Towards An End-to-End Framework for Flow-Guided Video Inpainting
– *Supplementary Materials* –

Zhen Li[1*] Cheng-Ze Lu[1*] Jianhua Qin[2] Chun-Le Guo[1†] Ming-Ming Cheng[1]
[1]TMCC, CS, Nankai University [2]Hisilicon Technologies Co. Ltd.
zhenli1031@gmail.com, czlu919@outlook.com, qinjianhua@hisilicon.com
{guochunle, cmm}@nankai.edu.cn

1. Architecture and Training Details

Architecture. In our model, the encoder and the decoder use the same architecture as FuseFormer [5]. The channel dim C of the encoder and the decoder is set as 128. A lightweight model SPyNet [7] is employed as our flow completion module for computational efficiency. To utilize the learned flow prior in original SPyNet, we use pre-trained weights to initialize this module. The architecture details of the T-PatchGAN are identical to previous works [1, 5, 10]. The kernel size K and the group number G of deformable convolution are set as 3 and 16, respectively. The number of focal transformer blocks N is set as 8 and the embedded dim of tokens C_e is set as 512. The embedded spatial dimension $M \times N$ is 20×36. The size of partitioned sub-window $s_t \times s_h \times s_w$ is set to $(T_l + T_{nl}) \times 5 \times 9$. At the end of the content hallucination module, we use a soft composite operator [5] to composite the embedded tokens to features, which share the same spatial size as the original ones.

Training details. For training objectives, the weights of $\mathcal{L}_{rec}$, $\mathcal{L}_{adv}$, and $\mathcal{L}_{flow}$ are 1, 10^{-2}, and 1, respectively. Taking the memory limitations of GPUs into account, we resize all frames from videos into 432×240 for training, evaluation, and test. During training, the numbers of local (T_l) and non-local frames (T_{nl}) are 5 and 3, respectively. Local frames are continuous clips, while non-local frames are randomly sampled from videos for training. Following STTN [10] and FuseFormer [5], during evaluation and test, we use a sliding window with the size of 10 to get local neighboring frames and uniformly sample the non-local neighboring frames with a sampling rate of 10. We adopt Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.99$. The final model is trained for 500K iterations, and the initial learning rate is set as 0.0001 for all modules and reduced by the factor of 10 at 400K iteration. In our ablation studies, we train the model for 250K iterations. We use 8 NVIDIA Tesla V100 GPUs for training and the batch size is set as 8. Our code is available [1] for reproducibility.

2. More Experiments

2.1. Completing flows in a offline manner.

To verify the effectiveness of online flow completion, we prepare completed flows using the FGVC [3] flow completion module in an offline manner. We then retrain a model with the FGVC completed flows. The PSNR value of this model is slightly higher than our end-to-end setting (32.38 vs. 32.35 (dB)). However, the inference speed is much slower than ours (1.21 vs. 0.16 (s/frame)).

2.2. Taking a deeper look to flow-guided feature propagation module

To further investigate the effectiveness of the feature propagation module, we visualize averaged local neighboring features with the temporal size of 5 before conducting content hallucination in Fig. 1. The four cases in Fig. 1 correspond to the four variants in the Tab. 3 of our main paper. For the model without feature propagation (Fig. 1(a)), obviously, we can see that corrupted regions from all frames still exist in these features, further restricting the performance of content hallucination. For the model only using flow-based warping (Fig. 1(b)) or deformable convolution-based warping (Fig. 1(c)), corrupted regions are filled with the contents warped from adjacent frames. And the deformable convolution-based warping can generate smoother content than flow-based one due to more sampling feature points. However, especially for the last two temporal features (last two columns in Fig. 1), the regions filled by the model without flow guidance have more distinct boundaries in contrast to flow-based warping, which implies that less faithful content are propagated without motion information. Through adopting deformable convolution with flow guidance, the final propagation module (Fig. 1(d)) fills the holes with the most reasonable and natural content among all cases. This is a promising demonstration of the mutually beneficial relationship between deformable offsets and completed flow fields.

*Equal contribution
†C.L. Guo is the corresponding author.

[1] https://github.com/MCG-NKU/E2FGVI

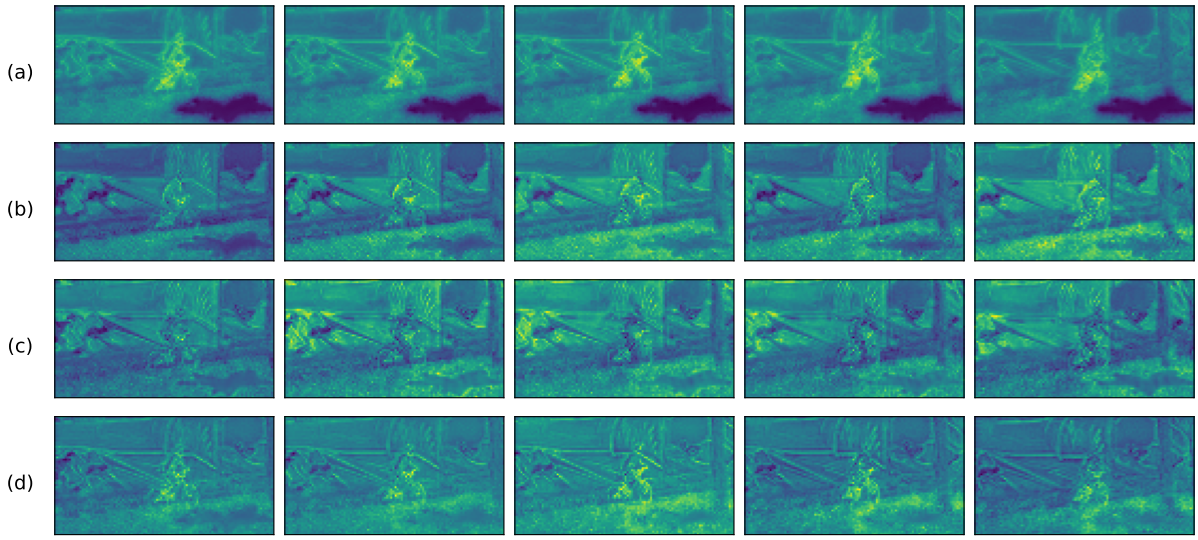

Figure 1: Visualization of the frame-wise average features before feeding into the content hallucination stage under different experimental settings: (a) without flow-guided feature propagation, (b) flow-guided feature propagation without deformable convolution (Eq. 3 of the main paper), (c) feature propagation without flow guidance, and (d) final flow-guided feature propagation module with the assistance of both flow fields and deformable convolution. (**Zoom-in for best view**)

2.3. Study of the hallucination ability

To purely evaluate the hallucination ability of our method, we first pre-fill the pixels which can be traced by flow fields [2]. The remaining unfill pixels are thus most likely not visible in other video frames. We then feed the pre-filled videos to an image inpainting model [9] and our model, respectively. Our hallucinated result has a much larger PSNR value than the image inpainting model on DAVIS dataset (31.74 vs. 30.80 (dB)).

Table 1: Parameters comparisons. FuseFormer* denotes a larger version of original FuseFormer.

	FuseFormer [5]	FuseFormer*	E^{2}FGVI
Params. (M)	36.6	41.6	41.8
PSNR/SSIM	31.74/0.9662	31.91/0.9669	32.35/0.9688

2.4. Parameter comparison

We report the parameters in Tab. 1. Although our method consumes $\sim$14% more parameters than the SOTA method (*i.e.*, FuseFormer [5]), it achieves a great trade-off between performance and computational complexity among other methods (see Tab. 1). For further comparison, we add residual blocks in FuseFormer to achieve similar parameters with ours. Our method still performs better than the larger version of FuseFormer.

2.5. More Qualitative Results

In this section, we provide additional visual results on two benchmark datasets, including YouTube-VOS [8] and DAVIS [6], to further show the superiority of the proposed E^{2}FGVI. The reconstruction results of CAP [4], FGVC [2], and FuseFormer [5] are presented for comparisons. As shown in Fig. 2-5, our E^{2}FGVI can generate more faithful textural and structural information and more coherent contents in masked regions than other methods. **Our demo is shown in our project page.**

References

[1] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. *ICCV*, 2019. 1

[2] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, 2020. 2

[3] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 1

[4] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *ICCV*, 2019. 2

[5] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *ICCV*, 2021. 1, 2

[6] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 5, 6

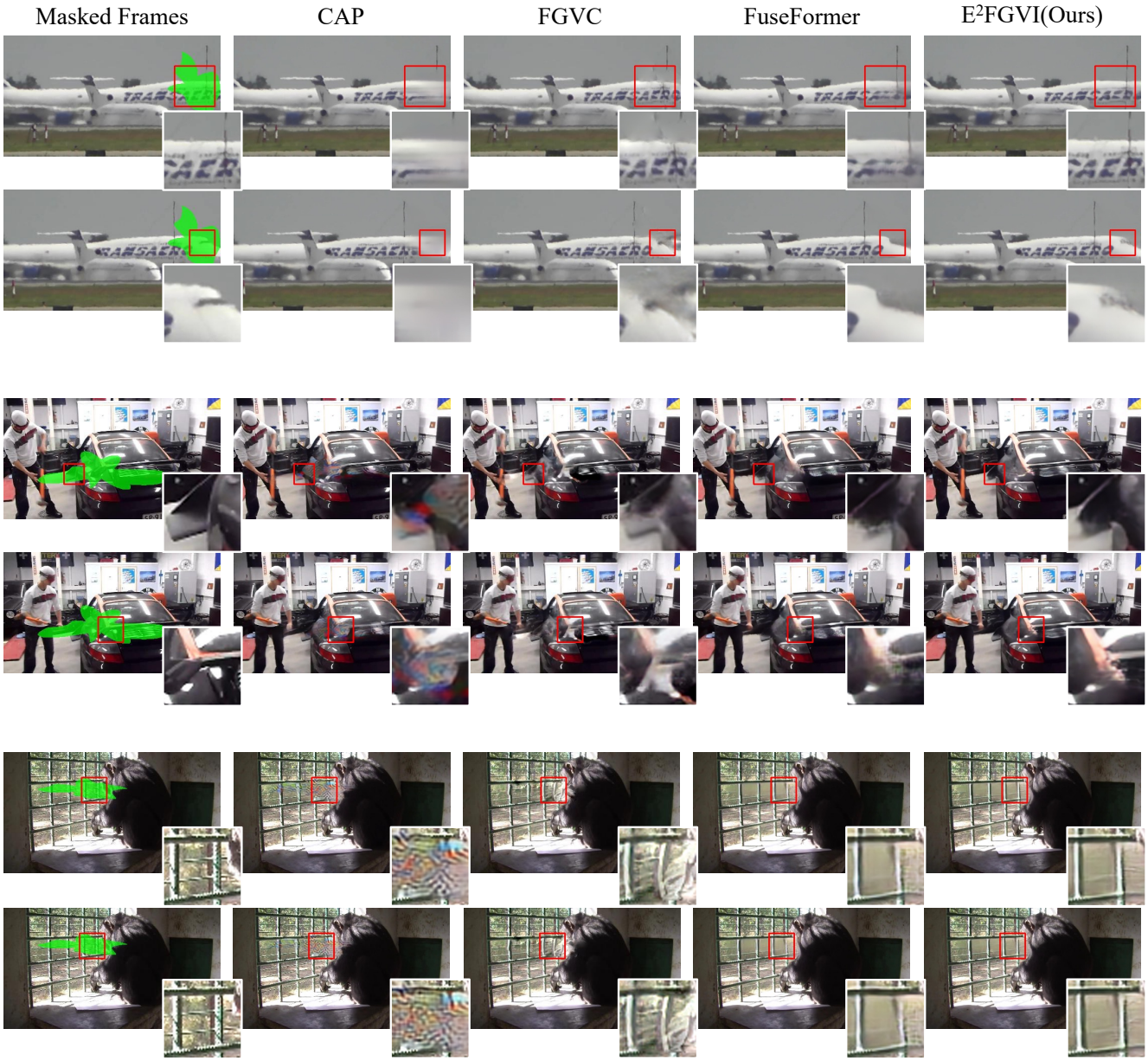

Figure 2: Qualitative video completion results on YouTube-VOS [8].

[7] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 1

[8] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 2, 3, 4

[9] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. 2

[10] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, 2020. 1

Figure 3: Qualitative video completion results on YouTube-VOS [8].

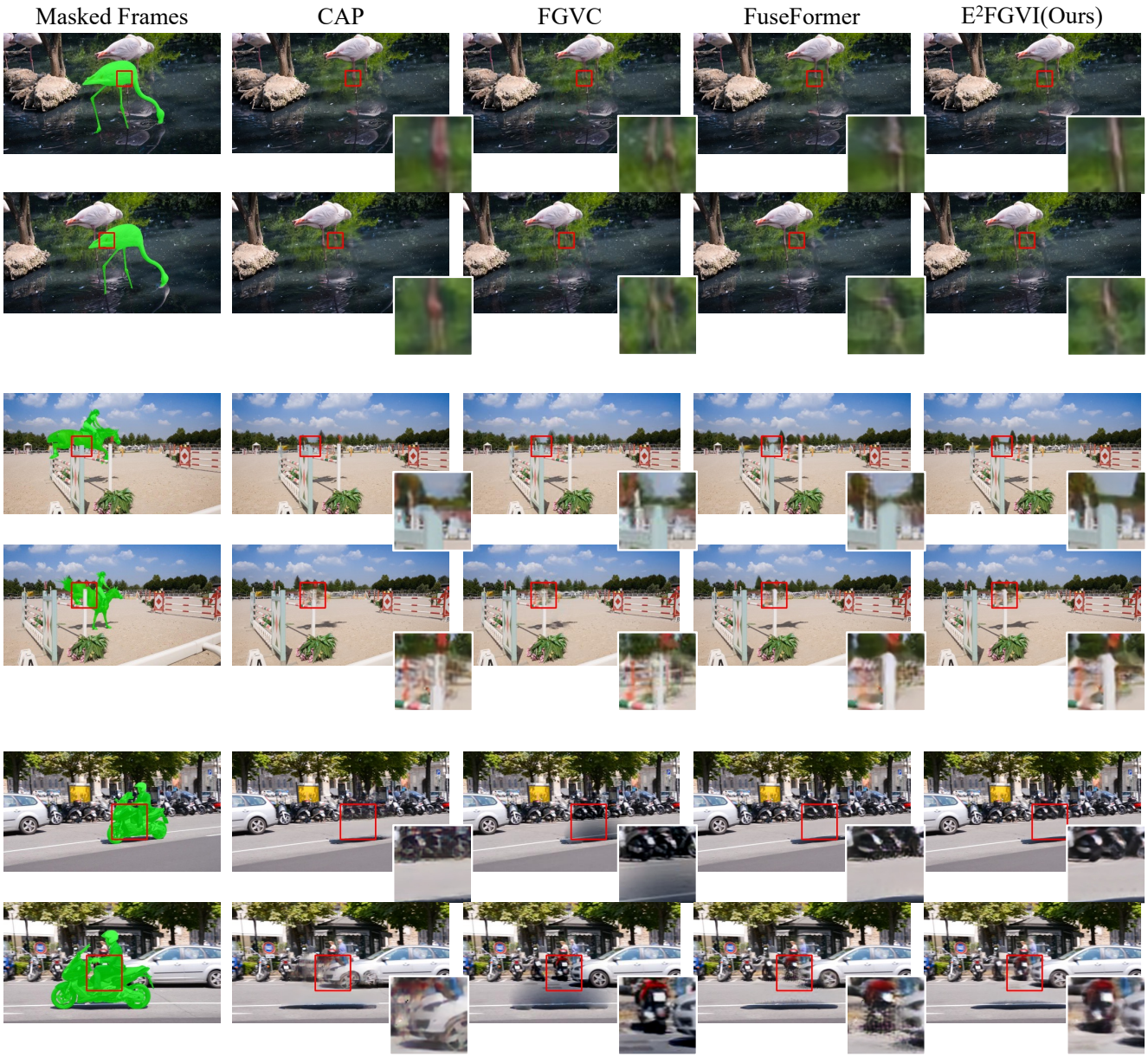

Figure 4: Qualitative object removal results on DAVIS [6].

Figure 5: Qualitative video completion results on DAVIS [6].