Sparse Global Matching for Video Frame Interpolation with Large Motion

Supplementary Material

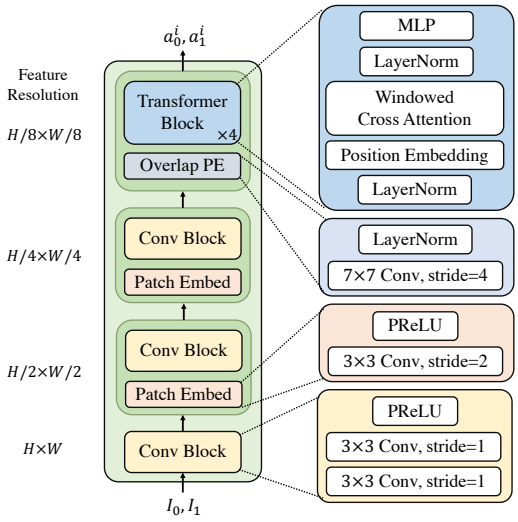

Figure A1. Model Structure of Local Feature Branch. $a_0^i, a_1^i, i \in \{0, 1, 2, 3\}$ is the extracted local feature, corresponding to the feature resolution of $\{H\times W, H/2\times W/2, H/4\times W/4, H/8\times W/8\}$

A. Local Feature Branch Model Structure

A.1. Local Feature Extractor

The structure of our local feature extractor is illustrated in Figure A1. As mentioned in Section 3.1, we adopt a CNN and Transformer hybrid structure for local feature extraction. This design diverges from that of EMA-VFI[37] by reducing the network depth. Furthermore, to enhance discriminability within local windows, we incorporate sine-cosine positional embeddings before the windowed cross-attention operation.

A.2. Flow Estimation

The Flow Estimation Structure, depicted in Figure 1, consists of two sequential Flow Estimation blocks, as shown in Figure A2. These two blocks are not identical. The first block, detailed in Figure A2 takes input frames $I_0, I_1 \in H \times W \times 3$ and local features $a_0^3, a_1^3 \in H/8 \times W/8 \times C_3$ as input. Its output includes the initial intermediate flow estimations $\widetilde{F}_{t\to0}, \widetilde{F}_{t\to1}$, along with the initial fusion map $\widetilde{M}$.

When pretraining on Vimeo-90K, $\widetilde{F}_{t\to0}, \widetilde{F}_{t\to1}$ and $\widetilde{M}$ are directly fed into the second block, along with warped images $I_{0\to t}, I_{1\to t}$ and the finer local features $a_0^2, a_1^2 \in H/4 \times W/4 \times C_2$. In the stages of finetuning and inference, however, $\widetilde{F}_{t\to0}, \widetilde{F}_{t\to1}$ and $\widetilde{M}$ are processed by the sparse global matching block for correction, resulting in refined flow estimations $F_{t\to0}, F_{t\to1}$ and an updated fusion map M, which are then input to the second block with $I_{0\to t}, I_{1\to t}$ and a_0^2, a_1^2.

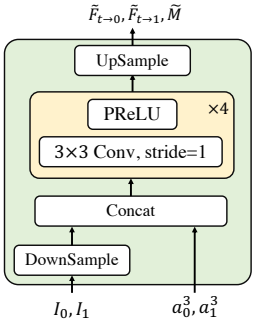

Figure A2. Model Structure of the Initial Flow Estimation Block.

A.3. Refine Net

We follow a similar design in RIFE[9]. We use Context Net to first extract the low-level contextual features. These features are then processed through backward warping, guided by the intermediate flows. The refinement stage involves a U-net shaped network, which can enhance the output frame in a residual form, using the warped features and flows.

B. Model Loss

We use the same training loss with EMA-VFI [37], which is the combination of Laplacian loss and warp loss, defined as:

$$\mathcal{L} = \mathcal{L}_{\text{lap}} + \lambda \sum_i \mathcal{L}_{\text{warp}}^i \tag{12}$$

where λ is the loss weight for warp loss. Following [37], we set $\lambda = 0.5$.

C. Generalizability

We apply our sparse global matching block on RIFE[9] and EMA[37] to show that our two-step strategy is applicable in more similar flow-based structures. The result is presented in Table A2 and Table A1 accordingly.

D. Scalability

We scaled our model to a bigger model size with 59.3M parameters, basically aligned with EMA-VFI-base [37] which

Table A1. **Results after applying sparse global matching block on EMA-VFI-small.** *1/N* means that we sparsely select 1/N points of the initial intermediate flows estimation.

	X-Test-L		SNU-FILM-L		Xiph-L	
	2K	4K	hard	extreme	2K	4K
EMA-VFI	29.51/0.8775	28.60/0.8733	28.57/0.9189	23.18/0.8292	30.54/0.8718	28.40/0.8109
EMA-VFI-1/8	29.65/0.8788	28.77/0.8753	28.62/0.9192	23.31/0.8306	30.59/0.8712	28.61/0.8114
EMA-VFI-1/4	29.81/0.8816	28.91/0.8776	28.68/0.9196	23.41/0.8326	**30.64**/0.8720	28.78/0.8128
EMA-VFI-1/2	**30.12/0.8886**	**29.24/0.8840**	**28.70/0.9196**	**23.46/0.8343**	30.63/**0.8722**	**28.91/0.8146**

Table A2. **Results after applying sparse global matching block on RIFE.** *1/N* means that we sparsely select 1/N points of the initial intermediate flows estimation.

	X-Test-L		SNU-FILM-L		Xiph-L	
	2K	4K	hard	extreme	2K	4K
RIFE	29.87/0.8805	28.98/0.8756	28.19/0.9172	22.84/0.8230	30.18/0.8633	28.07/0.7982
RIFE-1/8	30.50/0.8902	29.52/0.8838	28.61/0.9189	23.35/0.8298	30.26/0.8637	28.45/0.8023
RIFE-1/4	30.68/0.8981	29.72/0.8901	28.63/0.9191	**23.52**/0.8340	30.30/0.8643	28.66/0.8048
RIFE-1/2	**30.88/0.9034**	**29.90/0.8944**	**28.66/0.9195**	**23.52/0.8350**	**30.35/0.8656**	**28.69/0.8066**

has 65.7M parameters. Results listed in Table D3. From Table D3, we can draw the following conclusion.

As more points are incorporated into sparse global matching, the performance gradually saturates. This observation is intuitive, considering that not every aspect of the initial estimated flow is inaccurate, nor is every aspect of the global matching flow entirely precise. This is evidenced by Table 5, where the merge block is absent in this ablation. However, upon integrating the merge block (refer to Table 3), with more points are involved, up to full global matching, performance still has a little improvement with increased point involvement, meaning that there is still potential for enhancement within the local branch of the smaller model with the help of our merge block.

But when we change our model with a larger local branch with more parameters, the capacity of the local branch becomes stronger. Consequently, it becomes evident that involving all points in global matching leads to performance degradation compared to utilizing only half the points, thus affirming our pursuit of sparsity.

E. Model Size Comparisons

We conduct a series of parameters and runtime comparisons on an Nvidia RTX 2080Ti GPU. Illustrated in Table E4, our local branch is aligned with EMA-VFI-small in terms of runtime and parameters, therefore, we mainly compare our results with EMA-VFI-small model setting.

Table D3. **Results on a larger local branch.** Note that we disable the test-time augmentation when testing for direct comparison.

	XTest-L-2K	
	PSNR	SSIM
EMA-VFI [37]	30.85	0.9005
Ours-local branch	30.68	0.9010
Ours-1/8	31.10	0.9080
Ours-1/4	31.19	0.9102
Ours-1/2	**31.27**	**0.9115**
Full Global Matching	31.20	0.9104

Table E4. **Comparisons of model size and corresponding performance.** We only list the X-Test-L-2K results for simplicity.

	Inference Time on 512x512 Resolution	Parameters	X-Test-L-2K	
			PSNR	SSIM
RIFE	**10ms**	10M	29.87	0.8805
EMA-VFI-small	25ms	14.5M	29.51	0.8775
EMA-VFI-base	132ms	65.7M	30.85	0.9003
XVFI	22ms	**5.6M**	29.82	0.8493
BiFormer	59ms	11M	30.32	0.9067
Ours-local-branch	23ms	15.4M	30.39	0.8946
Ours-1/2-Points	74ms	20.8M	**31.03**	**0.9075**

F. Different Flow Reversal Teqniques

We compare our flow shift strategy with the flow reversal layer in [35], complementary flow reversal in [29], lin-

Table F5. **Comparisons between different flow reversal techniques.**

	X-Test-L-2K		X-Test-L-4K	
	PSNR	SSIM	PSNR	SSIM
flow reversal layer [35]	30.57	0.8977	29.45	0.8886
CFR [29]	30.73	0.9001	29.63	0.8913
linear combination [13]	30.69	0.9000	29.59	0.8907
CNN layer	30.18	0.8932	29.13	0.8853
linear reversal	30.70	0.9017	29.59	0.8924
flow shift (Ours-1/8)	**30.83**	**0.9022**	**29.73**	**0.8928**

Table G6. 8× **Interpolation Results on X-Test (PSNR)**.

	X-Test (8× interpolation)	
	2K	4K
EMA-VFI-small-t [37]	31.75/0.9164	30.59/0.9078
RIFE-m [9]	32.23/0.9229	31.09/0.9141
FILM [27]	31.50/0.9162	OOM
Ours-1/2	**32.38/0.9272**	**31.35/0.9179**

ear combination in [13], CNN layer and linear reversal on Ours-1/8 setting. Shown by Table F5, our flow-shifting strategy is the most suitable for sparsely sampled flows.

G. Interpolating multiple frames into two frames

We follow the recursive interpolation method in FILM [27] and present our multi-frame interpolation (between two frames) results in Table G6.

H. Finetuning or Training From Scratch

In our experiments, we conducted training from scratch on the Vimeo-90K [4] dataset using a sparse global matching block with full global matching. This approach still demonstrated noticeable effects attributed to the global matching process. However, as indicated in Table H7, the ability to capture large motion was not on par with the results obtained after finetuning on a dataset with larger motion. Therefore, finetuning on a small batch of large motion datasets (X-Train) is more efficient than training from scratch on a large batch of small motion datasets (Vimeo-90K). This efficiency is evidenced by the reduced number of required training steps, with finetuning necessitating only 13.7k steps as opposed to 480k steps for training from scratch. This finding aligns with the observations reported in FILM [27], suggesting that large motion datasets can bring large motion capturing ability.

Table H7. **Comparisons between from scratch and finetuning.**

	X-Test-L-2K		X-Test-L-4K	
	PSNR	SSIM	PSNR	SSIM
Ours-local-branch	30.39	0.8946	29.25	0.8861
Global-From Scratch	30.63	0.9012	29.61	0.8958
Global-Finetuning	**31.03**	**0.9074**	**29.95**	**0.8974**

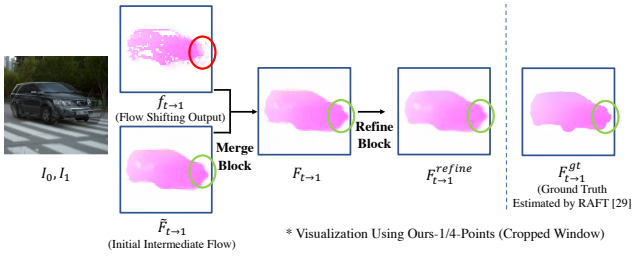

Figure I3. Visualization of Matching Failure and Repair

Operations	Inference Time
Local Feature Branch	23 ms
Flow Compensation Branch	50.6 ms
- Global Feature Extraction	**45 ms**
- Others	5.6 ms
(512 × 512 Resolution) Total	73.6ms

Table J8. **Time Profile on Our Proposed Algorithm.** Measured on an Nvidia RTX 2080Ti GPU.

I. Failed Matching

When matching fails, the merge block in our method can adaptively merge the flows, depressing the impact of matching failure. Moreover, we have a refine block to further repair the merged flow. We also provide a visualization in Figure I3.

J. Inference Speed Bottleneck

As shown in Table J8, the bottleneck of our pipeline lies in the global feature extractor, instead of other parameter-free components. One naive solution is to replace it with a simpler and lighter global feature extractor in the future. And another solution is to distill the global feature extraction ability from GMFlow [34] to our own feature extractor, which needs more experiment and probably even training data from optical flow datasets.