Scale-Adaptive Feature Aggregation for Efficient Space-Time Video Super-Resolution (Appendix)

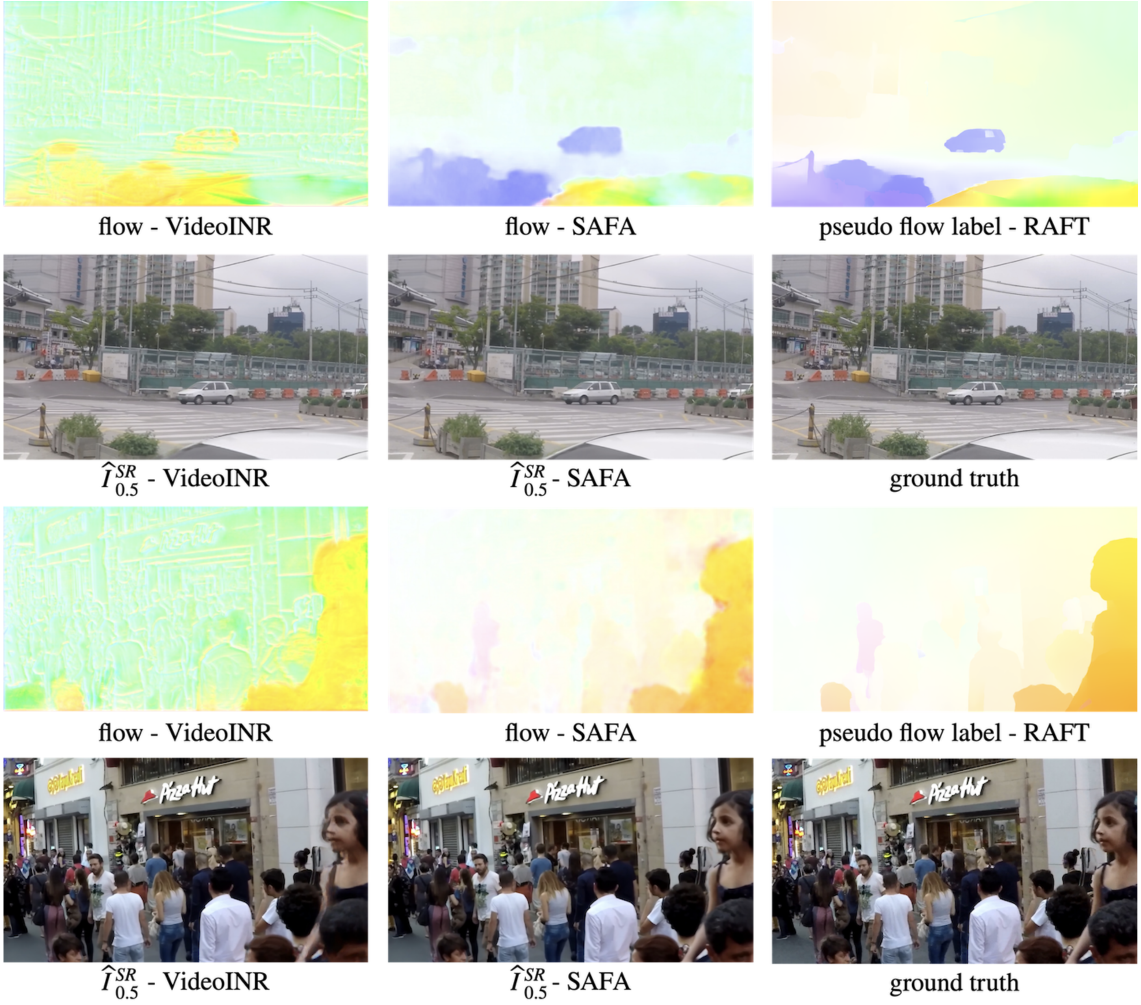


Figure 1. **Visualization of the intermediate flow estimated by SAFA. The pseudo label is obtained using RAFT [6].**

Societal Impact. STVSR methods can remove and synthesize frames, and the processed video may reflect different facts. This can be used by artists as a creative tool, but it may be used inappropriately. The related editing detection and reliability verification methods require further research.

1. Video Effect and Failure Case

The results of this Appendix are generated on the GoPro dataset [4]. The video demo is attached. We mainly compare SAFA with VideoINR [1] because it has state-of-the-art quantitative results. By observing the video results, we find that SAFA has an advantage over VideoINR [1] mainly when the object or camera motion is large. In addition, the recovery of regions with complex textures using SAFA is also basically better. SAFA still has two types of artifacts that affect the perception. 1) At the border of the video, some objects will move out of the screen, similar to VideoINR. At this time, it is difficult for the model to learn a reasonable transition, showing the fading in and fading out effects. Designing inpainting components may be able to remedy this shortcoming. 2) In some repetitive texture areas, such as fences, floor tiles, etc., the model may distort the lines. Such artifacts may be counteracted by adding smoothness constraints to the flow fields.

2. Architecture of Feature Extractor

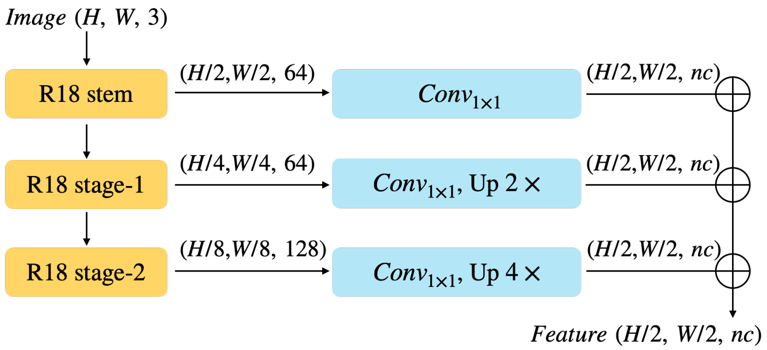


Figure 2. **Architecture of R18 Feature Extractor.** We use a 1 × 1 convolutional layer and bilinear up-sampling to adjust the number of channels and feature map size.

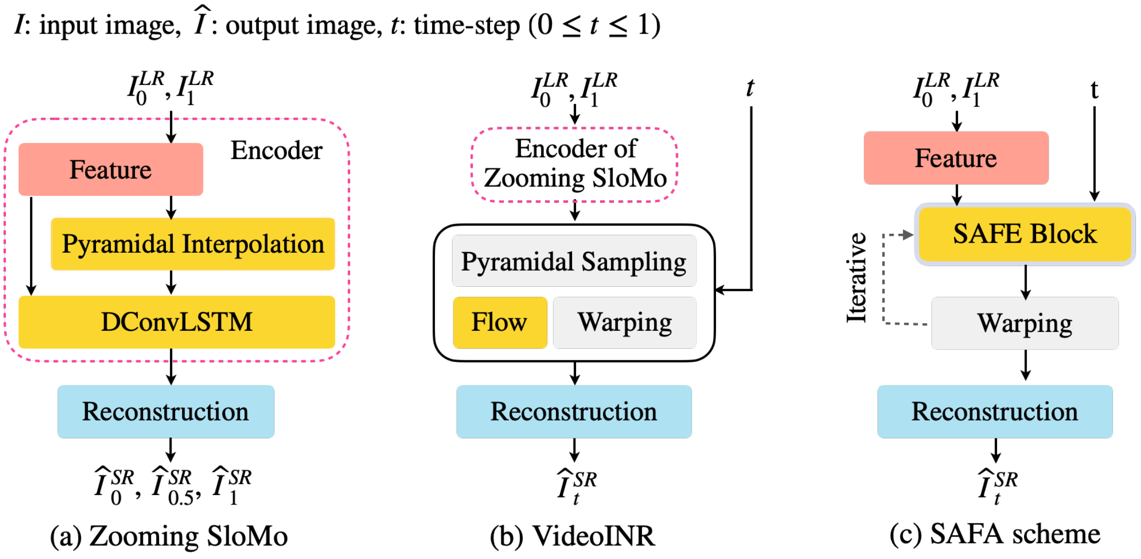


Figure 3. **Comparison of different structures.** We color blocks with similar functions the same.

3. Analysis of Intermediate Flow

Our proposed SAFA explicitly uses intermediate flows for feature propagation. To confirm that the architecture of SAFA indeed learns an optical flow-like representation, we show the visualization of approximated intermediate flow in Figure 1. We use the state-of-the-art optical flow model, pre-trained RAFT-things [6], to generate pseudo flow labels on the ground truth image and observe the difference.

We show that the intermediate flow estimated by SAFA is similar to the pseudo flow label of RAFT [6]. From the appearance, the flow pseudo label has sharper boundaries and is cleaner. This is mainly due to the difference in the definition of task-oriented flow [9] and optical flow. On the other hand, it is also partly due to the low resolution of the input of STVSR. Whereas the VideoINR [1] estimated flow is quite different. We can only see similar object edges. This demonstrates that the Zooming Slomo [7] encoder in VideoINR [1] has already undertaken part of the feature alignment. We depict the architecture of these previous methods and SAFA in Figure 3. The encoder is entangled with flow estimation. We argue that an explicit modular structure is important for designing efficient models. The intrinsic relationship of estimated flows at different time-steps is shown in Figure 4. It can be seen that SAFA maintains good consistency.

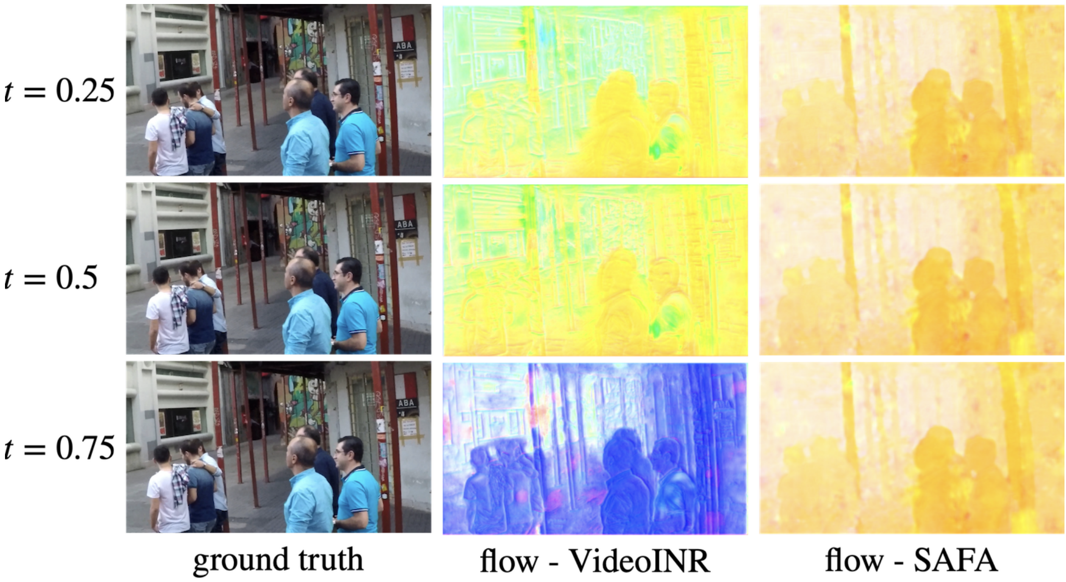


Figure 4. **Visualization of the estimated flow in different time-step.**

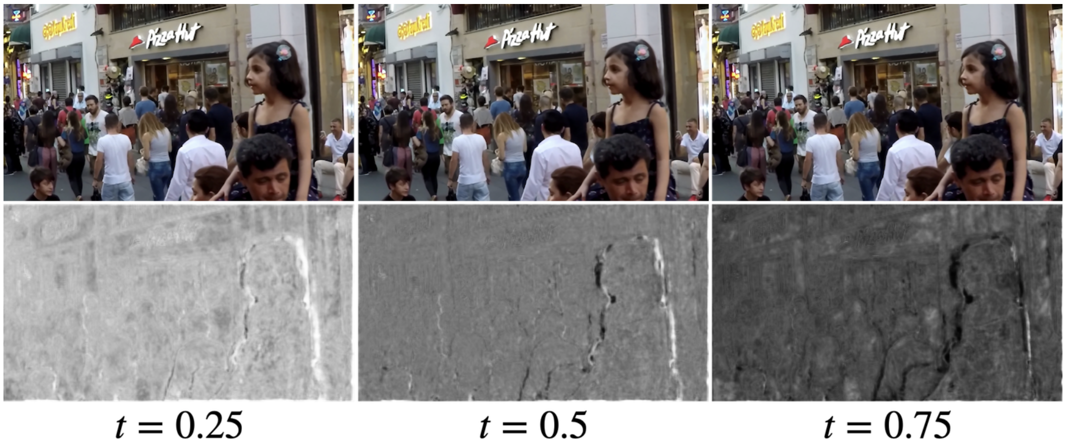


Figure 5. **Visualization of generated frames and the corresponding occlusion map.**

4. Analysis of Fusion Map and Refinement

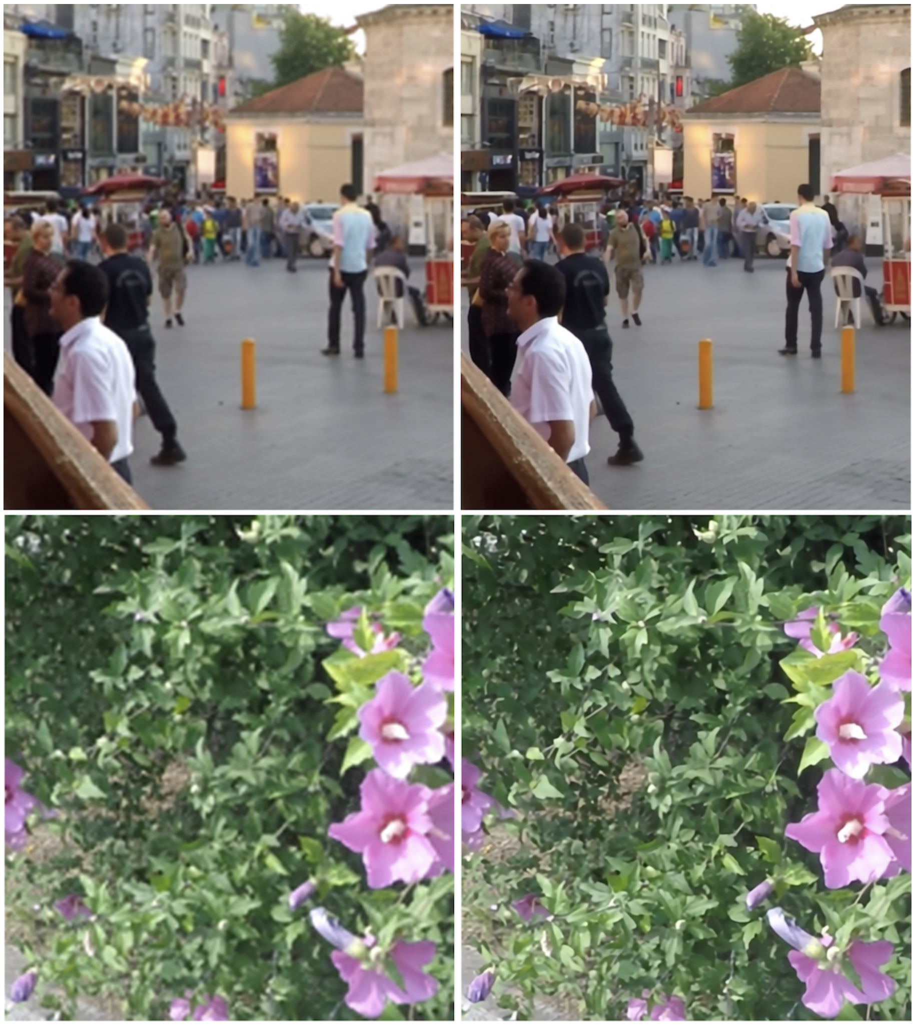
In SAFA, the formula that produces the final result is:

$$\widehat{I}_t^{SR} = [\mathbf{m} \odot \widehat{I}_{t\leftarrow 0} + (1 - \mathbf{m}) \odot \widehat{I}_{t\leftarrow 1}] + \Delta, \tag{1}$$

where m is usually called "fusion map" or "occlusion map" [2, 3, 5]. For non-occluded regions, it is used to weigh between the two results. Intuitively, when the time-step t is smaller, m is closer to 1 (visualized as white), making the model consider more the results from I_0. The occluded area often appears at the edge of the moving objects, and the model will choose one of the two results adaptively. The visualization is shown in Figure 5. Because I_0 and I_1 are both low-resolution images, it is intuitively impossible to obtain high-resolution images simply by warping and fusing them. The visual effect without feature-based refinement Δ (reconstruction model) is shown in Figure 6.

5. Comparison with Pyramidal Design.

Scale-selection increases the flexibility (and thus reduces the burden) of hand-crafted pyramid design models. On the other, we can share parameters at different scales (Table 3, **c7**). We fix the scale of the 6 blocks to (0.25, 0.25, 0.5, 0.5, 1, 1)



w/o Δ SAFA

Figure 6. **Visualization of generated frames with/without △. They have a noticeable difference in image sharpness.**

to construct a pyramidal-like structure. It cannot scale adaptively during inference.

Supplementary Table 1	GoPro PSNR	Adobe240 PSNR	# Param (M)
f1: SAFA	**31.28**	**30.97**	5.0
f2: Manually Set Scale	31.04	30.73	5.0

6. Specific Training Cost

For these methods for comparison, we u se open-sourced codes. On four Pascal TITAN X GPUs, TMNet [8] and VideoINR [1] take about 200 hours and 140 hours to train, respectively. While SAFA takes only 50 hours. The three methods use the same number of training iterations, TMNet [8] outputs 7 frames per iteration, while VideoINR [1] and SAFA output 3 frames at each forward pass. This is one reason why the training overhead of TMNet [8] is higher. In other words, TMNet undergoes more data iterations.

References

[1] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 4

[2] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[3] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[4] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[5] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[6] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[7] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[8] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Ming-Ming Cheng. Temporal modulation network for controllable space-time video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

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