Video Super-Resolution Transformer with Masked Inter&Intra-Frame Attention

Supplementary Material

In this file, we provide more implementation and experimental details which are not included in the main text. In Section A, we provide more implementation details and more information about the dataset. In Section B, we provide two more instantiations of our model, i.e., the MIA-VSR-small and MIA-VSR-tiny to compare with other light-weight VSR models. In Section C, we provide a further result of fine-tuning our MIA-VSR model by longer sequences. More visual examples of different VSR models are presented in Section D.

A. Dataset and implementation details

A.1. Datasets

REDS [7] REDS is a widely-used video dataset for evaluating video restoration tasks. It has 270 clips with a spatial resolution of 1280 × 720. We follow the experimental settings of [1, 2, 9] and use REDS4 (4 selected representative clips, i.e., 000, 011, 015 and 020) for testing and training our models on the remaining 266 sequences.

Vimeo-90K [10] Vimeo-90K is a commonly used dataset which contains 64,612 training clips and 7,824 testing clips (denoted as Vimeo90K-T). Each clip contains 7 frames of images with a spatial resolution of 448 × 256. We follow the experimental settings of [1, 2, 9] and evaluate our proposed MIA-VSR method with the Vimeo-90K dataset.

Vid4 [5] Vid4 is a classical dataset for evaluating video super-resolution methods. It contains 4 video clips (i.e., calendar, city, foliage and walk) and each clip has at least 34 frames (720 × 480). We follow the experimental settings of [1, 2, 9] and use the 4 sequences in the Vid4 dataset to compare different VSR models.

A.2. Training and testing details

Implementation details for the REDS model. We train our MIA-VSR model with the REDS [7] training dataset with zooming factor 4. We follow the experimental settings of BasicVSR++ [2] and train our MIA-VSR model for 600K iterations. The initial learning rate is set as 2×10^{-4}. We train our model with Adam optimizer and the batch size is set as 24. In the testing phase, we evaluate MIA-VSR model's performance on the REDS4 [7] dataset.

Implementation details for the Vimeo-90K and Vid4 model. We train our MIA-VSR model with the Vimeo90K [7] training dataset with zooming factor 4. We follow the experimental settings of BasicVSR++ [2] and train our MIA-VSR model for another 300K iterations with its well-trained model on the REDS dataset. The initial learning rate is set as 1×10^{-4}. We train our model with Adam optimizer and set the batch size at 24. In the testing phase, we evaluate the performance of the MIA-VSR model on Vimeo90K-T [10] and Vid4 [5] datasets.

B. Light-weight MIA-VSR models

In order to compare with recently proposed light-weight methods, we establish two light-weight versions of our MIA-VSR model, i.e., the MIA-VSR-small and the MIA-VSR-tiny model. Both the MIA-VSR-small and the MIA-VSR-tiny models contain 4 feature propagation modules and each feature propagation module comprises 6 MIIA blocks with a skip connection. The spatial window size and the head size are set to 8×8 and 6. While, the major difference between the two models lies in their respective numbers of channels, we set the channel number of MIA-VSR-small as 120 and set the channel number of MIA-VSR-tiny as 96.

The PSNR values, number of parameters and running FLOPs by different VSR models are presented in Fig.1 and Table 1. In addition to our comparison models in the main paper, two recently proposed efficient VSR models, i.e., FTVSR [8] and TTVSR [6], are also included for reference. Generally, Transformer-based VSR models could achieve better VSR results than CNN-based methods. MIA-VSR outperforms the state-of-the-art CNN-based model BasicVSR++ by a large margin. Furthermore, with less number of parameters and less running FLOPs, our MIA-VSR-tiny model could achieve a better trade-off between computation burden and VSR results over the existing CNN-based models. In comparison with state-of-the-art Transformer-based methods, our MIA-VSR model could achieve better VSR results with much less computational cost. While, our light-weight models MIA-VSR-small and MIA-VSR-tiny also achieved a better trade-off between VSR results and computational cost than the existing light-weight Transformer-based methods; with 45% less number of FLOPs, our MIA-VSR-tiny could improve the TTVSR model by 0.28 dB.

C. Fine-tune MIA-VSR with longer sequences.

For further proving training VSR model with longer sequences can get a better result, we chose the MIA-VSR model which is trained for 450K iterations with 16 frames from the REDS [7] dataset and fine-tuned it for another 150K iterations with 40 frames, named MIA-VSR†.

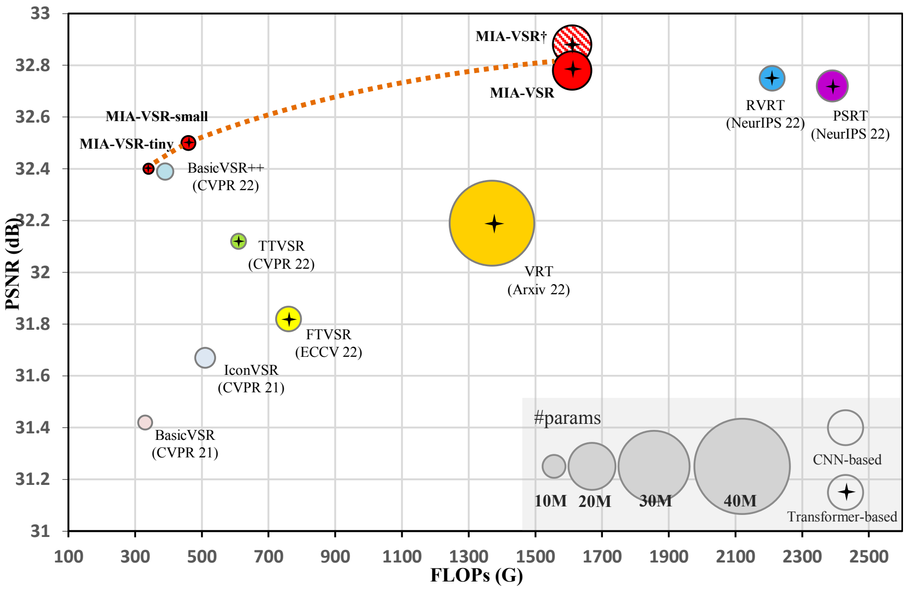

Figure 1. **PSNR(dB) and FLOPs(G) comparison on the REDS4 [7] dataset.** In comparison with the existing video super-resolution methods, our proposed MIA-VSR model, MIA-VSR-small and MIA-VSR-tiny could obtain better trade-offs between VSR results and computational cost. Our fine-tuned model MIA-VSR† outperforms the current state-of-the-art model by more than 0.1 dB with nearly 40% less number of FLOPs. Our light-weight model MIA-VSR-tiny outperforms the recent light-weight Transformer-based VSR model TTVSR[6] by 0.28 dB, with 45% less number of FLOPs. More details can be found in Section B.

Table 1. Quantitative comparison (PSNR/SSIM) on the REDS4 [7], Vimeo90K-T [10] and Vid4 [5] dataset for 4× video super-resolution task. For each group of experiments, we color the best and second-best performance with red and blue, respectively.

Method	Frames	Params	REDS4			Vimeo-90K-T			Vid4		
	REDS/Vimeo	(M)	PSNR	SSIM	FLOPs	PSNR	SSIM	FLOPs	PSNR	SSIM	FLOPs
BasicVSR [1]	15/14	6.3	31.42	0.8909	0.33	37.18	0.9450	0.041	27.24	0.8251	0.134
IconVSR [1]	15/14	8.7	31.67	0.8948	0.51	37.47	0.9476	0.063	27.39	0.8279	0.207
BasicVSR++ [2]	30/14	7.9	32.39	0.9069	0.39	37.79	0.9500	0.049	27.79	0.8400	0.158
FTVSR [8]	40/-	10.8	31.82	0.8960	0.76	-	-	-	-	-	-
TTVSR [6]	50/-	6.8	32.12	0.9021	0.61	-	-	-	-	-	-
MIA-VSR-tiny	80/-	4.8	32.40	0.9176	0.35	-	-	-	-	-	-
MIA-VSR-small	50/-	6.3	32.50	0.9197	0.47	-	-	-	-	-	-
VRT [3]	16/7	35.6	32.19	0.9006	1.37	38.20	0.9530	0.170	27.93	0.8425	0.556
RVRT [4]	30/14	10.8	32.75	0.9113	2.21	38.15	0.9527	0.275	27.99	0.8462	0.913
PSRT-recurrent [9]	16/14	13.4	32.72	0.9106	2.39	38.27	0.9536	0.297	28.07	0.8485	0.970
MIA-VSR	16/14	16.5	32.78	0.9220	1.61	38.22	0.9532	0.204	28.20	0.8507	0.624
MIA-VSR†	40/-	16.5	32.88	0.9241	1.61	-	-	-	-	-	-

MIA-VSR† is fine-tuned with the well-trained MIA-VSR model by 40 frames from the REDS [7] dataset.

The comparison with the state-of-the-art Transformer-based VSR methods (i.e., VRT [3], RVRT [4] and PSRT [9]) can be found in Fig.1 and Table 1. It has a further improvement of the MIA-VSR model trained with 16 frames from the REDS dataset by 0.1dB without adding the number of FLOPs.

D. Visual results

We show more visual comparisons between the existing VSR methods and the proposed VSR Transformer with masked inter&intra-frame attention (MIA). We use 16 frames to train on the REDS dataset and 14 on the Vimeo-90K dataset. Fig.2 and Fig.3 show the visual results. It can be seen that, in addition to its quantization improvement, the proposed method can generate visually pleasing images with sharp edges and fine details, such as horizontal bar patterns of buildings and numbers on license plates. On the contrary, existing methods suffer from texture distortion or loss of detail in these scenes.

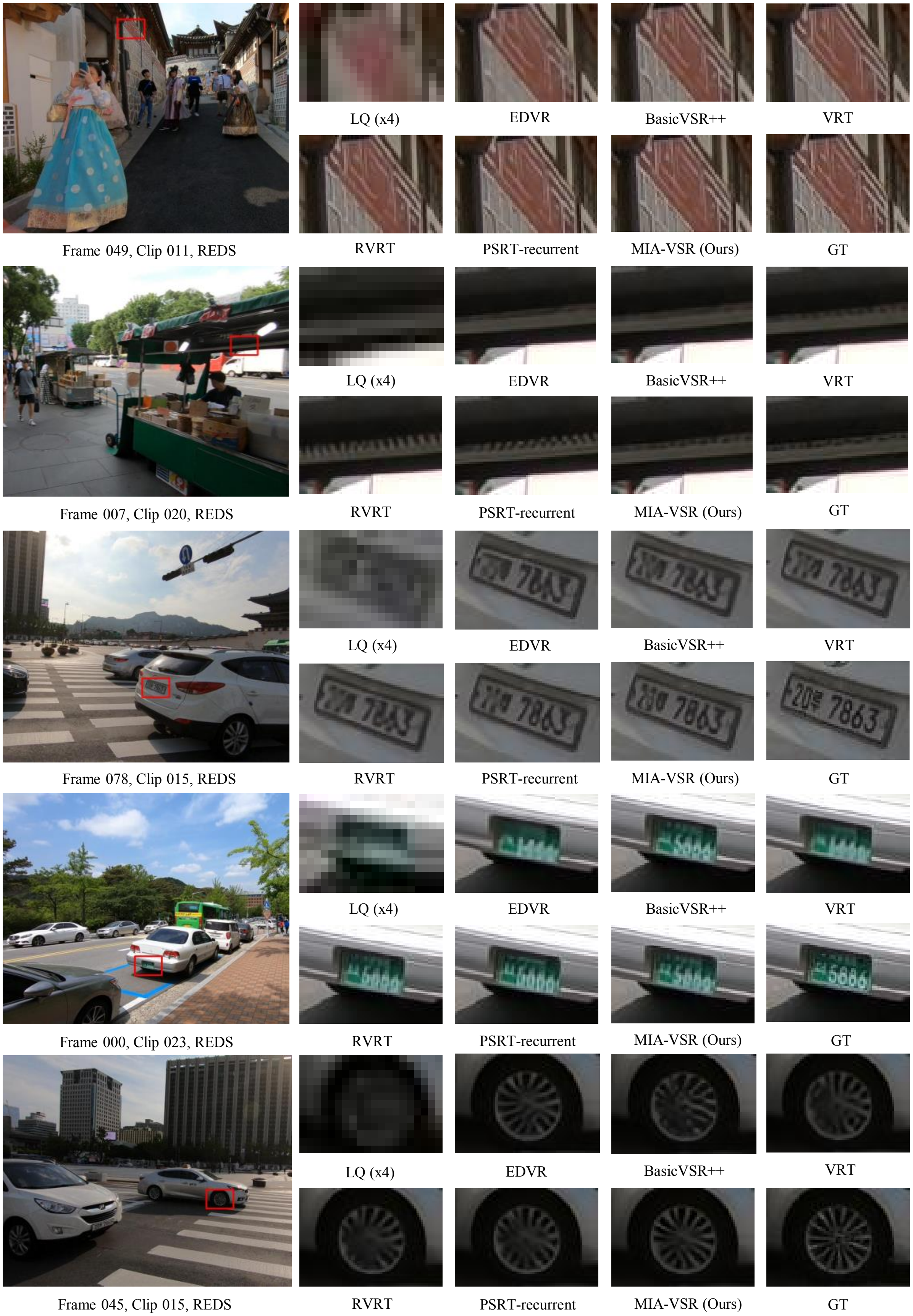

Figure 2. Visual comparison for $4\times$ VSR on REDS4 dataset.

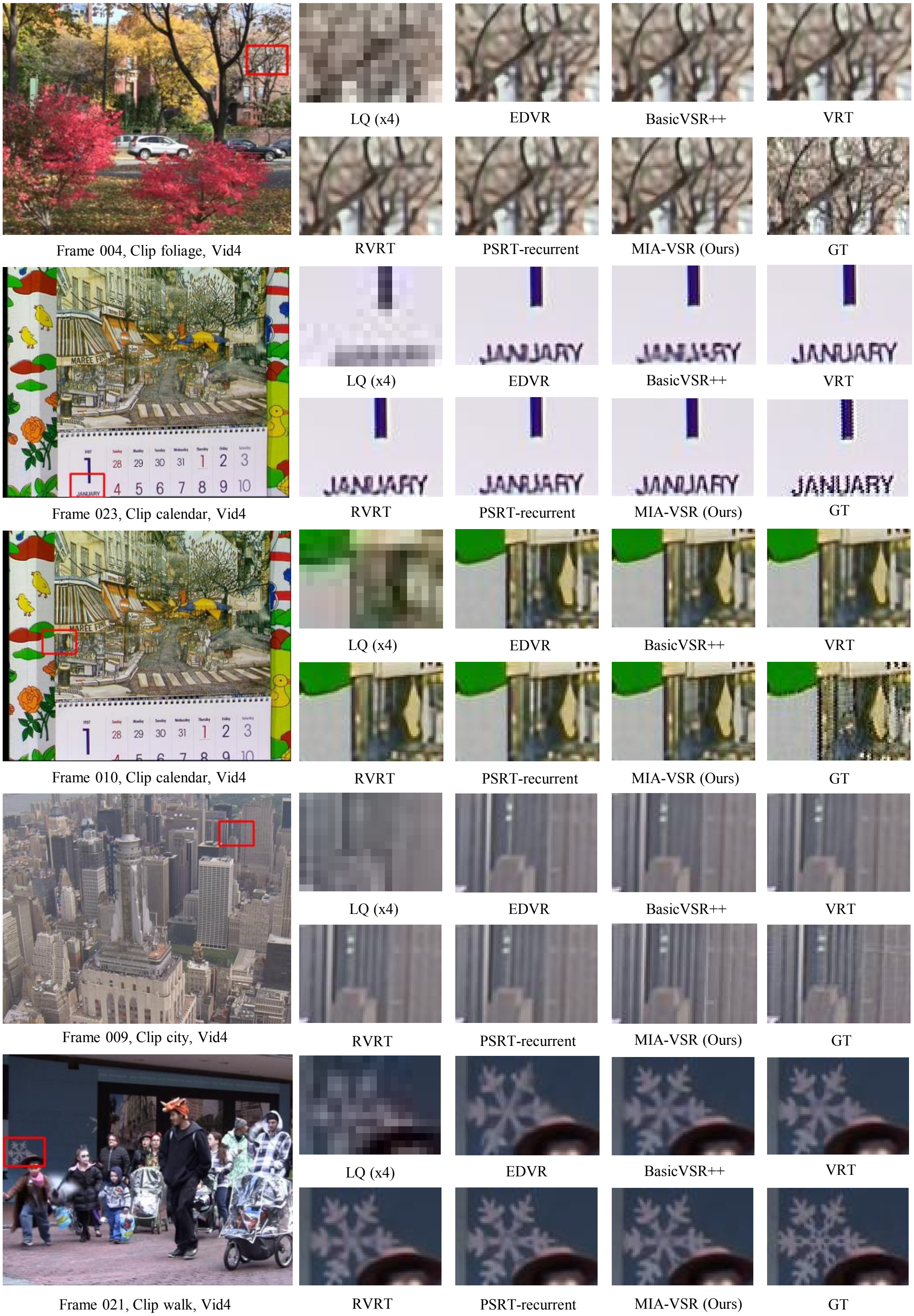

Figure 3. Visual comparison for $4\times$ VSR on Vid4 dataset.

References

[1] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. *arXiv preprint arXiv:2012.02181*, 2020. 2, 3

[2] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5972–5981, 2022. 2, 3

[3] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022. 3

[4] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. *Advances in Neural Information Processing Systems*, 35:378–393, 2022. 3

[5] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):346–360, 2013. 2, 3

[6] Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian. Learning trajectory-aware transformer for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5687–5696, 2022. 2, 3

[7] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 3

[8] Zhongwei Qiu, Huan Yang, Jianlong Fu, and Dongmei Fu. Learning spatiotemporal frequency-transformer for compressed video super-resolution. In *European Conference on Computer Vision*, pages 257–273. Springer, 2022. 2, 3

[9] Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. *arXiv preprint arXiv:2207.08494*, 2022. 2, 3

[10] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8): 1106–1125, 2019. 2, 3