QVI	QVI-RAFT	EQVI	EQVI-RAFT	DBVI	Ours
PSNR	30.52	30.91	30.81	31.12	31.73	32.31
SSIM	0.941	0.942	0.942	0.943	0.947	0.951

Table 6. Comparing our reimplementations with reference results on the GoPro dataset in terms of PSNR/SSIM.

	EQVI	M2M	IFRNet	RIFE$_m$
Reference	28.10/0.915	29.76/0.919	29.84/0.920	29.68/0.924
Retrained	30.81/0.942	30.52/0.933	30.00/0.928	29.79/0.925

Table 7. Comparing our reimplementations with reference results on the Vimeo-90K (setpulates) dataset in terms of PSNR/SSIM.

	EQVI	M2M	IFRNet	RIFE$_m$	GDConv	ST-MFNet
Reference	29.71/0.934	35.16/0.971	35.90/0.973	35.27/0.972	35.58/0.958	36.49/0.976
Retrained	35.16/0.973	35.56/0.973	36.37/0.976	35.87/0.974	35.58/0.972	36.46/0.976

Table 8. Quantitative comparison with FILM for 2× VFI.

	Vimeo-90K (septulets)	UCF101	DAVIS	SNU-FILM			
				Easy	Medium	Hard	Extreme
FILM-$\mathcal{L}_1$	35.83 / 0.972	32.85 / 0.969	27.59 / 0.881	40.20 / 0.991	36.02 / 0.980	30.49 / 0.936	24.87 / 0.854
Ours	36.33 / 0.975	33.25 / 0.970	28.84 / 0.905	40.67 / 0.991	37.36 / 0.985	32.21 / 0.955	26.22 / 0.877

where $\mathbf{A}$ and $\mathbf{B}$ can be arbitrary values, but their sum should be equal to $\boldsymbol{F}_{01}$. If $\mathbf{A}$ is set to 0, then $\mathbf{B} \equiv \mathbf{F}_{01}$, yielding $f(\tau, \mathbf{A}, \mathbf{B}) = \mathbf{F}_{0\to1}\tau$, which is in line with the linear approximation.

C.2. Influence of optical flow extractor

Following DBVI [19], we use RAFT [16] to compute the optical flows among the input frames, which is a more accurate flow extractor compared with those used in QVI [18] and EQVI [11]. To prove that the superiority of our proposed method does not merely benefit from a better flow extractor, we establish control groups by retraining QVI and EQVI with RAFT on the training set of GoPro under the same settings as our submitted paper. The comparisons are reported in Table. 5, where all the methods are based on the quadratic motion model. Among the 6 methods, QVI-RAFT, EQVI-RAFT, DBVI, and our method are powered with RAFT. By comparing QVI-RAFT with QVI, or EQVI-RAFT with EQVI, it can be concluded that a better optical flow extractor does improve the performance of VFI. However, just improving the flow extractor is inadequate to make them comparable with the leading methods like DBVI. Our method still achieves a significant advantage thanks to the novel Focalized Motion Estimation and Range-nullspace Synthesis.

D. Benchmarking Reproducibility

D.1. Experiments for 8× VFI

In the submitted paper, 11 methods have been compared in the 8× VIF benchmarks trained on GoPro's training set, among which, unified evaluation of SloMo [8], QVI [18], DAIN [1], EDSC [3], FALVR [9], XVFI [15], and DBVI [19] have been provided by [19]. To guarantee the reproducibility, we firstly reproduced all the existing methods reported in [19], under the same settings of [19] or [9], and then extended the benchmarks with EQVI [11], M2M [6], IFRNet [10], RIFE$_m$ [7]. To verify the reproducibility of these extra-retrained methods, we use the evaluation results of their pre-trained models as references. Comparisons are shown in Table 6, where the results of our retrained models are better than the reference.

After verifying the reproducibility of these methods, we then retrain them on the X4K100FPS dataset and Vimeo (setpules) for further evaluation.

D.2. Experiments for 2× VFI

In the benchmarks of 2× interpolation, in addition to the methods which have been already evaluated in [9, 14, 19] and the methods we used for 8× evaluation (EQVI, M2M, IFRNet, RIFE), we additionally retrained the GDConv [13]

and ST-MNet [4][1]. See Table 7 for comparisons between our reimplementations and the references evaluated with the pre-trained models. All of our reimplementations are comparable with the corresponding reference.

The comparison with FILM can be found in Table 8, where FILM is evaluated with the weight released officially, which was trained with $\mathcal{L}_1$ loss alone for higher scores. Our method still maintains superiority.

E. Architecture

The proposed framework comprises five deep modules, which are mainly responsible for contextual feature extraction $\mathcal{C}$, flow confidence estimation $\mathcal{W}$, motion refinement $\mathcal{M}$, range space estimation r, and null space estimation g, respectively.

As shown in Fig. 7, the input frame of $\mathcal{C}$ is filtered by three convolution layers separated by max pooling, yielding three-scale features. Then, each feature is resized back to the input scale via nearest upsampling and fused by one additional convolution layer. The final contextual features consist of the connection of the fused features and the original input frame. Remember that only half of the extracted features are used for confidence estimation. In this paper, we use the part containing the input frames as $\boldsymbol{C}_i^b$.

As shown in Fig. 8, the concatenated features fed in $\mathcal{W}$ are firstly compressed by two-layer convolutions and then encoded by three layers for estimating the confidences, which are normalized by Softmax.

As shown in Fig. 9, 10, and 11, $\mathcal{M}$, r and g are all equipped with similar three-scale Unets of different hyperparameters. The first block of each encoder transforms the channels of the input features and maintains the spatial resolution, while the others sequentially half the spatial resolution via strided convolution. The decoders are set up symmetrically with skip connections, whose upsampling layer is also implemented by strided convolution. Then, the outputs of each backbone are processed by two heads for estimating different targets.

Except for g, all outputs of convolution layers are normalized by the Group Normalization [17], whose group size is four. As g should be a Lipschitz continuous network for reasonable generalization [2], we use the Spectral Normalization [12] instead, which is approximated by one power iteration[2]. Besides that, we use GELU [5] for activating the normalized features.

[1]The original ST-MNet was trained on the mixed datasets of Vimeo-90K (septuplets) and BVI. The retrained model we used in our unified benchmarks for 2× VFI was kindly provided by the authors.

[2]We recommend referring to [12] for more details about the spectral normalization

F. More visual results

More visual results are illustrated in Fig. 12, 13, 14, 15, 16, and 17. We also provide more video comparisons in demo_7367.mp4.

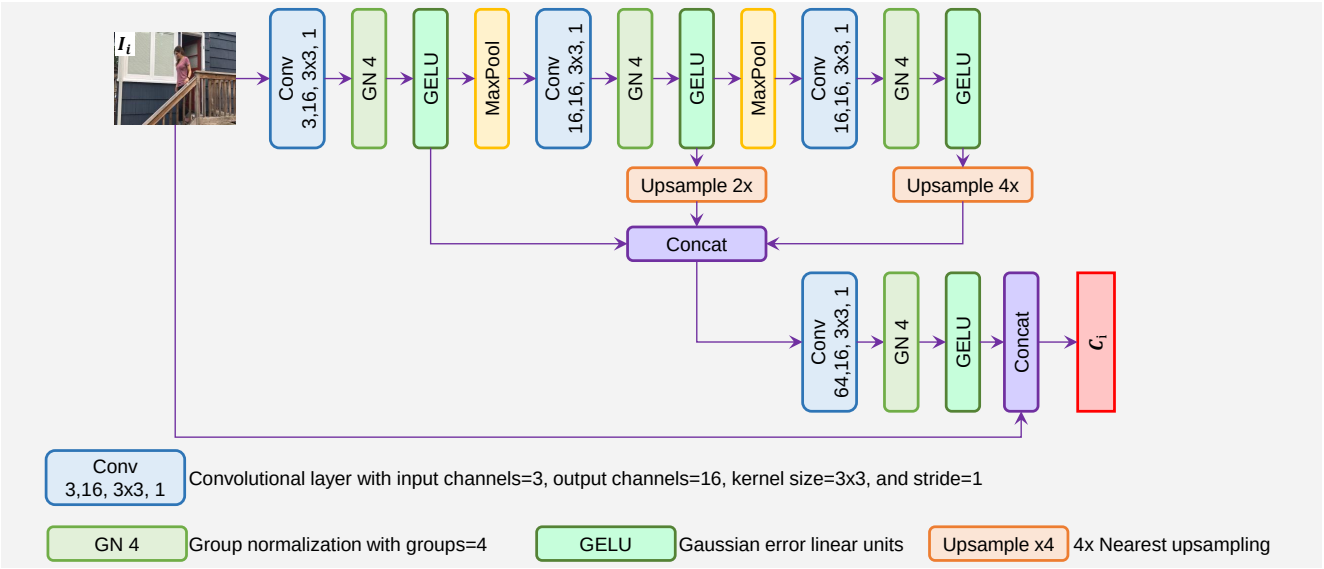

Figure 7. Architecture of the contextual network $\mathcal{C}$.

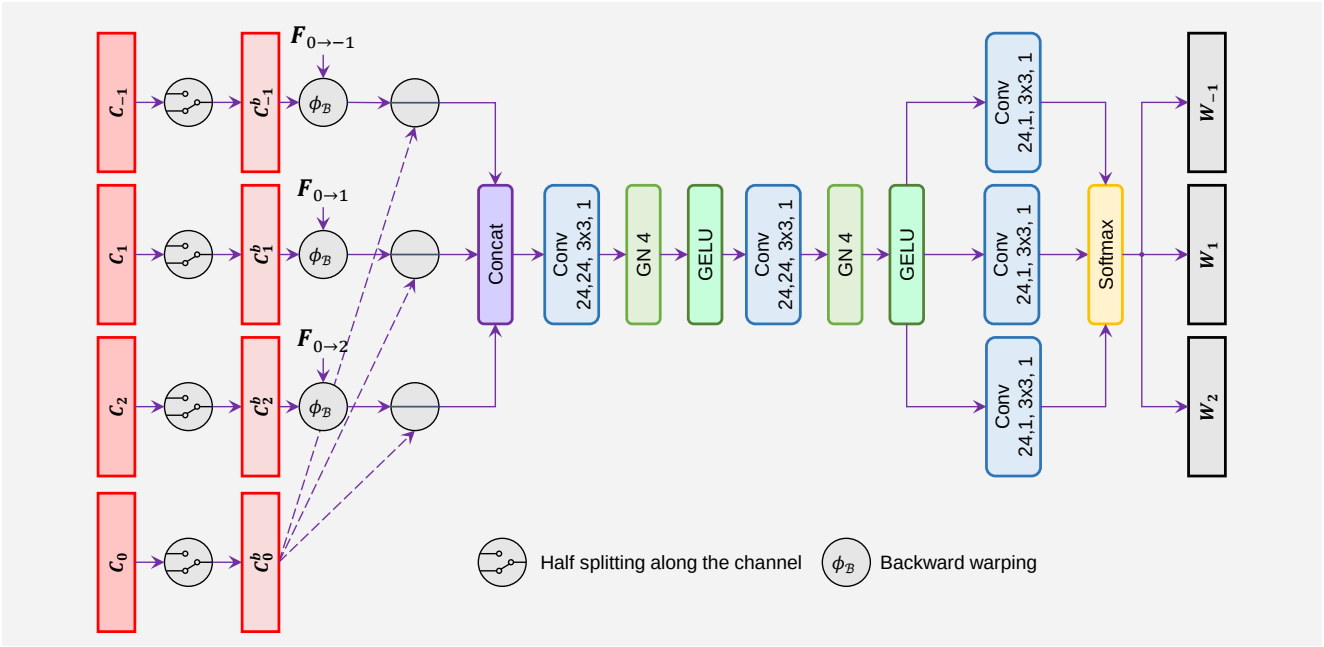

Figure 8. Architecture of the confidence estimation network $\mathcal{W}$.

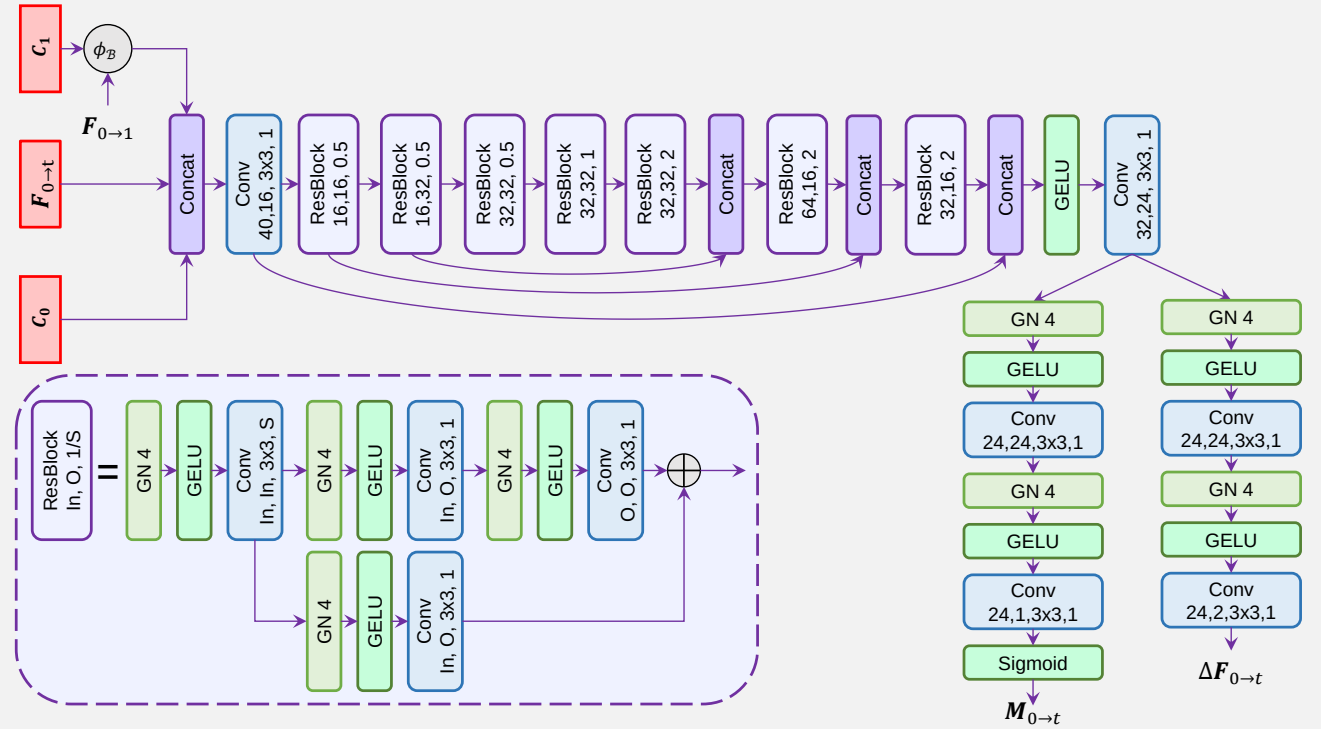

Figure 9. Architecture of the refinement network $\mathcal{M}$.

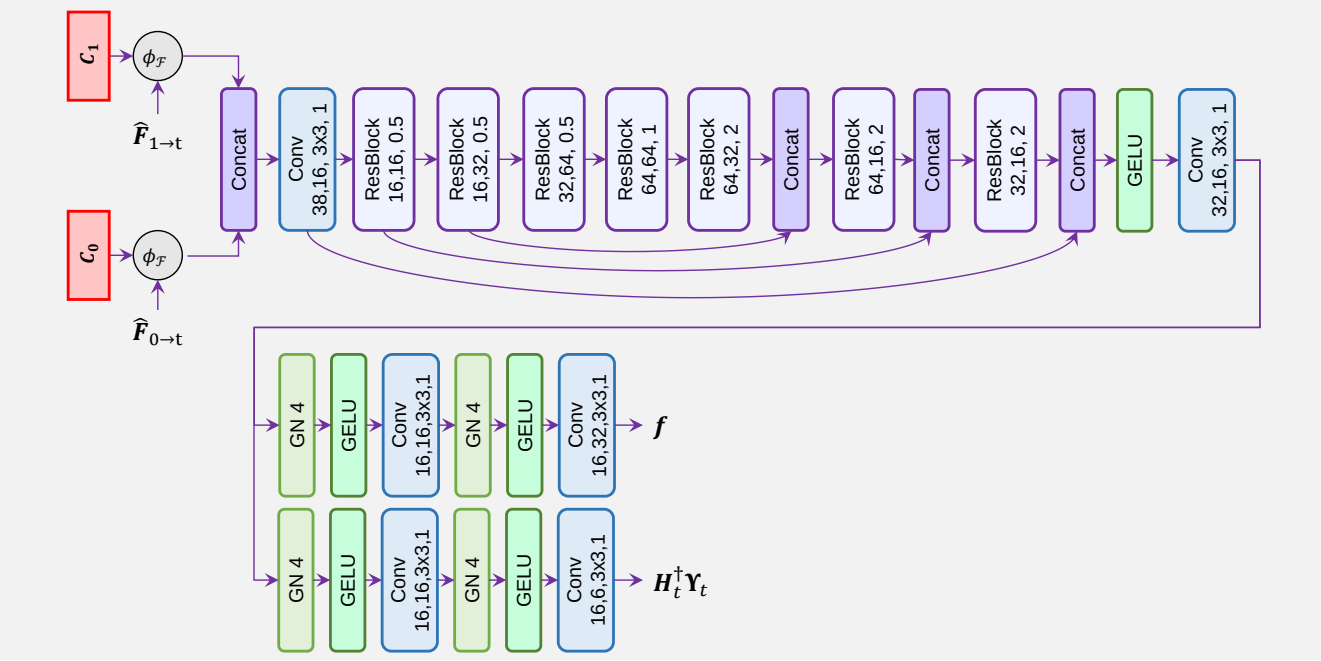

Figure 10. Architecture of the range space estimation network r.

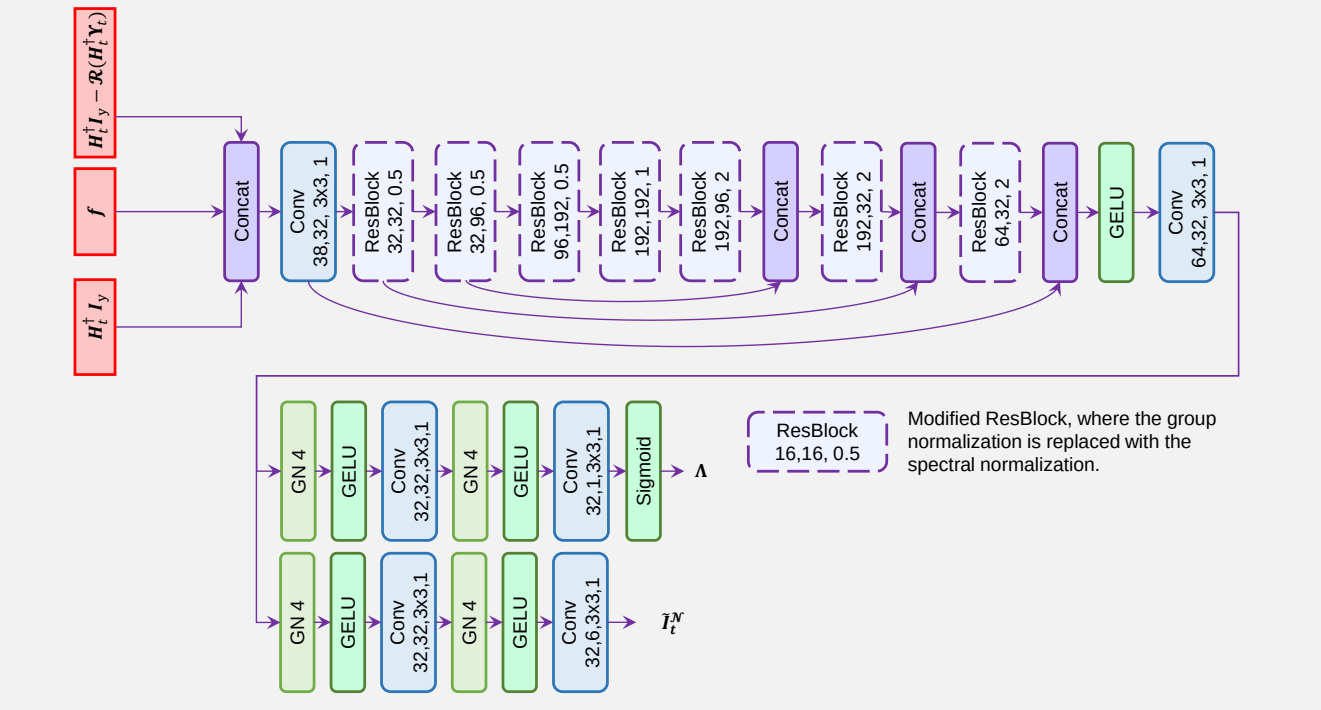

Figure 11. Architecture of the nullspace estimation network g.

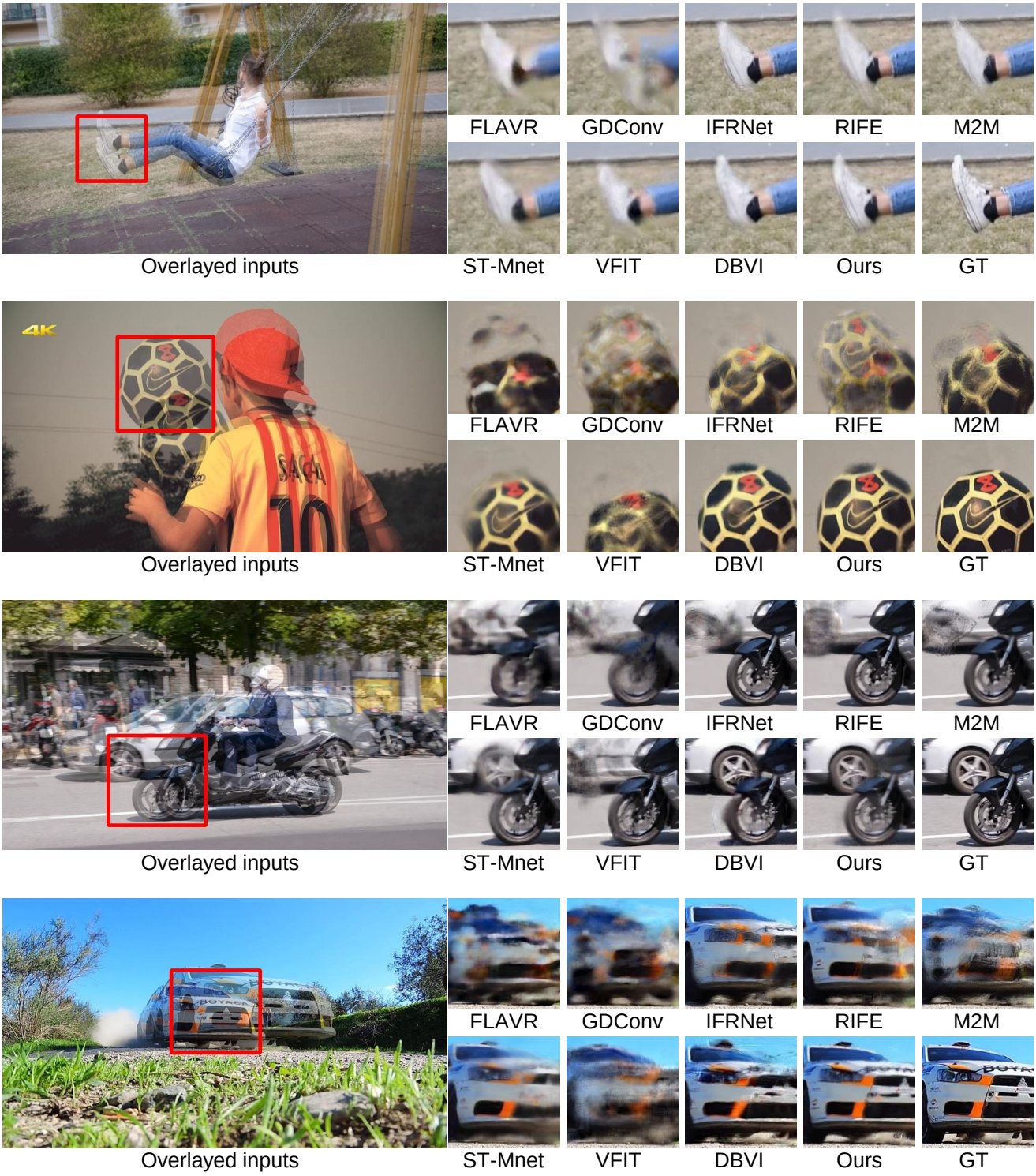

Figure 12. Visual comparisons on DAVIS. We overlay the nearest 2 input frames to illustrate the input motion. Best compared in the electronic version of this paper with zoom.

Figure 13. Visual comparisons on DAVIS.

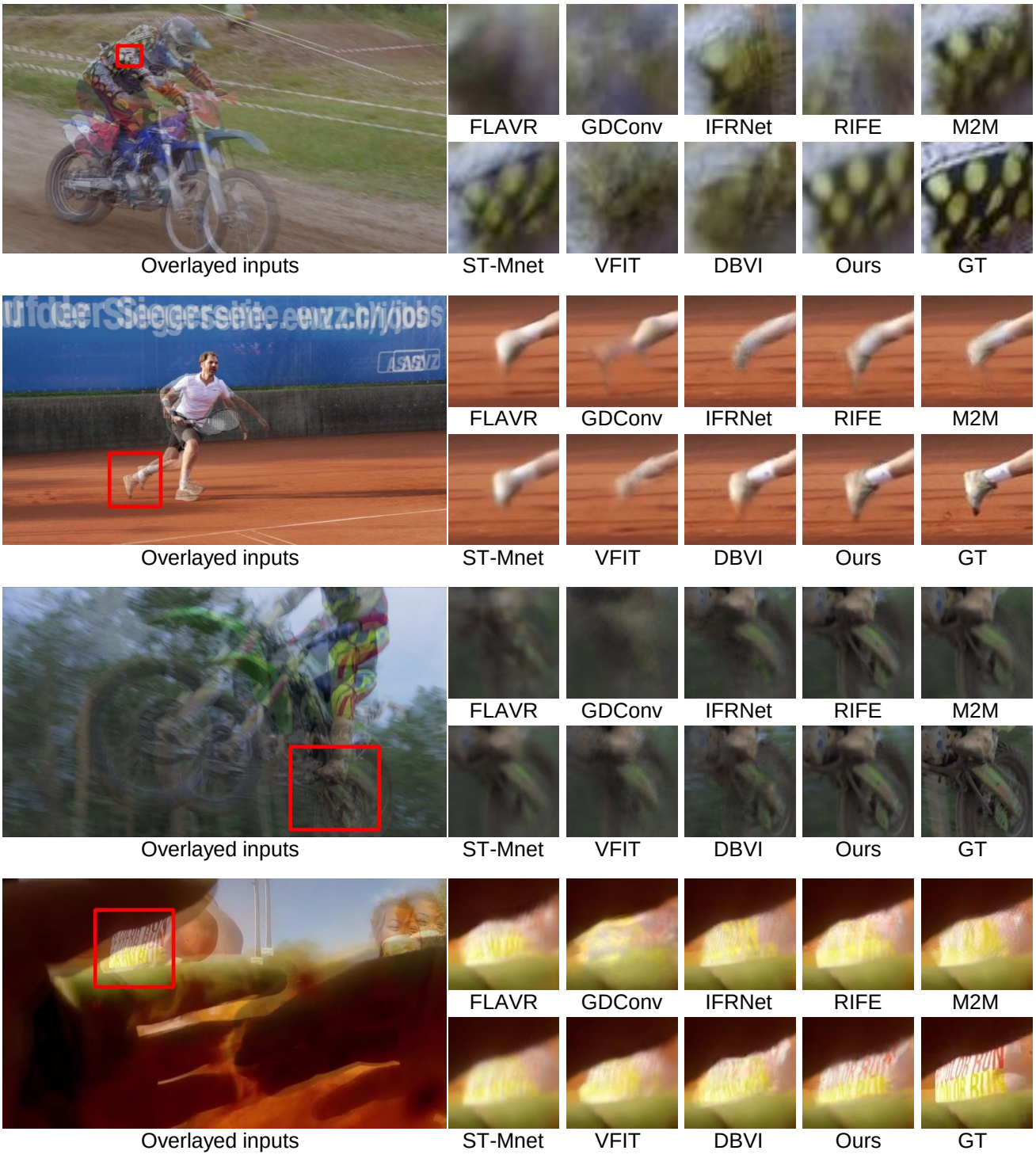

Figure 14. Visual comparisons on DAVIS.

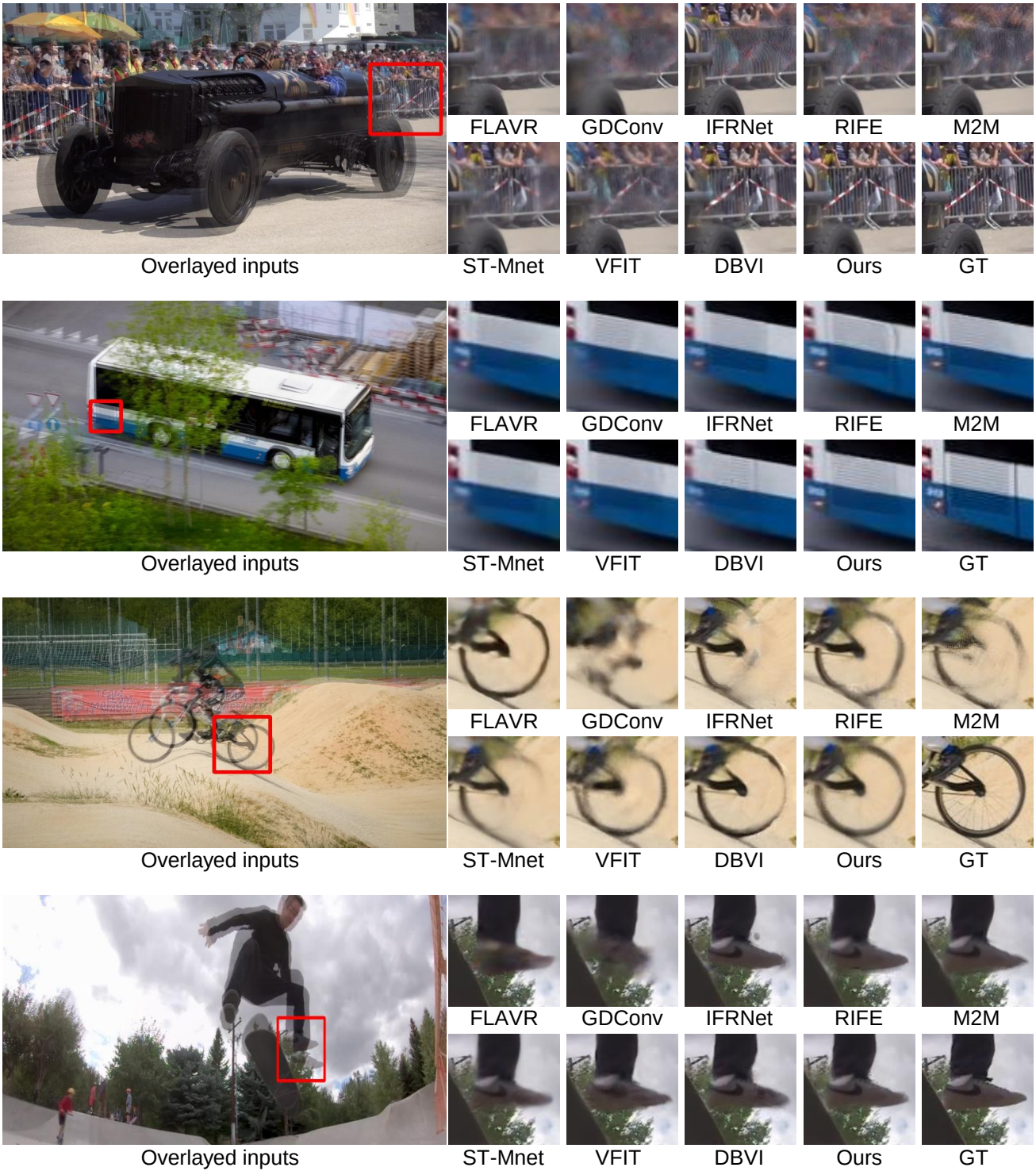

Figure 15. Visual comparisons on DAVIS (top 3 rows) and SNU-FILM (bottom 1 row).

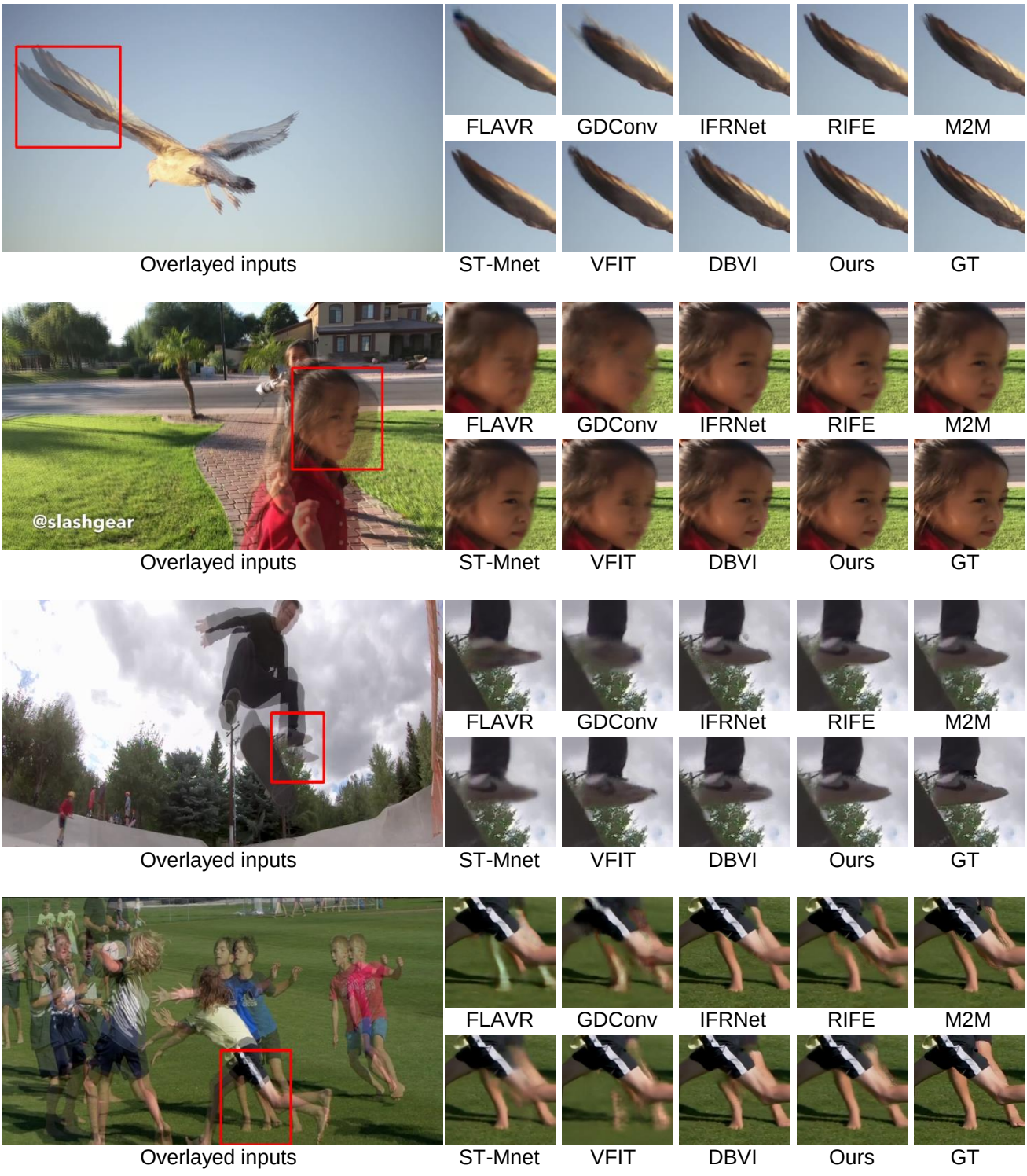

Figure 16. Visual comparisons on SNU-FILM.

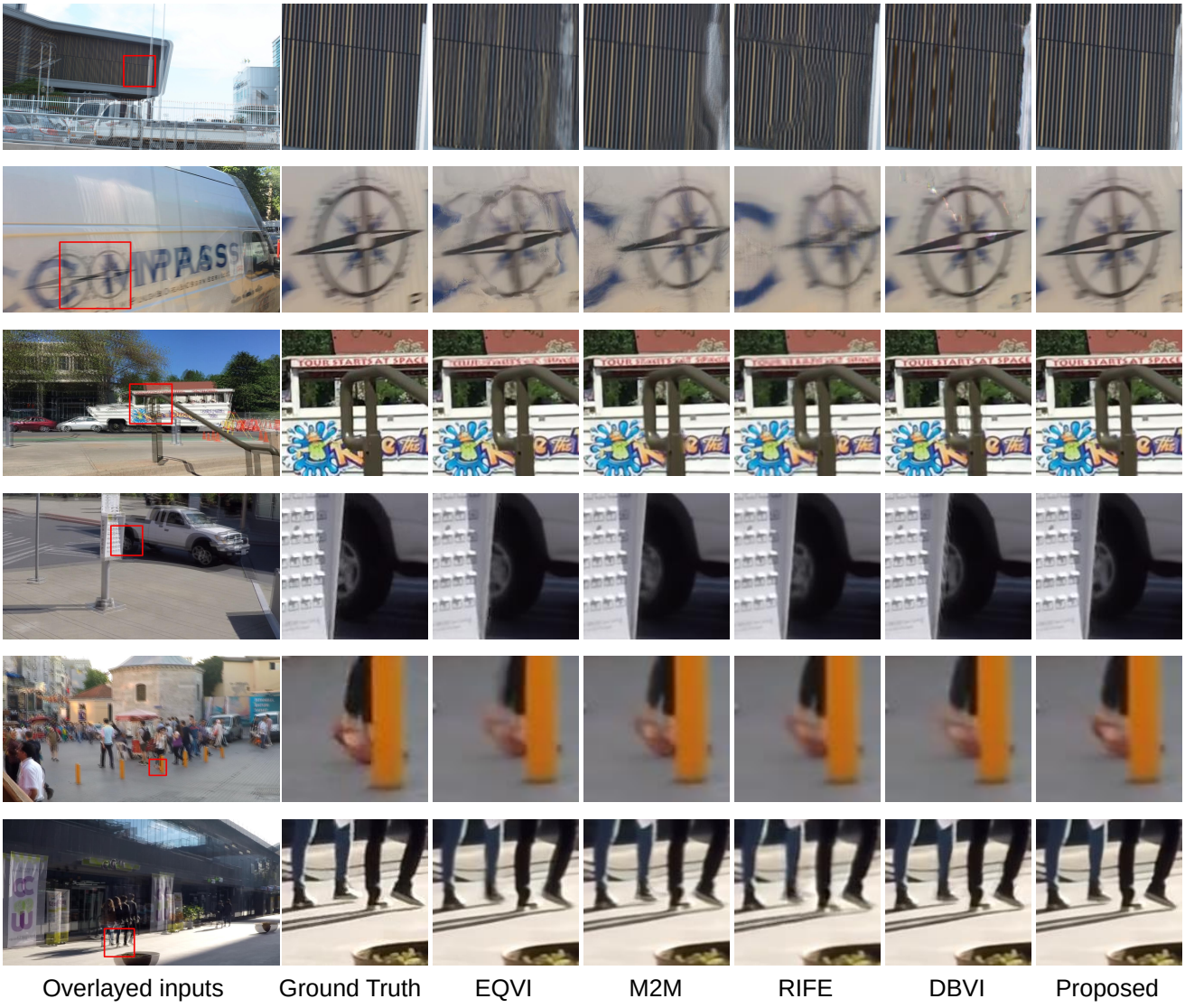

Figure 17. Visual comparisons on X4K1000FPS (top 1 row), Adobe240 (middle 3 rows) and GoPro (bottom 2 rows).

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3703–3712, 2019. 2

[2] Dongdong Chen and Mike E Davies. Deep decomposition learning for inverse imaging problems. In *Proc. Eur. Conf. Comput. Vis.*, pages 510–526, 2020. 3

[3] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–1, 2021. 2

[4] Duolikun Danier, Fan Zhang, and David Bull. ST-MFNet: A spatio-temporal multi-flow network for frame interpolation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3521–3531, June 2022. 3

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv:1606.08415*, 2016. 3

[6] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3553–3562, 2022. 2

[7] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proc. Eur. Conf. Comput. Vis.*, 2022. 2

[8] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. G. Learned-Miller, and J. Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9000–9008, 2018. 2

[9] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv:2012.08512*, 2020. 2

[10] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1969–1978, 2022. 2

[11] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *Proc. Eur. Conf. Comput. Vis. Workshops*, pages 41–56, 2020. 1, 2

[12] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Int. Conf. Learn. Representations*, 2018. 3

[13] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video frame interpolation via generalized deformable convolution. *IEEE Trans. Multimedia*, 24:426–439, 2021. 2

[14] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 17482–17491, 2022. 2

[15] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: Extreme video frame interpolation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 14489–14498, 2021. 2

[16] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. Eur. Conf. Comput. Vis.*, volume 12347, pages 402–419, 2020. 2

[17] Yuxin Wu and Kaiming He. Group normalization. In *Proc. Eur. Conf. Comput. Vis.*, volume 11217, pages 3–19, 2018. 3

[18] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1645–1654, 2019. 1, 2

[19] Zhiyang Yu, Yu Zhang, Xujie Xiang, Dongqing Zou, Xijun Chen, and Jimmy S. Ren. Deep bayesian video frame interpolation. In *Proc. Eur. Conf. Comput. Vis.*, pages 144–160, 2022. 2