VSRM: A Robust Mamba-Based Framework for Video Super-Resolution

Supplementary Material

1. Architecture

Architecture. In VSRM, two consecutive second-order bidirectional propagation blocks are utilized, following the PSRT [6] and IART [8]. We use pre-trained SPyNet [5] as our flow estimation network. The embedding dimension is set to 120 channels, and each propagation branch consists of 18 Dual Aggregation Mamba Blocks (DAMB) , with shortcut connections introduced every 6 blocks. The initial window size is set to 2×2 for the deformable cross-mamba, and the number of channels is also set to 120 in the DCA alignment module.

We use sinusoidal positional encoding $\gamma(\mathbf{p})$ for deformable cross-mamba module. The positional encoding $\gamma(\mathbf{p}) \in \mathbb{R}^2 \rightarrow \mathbb{R}^{4D}$ is computed by projecting low-dimensional input coordinates $\mathbf{p}$ to a $4D$ dimensional hypersphere.

$$\gamma(\mathbf{p}) = \left[\left[\sin(\omega\mathbf{p}), \cos(\omega\mathbf{p})\right], \ldots, \left[\sin(\omega^{D-1}\mathbf{p}), \cos(\omega^{D-1}\mathbf{p})\right]\right], \tag{1}$$

where ω is the angular speed and D controls the number of frequency bands from ω to ω^{D-1}. A larger D provides higher capacity for encoding higher frequency.

Computational Complexity Comparison. Given a visual sequence $X_{in} \in \mathbb{R}^{1\times T\times H\times W\times C}$, where T is the number of frames in the sequence, C is the number of channels, and H and W represent the height and width of each frame, we will perform a complexity analysis comparing full attention, window attention, and k-direction mamba (SSM in mamba scan by k directions). Notably, k is 2 for the Spatial-to-Temporal block (S2TMB) and 1 for the Temporal-to-Spatial block (T2SMB) because they scan in 2 and 1 directions, respectively. For window attention approach, X_{in} will be divided into $\frac{HW}{hw}$ non-overlapping cubes of size $T \times h \times w$. For mamba, N is a fixed parameter set to 16 by default. Let $M = THW$ be the sequence length. The computational complexity of these approaches can be expressed explicitly as:

$$\Omega(\text{self-attention}) = 4D^2(THW) + 2D(THW)^2 \tag{2}$$

$$\Omega(\text{window-attention}) = 4D^2(THW) + 2ThwD(THW) \tag{3}$$

$$\Omega(\text{k-mamba}) = k \times [N(3THW)(2D) + N^2(2D)(THW)] \tag{4}$$

It demonstrates that self-attention's computational demand scales quadratically with the sequence length M, whereas window attention and k-mamba operations scale linearly. While window attention reduces the quadratic complexity of full attention by limiting attention to small windows, mamba achieves true linear complexity with respect to the sequence length M, offering efficient global modeling without the constraints of fixed window sizes. The computational complexity of window attention depends significantly on the size of the window. If the window size is too small, the model's receptive field is limited, which is harmful for global modeling. Conversely, if the window size is too large, it increases the quadratic computational complexity overhead.

2. Experimental Settings

Datasets. In the video super-resolution (VSR) domain, the REDS [4] and Vimeo-90K [9] datasets serve as standard benchmarks and are widely used. The REDS dataset includes 270 video sequences, each consisting of 100 frames. Following common data-splitting practices, we allocate 266 sequences for training and 4 for testing. Meanwhile, the Vimeo-90K dataset comprises 64,612 training sequences and 7,824 testing sequences. Despite their widespread use, these datasets exhibit distinct motion characteristics. The motion in Vimeo-90K is relatively small. In contrast, the REDS dataset contains more significant motion. In our experiments, we focus on the ×4 VSR task and generate low-resolution (LR) video frames using bicubic interpolation for a fair comparison with other state-of-the-art methods.

Implementation. We use the Adam optimizer [2] along with the Cosine Annealing learning rate schedule [3]. We use a batch size of 8, and the input low-resolution (LR) frames have a patch size of 64×64. DCA and TGFN use 3x3 window/kernel size.

For VSRM on the REDS dataset, we train for 600K iterations using 16 input frames and 300K iterations using 6 input frames. The initial learning rate is set to 2×10^{-4} and gradually reduces to 1×10^{-7} using a cosine decay schedule. The batch size is fixed at 8.

For VSRM training on the Vimeo-90K dataset, we perform 300K iterations with 7 input frames. The initial learning rate is set to 2×10^{-4} and gradually decreases to 1×10^{-7} using a cosine decay schedule. Following [1, 6, 8], we initialize the model with the well-trained model using REDS dataset, and the batch size is maintained at 8.

The test results for the REDS model are evaluated on the REDS4 dataset, while the Vimeo-90K model is tested on Vimeo-90K-T and Vid4.

Ablation Study. For ablation studies on the REDS dataset, the total training iterations are set to 100K, with a learning rate initialized at 2×10^{-4} and subjected to a cosine learning rate decay, reaching 1×10^{-7} at the end of training. The batch size used for these experiments is 2. The embedding dimension is set to 84 channels. All ablation studies are run

on the RTX 3090 GPUs.

3. Limitation Discussion

Limitation of Recurrent Framework. We will discuss the limitation of VSRM and, more generally, the recurrent framework to provide more insights for future works. 1) Due to the purpose of VSRM and other recurrent networks to utilize long-term information, they are typically trained with extended sequences, such as 16 frames. Consequently, compared to sliding-window techniques like EDVR [7], the training duration for recurrent VSR models tends to be longer. 2) Similar to the most recent works, such as PSRT [6] and IART [8], VSRM operates on a bidirectional recurrent framework that demands a considerable amount of memory. In bidirectional recurrent networks, the intermediate features of the entire sequence must be stored, which means the memory requirements grow with the sequence length. Nonetheless, this issue can be mitigated with certain hardware solutions, such as storing the features in the CPU.

Limitation of Alignment Module. A fundamental drawback of the implicit alignment module is the diminished clarity regarding the alignment process. Nevertheless, we contend that it can be substantiated through thorough testing and experimentation.

4. Additional Results

Additional Ablation Study. We conduct an additional ablation study to observe the impact of position encoding in the alignment module. The results are shown in Tab. 1. This indicates that including positional encoding significantly enhances PSNR of 0.22 compared to the basic deformable window cross-mamba mechanism. When positional encodings are solely activated for the reference window PE_R, a substantial decline in PSNR indicates that estimating motion in integer values can lead to poorer results. The model experiences a minor decrease in performance when positional encodings are applied only to the query tensor PE_Q. The above results show the impact of sinusoidal position encoding on the overall model's performance.

PE_Q	PE_R	PSNR (dB)
✗	✗	30.87
✓	✗	29.93
✗	✓	30.96
✓	✓	**31.09**

Table 1. Effective of positional encodings in the alignment module. The positional encoding improves the alignment effectiveness compared to the naive deformable window-based cross-mamba.

Additional Qualitative Results. We offer additional visual comparisons of our method compared to the current SOTA methods on the REDS4 and Vid4 datasets in Fig. 1 and Fig. 2. All current techniques result in blur or distortion of features in the output frames, while our approach retains finer details more effectively and sharply.

References

[1] 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. 1

[2] P Kingma Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014. 1

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[4] 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. 1

[5] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network, 2016. 1

[6] Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. *Advances in Neural Information Processing Systems*, 35:36081–36093, 2022. 1, 2

[7] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2

[8] Kai Xu, Ziwei Yu, Xin Wang, Michael Bi Mi, and Angela Yao. Enhancing video super-resolution via implicit resampling-based alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2546–2555, 2024. 1, 2

[9] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019. 1

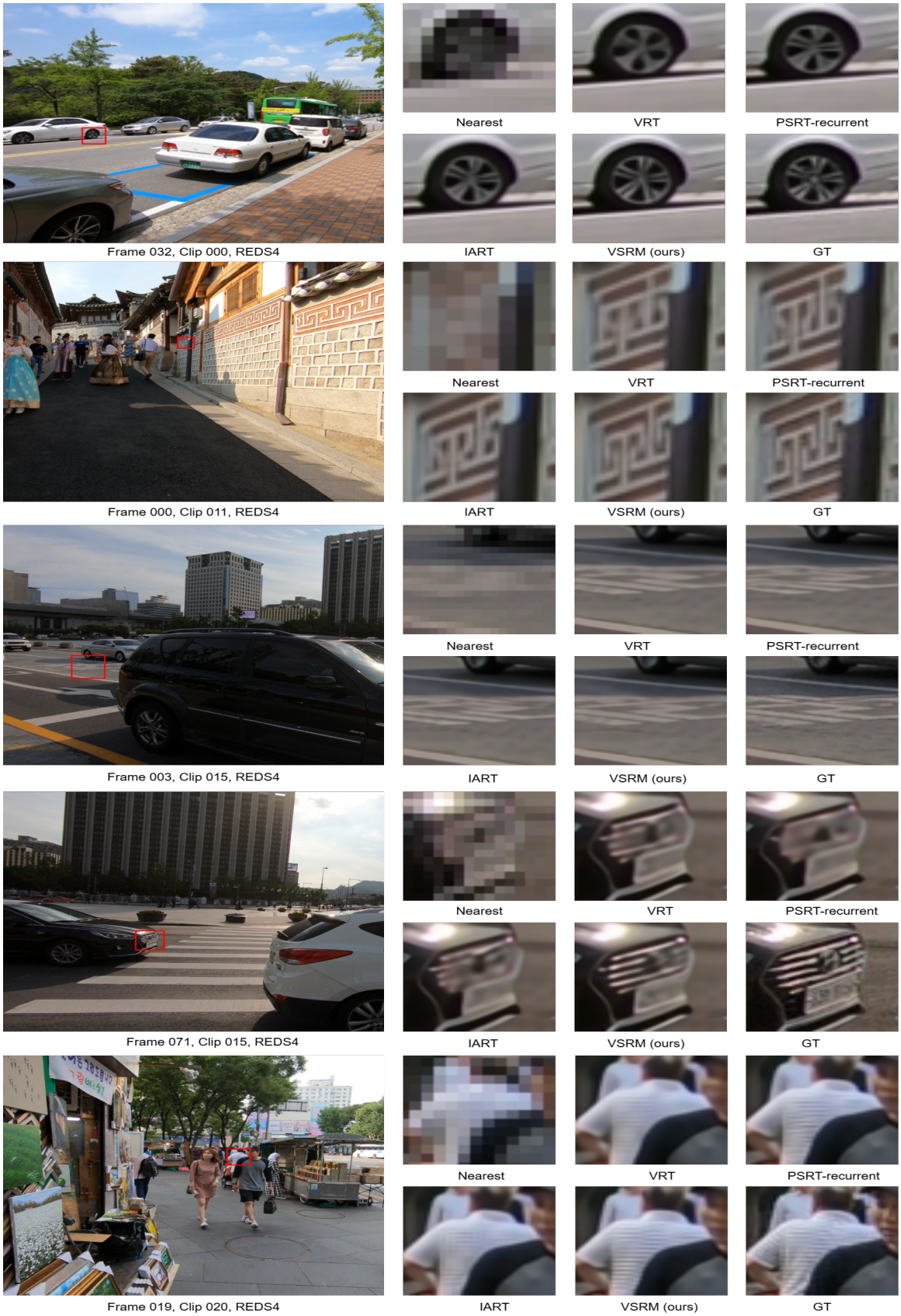

Figure 1. More qualitative comparisons on REDS4 dataset, VSRM shows sharper and more accurate results, revealing finer patterns.

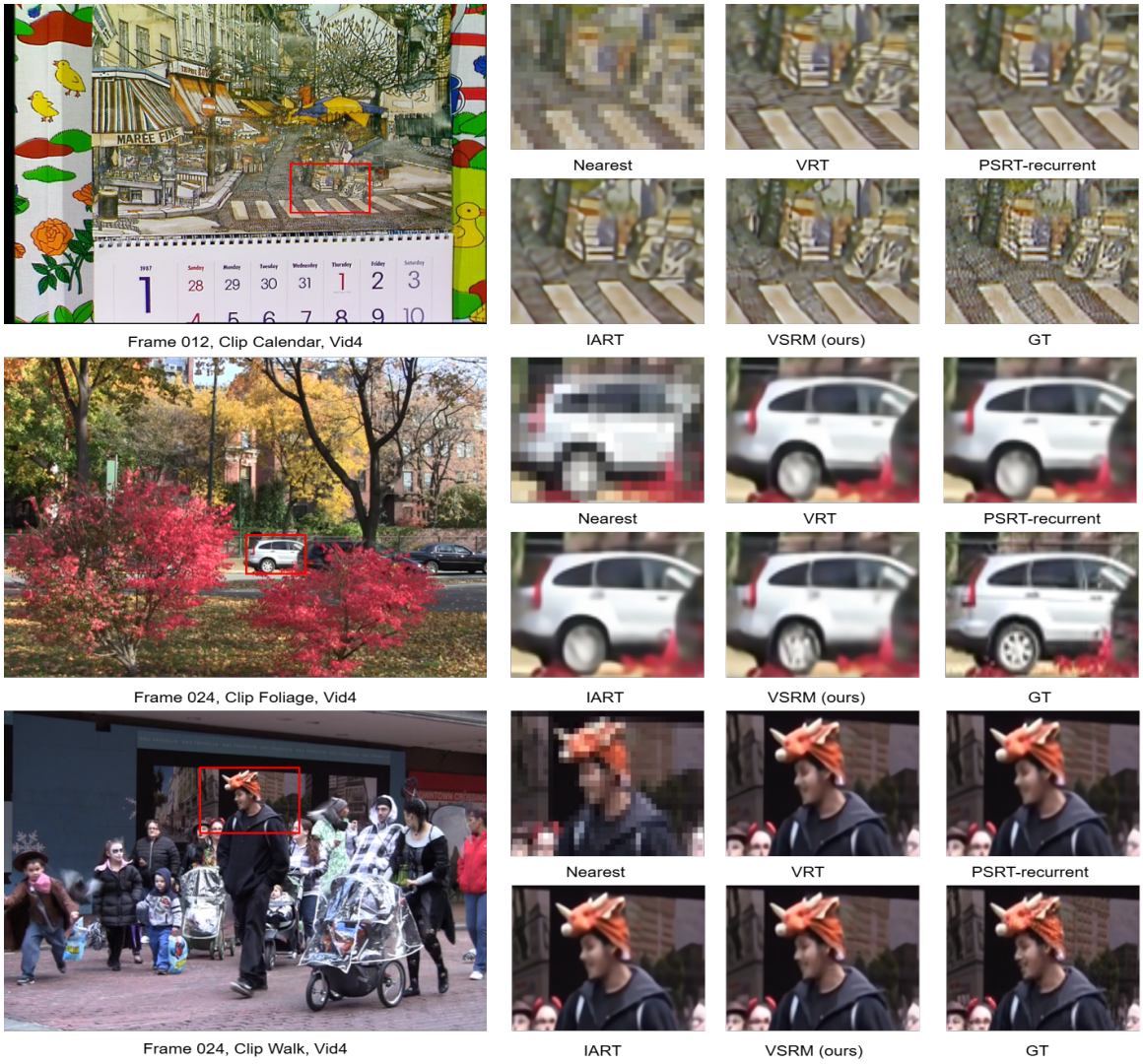

Figure 2. More qualitative comparisons on Vid4 dataset, VSRM shows sharper and more accurate results, revealing finer patterns.