VideoINR: Learning Video Implicit Neural Representation for Continuous Space-Time Super-Resolution

Zeyuan Chen[1] Yinbo Chen[2] Jingwen Liu[2] Xingqian Xu[3,6] Vidit Goel[6]
Zhangyang Wang[5] Humphrey Shi[6,5,3†] Xiaolong Wang[2†]
[1]USTC [2]UC San Diego [3]UIUC [4]UT Austin [5]U of Oregon [6]Picsart AI Research (PAIR)

Abstract

*Videos typically record the streaming and continuous visual data as discrete consecutive frames. Since the storage cost is expensive for videos of high fidelity, most of them are stored in a relatively low resolution and frame rate. Recent works of Space-Time Video Super-Resolution (STVSR) are developed to incorporate temporal interpolation and spatial super-resolution in a unified framework. However, most of them only support a fixed up-sampling scale, which limits their flexibility and applications. In this work, instead of following the discrete representations, we propose Video Implicit Neural Representation (**VideoINR**), and we show its applications for STVSR. The learned implicit neural representation can be decoded to videos of arbitrary spatial resolution and frame rate. We show that VideoINR achieves competitive performances with state-of-the-art STVSR methods on common up-sampling scales and significantly outperforms prior works on continuous and out-of-training-distribution scales. Our project page is at here and code is available at https://github.com/Picsart-AI-Research/VideoINR-Continuous-Space-Time-Super-Resolution.*

1. Introduction

We observe the visual world in the form of streaming and continuous data. However, when we record such data with a video camera in a computer, it is often stored with limited spatial resolutions and temporal frame rates. Because of the high cost on recording and storing large time-scales of video data, oftentimes our computer vision system will need to process low-resolution and low frame rate videos. This introduces challenges in recognition systems such as video object detection [54], and we are still struggling at learning to recognize motion and actions from discrete frames [4, 12]. When presenting the video back to humans (e.g., on a TV), it is essential to visualize it in high resolution and high frame rate for user experience. How to recover the low resolution video back to high resolution in space and time becomes an important problem and the first step for many downstream applications.

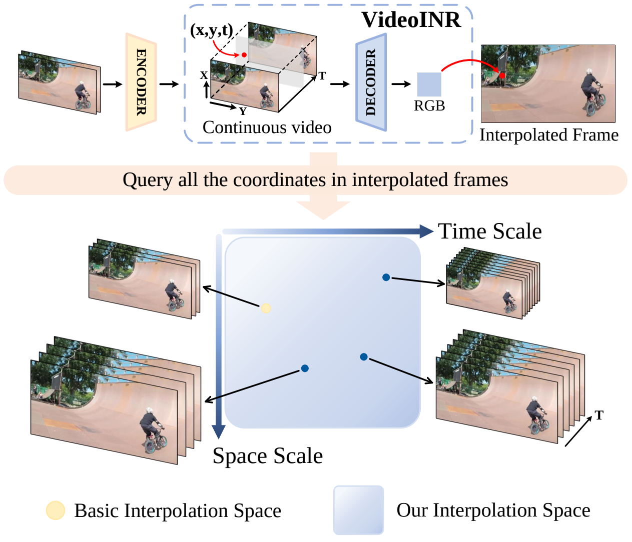

Figure 1. Video Implicit Neural Representation (VideoINR) maps any 3D space-time coordinate to an RGB value. This nature enables extending the latent interpolation space of STVSR from fixed space and time scales to arbitrary frame rate and spatial resolution.

Space-Time Video Super-Resolution (STVSR) approaches [15, 21, 29, 38, 39, 48, 49] are developed to increase the spatial resolution and frame rate at the same time given a low-resolution and low frame rate video as the input. Instead of performing super-resolution in space and time separately in two stages, researchers recently propose to simultaneously perform super-resolution in one stage [15, 21, 48, 49]. Intuitively, the aggregated information in time from multiple frames can reveal missing details for each frame when spatial scaling is applied, and the temporal interpolation can be more smooth and accurate given higher and richer spatial representation. The one-stage end-to-end training has shown to unify the benefits from both

[†] Corresponding authors.

sides. While these results are encouraging, most approaches can only perform super-resolution to a fixed space and time scale ratio.

In this paper, instead of super-resolution in a fixed scale, we propose to learn a continuous video representation, which allows to sample and interpolate the video frames in arbitrary frame rate and spatial resolution at the same time. Our key idea is to learn an implicit neural representation, which is a neural function that takes a space-time coordinate as input, and outputs the corresponding RGB value. Since we can sample the coordinate continuously, the video can be decoded in any spatial resolution and frame rate. Our work is inspired by recent progress on implicit functions for 3D shape representations [10, 13, 14, 27] and image representations with Local Implicit Image Functions (LIIF) using a ConvNet [7]. Different from images, where interpolation in space can be based on the gradients between pixels, pixel gradients across frames with low frame rates are hard to compute. The network will need to understand the motion of the pixels and objects to perform interpolation, which is hard to model by 2D or 3D convolutions alone.

We propose a novel Video Implicit Neural Representation (VideoINR) as a continuous video representation. In the STVSR task, two low-resolution image frames are concatenated and forwarded to an encoder which generates a feature map with spatial dimensions. VideoINR then serves as a continuous video representation over the generated feature map. It first defines a spatial implicit neural representation for a continuous spatial feature domain, from which a high-resolution image feature is sampled according to all query coordinates. Instead of using convolutional operations to perform temporal interpolation, we learn a temporal implicit neural representation to first output a motion flow field given the high-resolution feature and the sampling time as inputs. This flow field will be applied back to warp the high-resolution feature which will be decoded to the target video frame. Since all the operations are differentiable, we can learn the motion in feature level end-to-end without any extra supervision besides the reconstruction error. To summarize, given the input frames, an encoder generates a feature map, which can be then decoded by VideoINR to arbitrary spatial resolution and frame rate.

In our experiments, we demonstrate that VideoINR can not only represent video in arbitrary space and time resolutions on the scales within the training distributions, but also extrapolate to out-of-distribution frame rates and spatial resolutions. Given the learned continuous function, instead of decoding the whole video each time, it allows the flexibility to decode only a certain region and time scale when needed. We conduct experiments with Vid4 [24], GoPro [30] and Adobe240 [42] datasets. We demonstrate that VideoINR achieves competitive performances with state-of-the-art STVSR methods on in-distribution spatial and temporal scales and significantly outperforms other methods on out-of-distribution scales.

We highlight our main contributions as follows:

- We propose a novel Video Implicit Neural Representation as a continuous video representation.
- The proposed approach allows for representing videos in arbitrary space and time resolution efficiently with one single network.
- VideoINR achieves out-of-distribution generalization and outperforms baselines by a large margin.

2. Related Work

Implicit Neural Representation. Implicit neural representations have been demonstrated as compact yet powerful continuous representations for various tasks, including 3D reconstruction [10, 13, 14, 27] and generation [5, 11, 37]. These representations typically represent signals as a neural function that maps coordinates to signed distance [35], occupancy [8, 25], or density and RGB values in a neural radiance field (NeRF [28]). Recent works also show promising results of applying this idea for modeling 2D images [1, 7, 20, 41, 51]. Our continuous video representation is inspired by this rapidly growing field and has specific designs for videos, where a learnable flow can exploit the correspondences in video frames with inductive bias.

Video Frame Interpolation. Video Frame Interpolation (VFI) aims to synthesize unseen frames between the input video frames. Meyer *et al.* [26] proposed a phase-based method where information across levels of a multi-scale pyramid is combined for the synthesis of interpolated frames. Niklaus *et al.* [33, 34] introduced a series of kernel-based VFI algorithms in which they took pixel synthesis for the target frame as local convolution over input frames. Optical flow based VFI methods [2, 18, 31, 32, 50, 52] utilized optical flow prediction networks (*e.g.* PWC-Net [43]) to compute bidirectional flows between input frames, which served as the guidance for new frame synthesis. Additional information including occlusion masks [18, 52], depth maps [2], and cycle consistency [36] were also incorporated in the models for better performances.

Video Super-Resolution. Video Super-Resolution (VSR) aims at increasing the spatial resolutions of low-resolution videos. Earlier approaches [3, 44, 52] were typically built on the sliding-window framework, where they predicted optical flows between input frames and performed spatial warping for explicit feature alignment. Later on, implicit alignment started a new trend in this task [6, 17, 19, 45, 46]. For instance, TDAN [45] adopts deformable convolutions (DCNs) [9, 53] to align different input frames at feature levels. EDVR [46] further extends DCNs to a multi-scale fashion for more accurate alignment. Kelvin *et al.* introduced BasicVSR [6], in which they analyzed basic components

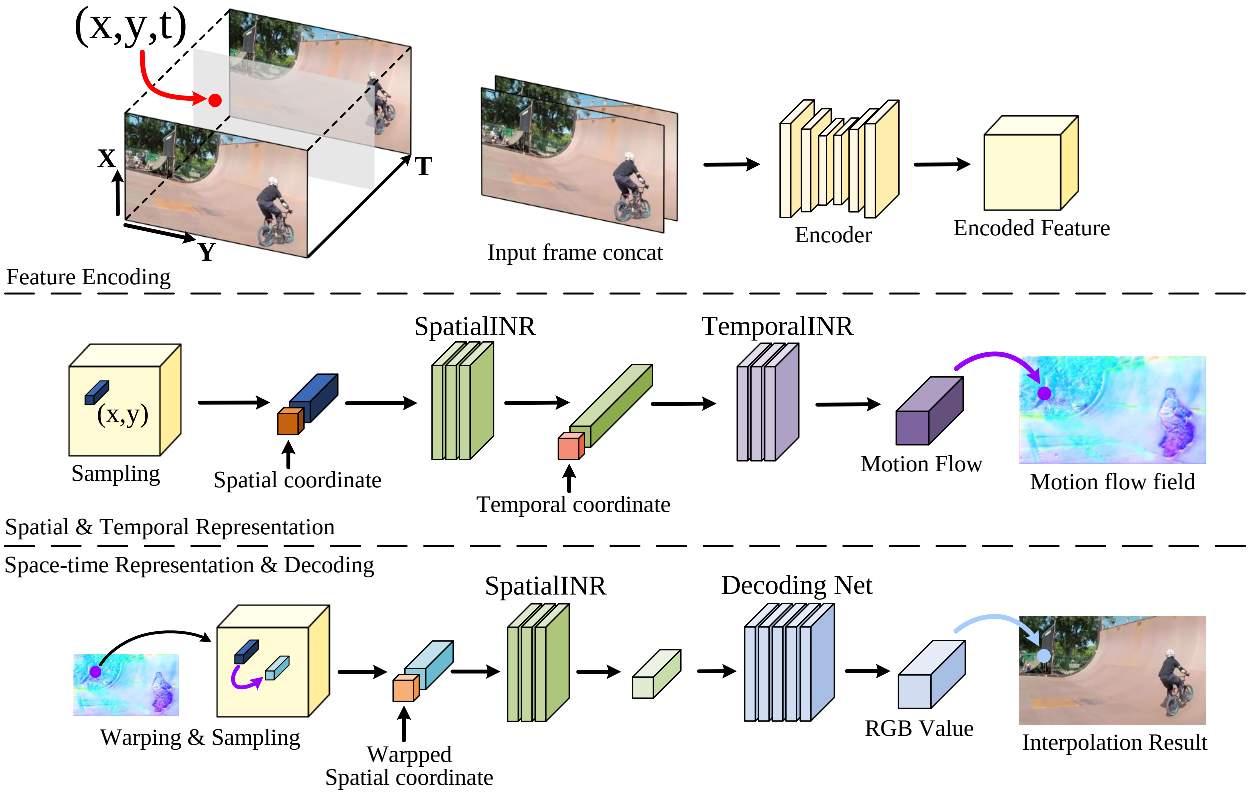

Figure 2. **An overview of our Video Implicit Neural Representation (VideoINR).** Two input frames are concatenated and encoded as a discrete feature map. Based on the feature, the spatial and temporal implicit neural representations decode a 3D space-time coordinate to a motion flow vector. We then sample a new feature vector by warping according to the motion flow, and decode it as the RGB prediction of the query coordinate. We omit the multi-scale feature aggregation part in this figure.

for VSR models and suggested a bidirectional propagation scheme to maximize the gathered information from input.

Space-time Video Super-Resolution The target of Space-Time Video Super-Resolution (STVSR) is to simultaneously increase the spatial and temporal resolutions of the given low-resolution low frame rate videos. Shechtman *et al*. [39] tackled this problem by combining information from multiple input video sequences and applying a directional space-time regularization. Mudenagudi *et al*. [29] proposed a unified framework for STVSR in which videos are modeled as Markov random fields, and the maximum a posteriori estimates are taken as final solutions. Shahar *et al*. [38] introduced an effective space-time patch recurrence prior for STVSR. Recently, with the advances in deep learning, researchers started to employ powerful convolutional neural networks to address the task [15, 21, 48, 49]. Xiang *et al*. [48] proposed a unified neural network for synthesizing the feature of the missing frame and used a deformable ConvLSTM to align and aggregate extracted temporal information for reconstruction. STARNet [15] leveraged mutually informative relationships between time and space with the assistance of additional optical flow inputs. TMNet [49] proposed a temporal modulation block to modulate deformable convolution kernels for supporting frame interpolation at arbitrary time instances. All these STVSR methods are designed to perform super-resolution on a specific up-sampling space scale defined before training, and some of them [15, 48] can only infer intermediate frames at pre-defined times. Therefore, the application scopes of these methods are limited. VideoINR serves as a continuous video representation that supports frame interpolation at arbitrary spatial resolution and frame rate. VideoINR is more flexible during the application and can be employed in more circumstances, such as non-uniform interpolation and video zoom-in in local regions.

3. Video Implicit Neural Representation

Given a video with limited spatial resolution and frame rate, our goal is to find a continuous representation for the video. The representation interprets arbitrary space-time coordinate (x_s, x_t) into RGB values. To this end, we introduce Video Implicit Neural Representation (VideoINR), which enables continuous space-time super-resolution. It is parameterized by multi-layer perceptrons (MLPs) and takes the form

$$s = f(x_s, x_t), \tag{1}$$

where f is the proposed video representation defined by the encoded feature and network parameters. x_s is the 2D spatial coordinate, x_t is the temporal coordinate, and s is the predicted RGB value. For learning such implicit neural representation, we propose to decouple space and time and learn a continuous representation for each of them.

Figure 2 illustrates an overview of our model. Given a space-time coordinate (x_s, x_t) and the feature extracted from input frames by an encoder, the Spatial Implicit Neural Representation (SpatialINR) decodes the spatial coordinate x_s and output a corresponding feature vector (Sec. 3.1). The feature is then forwarded to the Temporal Implicit Neural Representation (TemporalINR) for the motion flow at the query coordinate (Sec. 3.2). The flow is applied back to warp the continuous feature defined by SpatialINR for a new feature vector (Sec. 3.3) which is finally decoded to the target RGB value (Sec. 3.4).

3.1. Continuous Spatial Representation

Inspired by LIIF [7], we learn a Spatial Implicit Neural Representation (SpatialINR) that defines a continuous 2D feature domain by the discrete encoded feature map. This continuous domain decodes arbitrary 2D spatial coordinate into a corresponding feature vector. Specifically, the feature vectors generated by the encoder are evenly distributed in the 2D space. We sample the feature vector (the dark blue cuboid in Fig 2) nearest to the queried spatial coordinate x_s, concatenate it with the relative position information between query coordinate and feature vector, and input them into the a function f_s to output the continuous feature at x_s (the green cuboid in Fig 2). This process could be expressed as

$$\mathcal{F}_s(x_s) = f_s(z^*, x_s - v^*), \tag{2}$$

where $\mathcal{F}_s$ is the continuous feature domain defined by SpatialINR, z^* is the feature vector nearest to the query coordinate x_s and v^* is the spatial coordinate of the feature vector z^*.

The main difference between LIIF and SpatialINR is that LIIF is proposed for continuous image representation, while SpatialINR defines a continuous feature domain, which is supposed to be further utilized for modeling temporal information in videos.

3.2. Continuous Temporal Representation

The proposed SpatialINR defines a new continuous feature domain in 2D space. Our next step is to learn the continuous Temporal Implicit Neural Representation (TemporalINR) and extend the feature domain from 2D space to 3D space and time, which can be achieved by decoding the temporal coordinate x_t. Directly generating the target decoded feature by a network can be fairly difficult, as the network has to learn not only the motion patterns between input frames but also the context information. Instead, we propose to learn a continuous motion flow field for the continuous temporal representation.

Particularly, given a space-time coordinate (x_s, x_t) and two consecutive input frames I_0 and I_1, TemporalINR maps the coordinate to a motion flow

$$\mathcal{M}(x_s, x_t) = f_t(x_s, x_t, I_0, I_1), \tag{3}$$

where $\mathcal{M}$ is the continuous motion flow field and f_t is the function for TemporalINR. Benefiting from the 2D continuous feature domain provided by SpatialINR, we could replace I_0, I_1, and x_s by the continuous feature at x_s. Thus the equation can be written as

$$\mathcal{M}(x_s, x_t) = f_t(x_t, \mathcal{F}_s(x_s)), \tag{4}$$

where $\mathcal{F}_s(x_s)$ is the feature domain defined in Eq 2.

3.3. Space-Time Continuous Representation

With two continuous representations for space and time, we aim at combining them into a unified space-time continuous representation for videos. Starting from a space-time coordinate (x_s, x_t), we first use SpatialINR to predict the continuous feature at x_s. TemporalINR is then utilized for generating the motion flow of the query coordinate. Based on these outputs, we obtain the space-time feature by warping the continuous feature domain. The warped feature at x_s corresponds to the continuous feature at x'_s. The relationship between two coordinates can be written as

$$x'_s = x_s + \mathcal{M}(x_s, x_t), \tag{5}$$

where $\mathcal{M}(x_s, x_t)$ is the motion flow vector at (x_s, x_t).

We query this new spatial coordinate in the continuous 2D feature domain and obtain a new feature vector (the light green cuboid in Fig 2), which is treated as the feature of our continuous space-time representation at coordinate (x_s, x_t). Accordingly, the continuous space-time feature $\mathcal{F}_{st}$ can be formulated as

$$\mathcal{F}_{st}(x_s, x_t) = \mathcal{F}_s(x'_s) = \mathcal{F}_s(x_s + \mathcal{M}(x_s, x_t)), \tag{6}$$

In practice, we generate two independent flows for the motion flow field, and concatenate corresponding warped features. Intuitively, TemporalINR may implicitly learn bidirectional correspondences between the target frame and input frames, without explicit supervision.

3.4. Feature Decoding

Based on the continuous space-time representation, we can get the feature corresponding to any space-time coordinate. The final step is to decode the feature as an RGB value. A straightforward design is to take the obtained space-time feature for decoding directly. However, due to the MLP-based network architecture, the RGB value of every predicted pixel depends on a single feature vector, leading to a limited size of the network receptive field. To alleviate the negative impact of this disadvantage, we enrich the input information of the decoding network by aggregating features of different scales. In detail, we incorporate the encoded feature as well as two input frames for decoding. Since these additional features are typically of low-resolution compared with the target resolution, we sample feature vectors corresponding to the query coordinate by bilinear interpolation. All features are then combined together for predicting the RGB output.

3.5. Frame synthesis

From Section 3.1 to 3.4, we focus on predicting the RGB value at a specific coordinate. To synthesize an entire frame, we need to query coordinates of all pixels of it. Given these coordinates, we can convert the continuous feature from SpatialINR into a high-resolution feature map. We can also generate a whole motion flow field for the latent high-resolution interpolated frame. Therefore, we do not have to forward SpatialINR twice before and after warping as in the situation of one input coordinate. Instead, we directly warp the whole high-resolution feature map based on the motion flow and input the warped feature into the decoding network to synthesize the target frame at one time.

4. Experiments

4.1. Experimental Setup

Dataset. We use Adobe240 dataset [42] as the training set, which includes 133 videos in 720P taken by hand-held cameras. We follow [49] to split these videos into the train, validation, and test subsets with 100, 16, and 17 videos. All videos are converted into image sequences for training and testing. Each sequence contains approximately 3000 frames which are treated as high-resolution frames in training. The low-resolution counterparts are then generated by imresize function in Matlab with the default setting of bicubic interpolation. We use a sliding window to select frames from the image sequences for training. The length of the sliding window is set to 9. We take the 1^{st} and 9^{th} frames as network inputs. The 2^{nd} to 7^{th} frames serve as ground-truth frames, and we randomly select three of them as the supervision of our network in every iteration. VideoINR is trained by two stages. In the first stage, we fixed the down-sampling space scale to ×4. In the second stage, we randomly sample scales in a uniform distribution $\mathcal{U}(1, 4)$. We provide more discussion about this two-stage training strategy in Section 4.3.

Datasets including Vid4 [24], Adobe240 [42], and GoPro [30] are used for evaluation. On Vid4, we only conduct experiments on single frame interpolation of STVSR. For Adobe240 and GoPro, we evaluate on their test set. The image sequences extracted from videos in the datasets are split into groups of 9-frame video clips. We feed the 1^{st} and 9^{th} frames down-sampled by scale ×4 in each clip into models to generate 9 high-resolution frames from 1^{st} to 9^{th}. We separately evaluate the average metrics of the *center* frames (*i.e.* the 1^{st}, 4^{th}, 9^{th} frames) and all 9 output frames. They are denoted as -*Center* and -*Average* in Table 1.

Implementation details. We use Adam optimizer [22] with $\beta_1 = 0.9$ and β_2=0.999. The learning rate is initialized as 1×10^{-4} and is decayed to 1×10^{-7} with a cosine annealing for every 150,000 iterations. The model is trained in a total of 600,000 iterations with batch size 24. The first training stage includes 450,000 iterations while the second stage includes 150,000 iterations. The input frames in one batch are down-sampled by the same space scale and randomly cropped into patches with size 32×32. We perform data augmentation by randomly rotating 90°, 180° and 270°, and horizontal-flipping. We use Zooming SlowMo [48] as the encoder. For the two functions incorporated in continuous space and time representations, we utilize two 3-layer SIRENs [40] with hidden dimensions of $64, 64, 256$. For the decoding network, we employ a 4-layer SIREN with hidden dimensions of $64, 64, 256, 256$. As suggested in [48,49], we select the Charbonnier loss function for optimization.

Evaluation. Peak-Signal-to-Noise Ratio (PSNR) and Structual Similarity Index (SSIM) [47] are employed to evaluate model performances. We also compare the model size and inference time to measure the efficiency of models.

4.2. Comparison to State-of-the-arts

We compare VideoINR with state-of-the-art two-stage and one-stage STVSR methods. For two-stage methods, we employ SuperSloMo [18], QVI [50], and DAIN [2] for video frame interpolation (VFI); Bicubic Interpolation, EDVR [46], and BasicVSR [6] for video super-resolution (VSR). For one-stage methods, we compare VideoINR with recently developed Zooming SlowMo [48] and TMNet [49]. To perform fair comparisons, we train the three VFI methods and Zooming SlowMo from scratch on Adobe240 dataset. For TMNet, as mentioned in the original paper that a two-stage training scheme is needed for convergence, we pre-train the model on Vimeo90K [52] dataset and fine-tune it on Adobe240 dataset [42]. Therefore, TMNet is trained on more data compared with other methods, which may lead to some advantages in the comparison. To compare with Zooming SlowMo that only supports fixed frame interpolation, we train a new version of VideoINR named VideoINR-$fixed$ of which the interpolation time is fixed to 0.5.

Quantitative results. We present in-distribution quantitative comparisons between VideoINR and other STVSR methods in Table 1. On single frame interpolation of

Table 1. **Quantitative comparison on benchmark datasets** including Vid4 [24], GoPro [30] and Adobe240 [42]. The best three results are highlighted in red, blue, and **bold**. We omit the results of Zooming SlowMo and VideoINR-*Fixed* on GoPro-*Average* and Adobe240-*Average* as the two models are trained for synthesizing frames only at fixed times.

VFI Method	SR Method	Vid4 PSNR	 SSIM	GoPro-*Center* PSNR	 SSIM	GoPro-*Average* PSNR	 SSIM	Adobe-*Center* PSNR	 SSIM	Adobe-*Average* PSNR	 SSIM	Parameters (Million)
SuperSloMo [18]	Bicubic	22.42	0.5645	27.04	0.7937	26.06	0.7720	26.09	0.7435	25.29	0.7279	19.8
SuperSloMo [18]	EDVR [46]	23.01	0.6136	28.24	0.8322	26.30	0.7960	27.25	0.7972	25.95	0.7682	19.8+20.7
SuperSloMo [18]	BasicVSR [6]	23.17	0.6159	28.23	0.8308	26.36	**0.7977**	27.28	0.7961	25.94	0.7679	19.8+6.3
QVI [18]	Bicubic	22.11	0.5498	26.50	0.7791	25.41	0.7554	25.57	0.7324	24.72	0.7114	29.2
QVI [18]	EDVR [46]	23.60	0.6471	27.43	0.8081	25.55	0.7739	26.40	0.7692	25.09	0.7406	29.2+20.7
QVI [18]	BasicVSR [6]	23.15	0.6428	27.44	0.8070	26.27	0.7955	26.43	0.7682	25.20	0.7421	29.2+6.3
DAIN [2]	Bicubic	22.57	0.5732	26.92	0.7911	26.11	0.7740	26.01	0.7461	25.40	0.7321	24.0
DAIN [2]	EDVR [46]	23.48	0.6547	28.01	0.8239	26.37	0.7964	27.06	0.7895	26.01	0.7703	24.0+20.7
DAIN [2]	BasicVSR [6]	23.43	0.6514	28.00	0.8227	**26.46**	0.7966	27.07	0.7890	**26.23**	**0.7725**	24.0+6.3
Zooming SlowMo [48]		**25.72**	**0.7717**	30.69	0.8847	-	-	30.26	0.8821	-	-	11.10
TMNet [49]		25.96	0.7803	30.14	0.8692	28.83	0.8514	29.41	0.8524	28.30	0.8354	**12.26**
VideoINR-*fixed*		25.78	0.7730	30.73	0.8850	-	-	30.21	0.8805	-	-	11.31
VideoINR		25.61	0.7709	**30.26**	**0.8792**	29.41	0.8669	**29.92**	**0.8746**	29.27	0.8651	11.31

Table 2. **Quantitative comparison for out-of-distribution scales** on GoPro dataset. Model performances are evaluated by PSNR and SSIM. Some results of TMNet are bolded as it does not support generalizing to out-of-training-distribution space scales.

Time Scale	Space Scale	SuperSloMo [18] + LIIF [7]	DAIN [2] + LIIF [7]	TMNet [49]	VideoINR
×6	×4	26.70 / 0.7988	26.71 / 0.7998	30.49 / 0.8861	**30.78 / 0.8954**
×6	×6	23.47 / 0.6931	23.36 / 0.6902	-	**25.56 / 0.7671**
×6	×12	21.92 / 0.6495	22.01 / 0.6499	-	**24.02 / 0.6900**
×12	×4	25.07 / 0.7491	25.14 / 0.7497	26.38 / 0.7931	**27.32 / 0.8141**
×12	×6	22.91 / 0.6783	22.92 / 0.6785	-	**24.68 / 0.7358**
×12	×12	21.61 / 0.6457	21.78 / 0.6473	-	**23.70 / 0.6830**
×16	×4	24.42 / 0.7296	24.20 / 0.7244	24.72 / 0.7526	**25.81 / 0.7739**
×16	×6	23.28 / 0.6883	22.80 / 0.6722	-	**23.86 / 0.7123**
×16	×12	21.80 / 0.6481	22.22 / 0.6420	-	**22.88 / 0.6659**

Table 3. **Quantitative comparison of out-of-distribution performance between VideoINR and the baseline Zooming Slomo model [48].** Evaluated on GOPRO dataset. -×A×B refers to A up-sampling space scale and B up-sampling time scale.

Method	GoPro - ×4×2 PSNR	 SSIM	GoPro - ×16×4 PSNR	 SSIM
Zooming Slomo	30.69	0.8847	23.38	0.6708
VideoINR	30.26	0.8792	23.45	0.6710

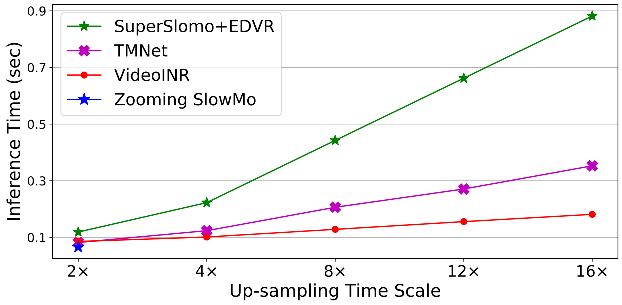

Figure 3. **Inference time of STVSR models on different up-sampling time scales.** Space scale is set to 4. We select the most efficient two-stage method (SuperSlomo + EDVR) as a baseline.

STVSR including Vid4, GoPro-*Center*, and Adobe-*Center*, VideoINR-*Fixed* achieves competitive performance compared with other state-of-the-art models, while the performance of VideoINR slightly suffers. We attribute this observation to the difference of training targets between VideoINR and VideoINR-*Fixed*. The training settings of VideoINR-*Fixed* aim for synthesizing frames at pre-defined times. Therefore, it only learns fixed patterns between input frames instead of learning a continuous representation as VideoINR does, leading to advantages in performances. On Vid4, TMNet performs the best, and we assume this is because TMNet is trained with more data as we noted in Section 4.2. For multiple frame interpolation of STVSR including GoPro-*Average* and Adobe-*Average*, VideoINR achieves the best performance, which indicates that the proposed implicit neural representation provides advances on modeling the temporal information in videos.

In Table 2, we present comparisons of STVSR methods on out-of-distribution space and time scales. For two-stage STVSR methods, we select SuperSloMo and DAIN as

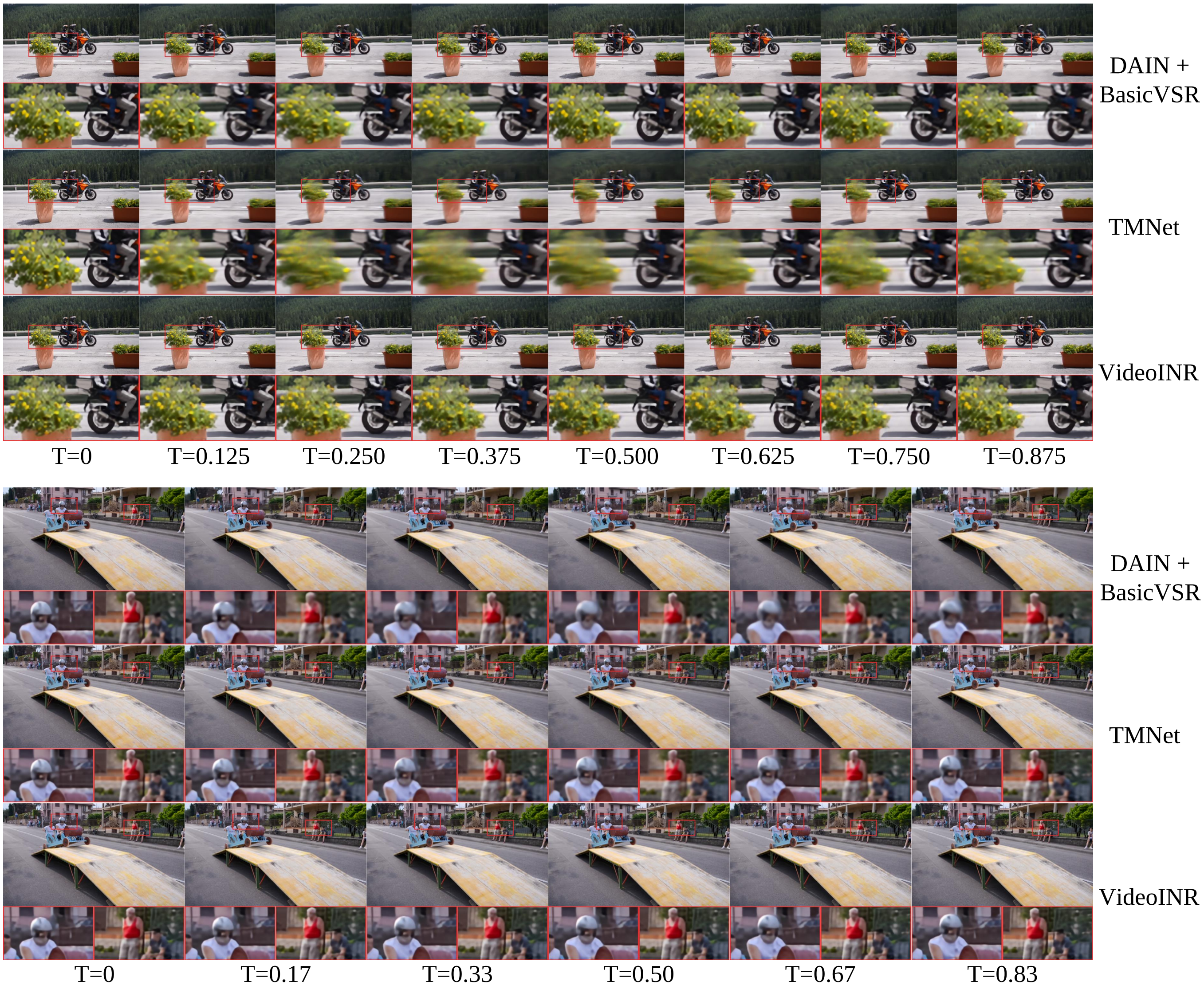

Figure 4. **Qualitative comparisons of different STVSR methods on arbitrary frame interpolation.** The interpolation times of the first example are in the training distribution and the times of the second example are out-of-distribution. Best zoom in for better visualization.

VFI methods, and LIIF as the SR method since it can perform super-resolution on arbitrary up-sampling scales. We also take TMNet into the comparison as it could generalize on time scales. We produce experiments on GoPro [30] dataset. We observe that VideoINR outperforms other methods by a large margin, demonstrating the advantage of our continuous video representation in out-of-distribution generalization. In addition, we further compare VideoINR with Zooming SlowMo (the encoder for VideoINR) in out-of-distribution scales. As Zooming SlowMo only supports interpolating fixed frames, we apply the model twice to achieve out-of-distribution inferences. In Table 3, we observe that while Zooming SlowMo performs slightly better on single frame interpolation ($\times 4 \times 2$), VideoINR achieves better performance in out-of-distribution testing ($\times 16 \times 4$).

We compare the inference time of STVSR methods in Figure 3. We observe that the efficiency of different methods is close at up-sampling time scale $\times 2$, and VideoINR inferences faster than other models on multi-frame interpolation. We attribute this feature to the design of VideoINR, where all the latent frames between two input frames can be directly synthesized by MLPs after encoding.

Qualitative Results We demonstrate a qualitative comparison in Figure 4. We compare VideoINR with two STVSR methods, DAIN + BasicVSR and TMNet. The selected temporal coordinates of the first sample are in the training distribution, while the coordinates of the second sample are out-of-distribution. We find that the performance of DAIN + BasicVSR degrades in out-of-distribution circumstances (see the rider's head in the second sample). TMNet fails to recover objects with large motion between two input frames (see the flowers in the first sample). The performance of

Table 4. **Ablation study on architecture designs of VideoINR.** Evaluated on GOPRO and Adobe240 dataset. -f/m refers to removing flow correspondence and multi-scale feature aggregation. -s refers to decoding both time and space by a single network.

Architecture Design	GoPro-*Center*		GoPro-*Average*		Adobe-*Center*		Adobe-*Average*	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
VideoINR	30.26	0.8792	29.41	0.8669	29.92	0.8746	29.27	0.8651
VideoINR (-f)	29.63	0.8719	28.76	0.8614	29.19	0.8641	28.50	0.8569
VideoINR (-m)	29.99	0.8751	29.28	0.8655	29.68	0.8690	29.04	0.8606
VideoINR (-s)	29.86	0.8741	29.20	0.8654	29.42	0.8678	28.95	0.8613

Table 5. **Ablation study on VideoINR trained with different data settings.** Evaluated on GOPRO-*Average*. -×4 refers to fixing the down-sampling space scale to ×4 throughout the training. -*continuous* refers to training VideoINR by continuous space scales from scratch.

Training Settings	Space ×2		Space ×3		Space ×4		Space ×6		Space ×12	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
VideoINR	29.61	0.8734	29.14	0.8685	29.41	0.8669	25.40	0.7590	24.11	0.6913
VideoINR (-×4)	28.25	0.8490	28.62	0.8626	29.50	0.8696	25.24	0.7567	23.82	0.6857
VideoINR (-*continuous*)	27.46	0.8268	28.35	0.8507	28.82	0.8541	25.10	0.7533	23.62	0.6801

VideoINR is steady across both in-distribution and out-of-distribution temporal coordinates, indicating that learning continuous video representations helps to improve model generalization in STVSR task.

4.3. Ablation Study

Motion Flow Field. Motion flow is one critical component of VideoINR. Previous video interpolation methods [16,18] have already demonstrated that such a learnable flow helps to interpolate frames with sharp edges and clear details. We propose that the motion flow field brings two main advantages. First, the flow field could capture non-local information and temporal contexts of large motions. Second, we explicitly apply spatial warping on features, which works as an inductive bias for the training. In Table 4 between VideoINR and VideoINR (-f), we show that the performance degrades when the motion flow is not incorporated.

VideoINR trained with different data settings. In Table 5, we compare the performances of VideoINR trained on different data settings. As noted before, VideoINR follows a two-stage training strategy: fixed down-sampling space scale for the first stage and continuous space scales sampled from a uniform distribution for the second stage. VideoINR-×4 indicates that the space scale is fixed to ×4 throughout the training of VideoINR. VideoINR-*continuous* represents VideoINR trained with continuous down-sampling space scales from scratch. We find that the performance suffers a significant drop when we train VideoINR only on continuous scales. We hypothesize this is because the network needs to learn spatial and temporal representations at the same time, and it becomes extremely difficult to learn such temporal representation when the scale of spatial features keeps varying. Besides, we observe that training VideoINR with a fixed space scale achieves slightly better performance for that specific scale. However, its generalization performance is competed by VideoINR trained by two stages, which is demonstrated by the comparisons between VideoINR and VideoINR (-×4) on space scales other than ×4.

Other design choices. We provide more ablation studies in Table 4. By comparing VideoINR with VideoINR (-m), we find that the proposed multi-scale feature aggregation contributes to performance improvement. We also try to replace SpatialINR and TemporalINR by a single network, that is, we use one network only for generating the continuous motion flow, and apply spatial warping only on the encoded feature and input frames. The results between VideoINR and VideoINR (-s) indicate that using two functions for representing space and time outperforms only one network for them all.

5. Discussion

Conclusion. In this paper, we present Video Implicit Neural Representation (VideoINR). It can represent videos in arbitrary spatial and temporal resolution, which brings natural advantages for solving Space-Time Video Super-Resolution (STVSR) tasks. Extensive experiments show that VideoINR performs competitively with state-of-the-art STVSR methods on common up-sampling scales and outperforms prior works by a large margin on out-of-distribution scales.

Limitations and Future Work. We observe that there exist few cases for which VideoINR does not perform very well. These cases typically need to handle very large motions, which is still an open challenge for video interpolation.

Acknowledgements. This work was supported, in part, by gifts from Picsart.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 2

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 2, 5, 6, 12

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4778–4787, 2017. 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 2

[6] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4956, 2021. 2, 5, 6, 12

[7] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 2, 4, 6

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 764–773, 2017. 2

[10] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 612–628. Springer, 2020. 2

[11] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. *arXiv preprint arXiv:2104.00670*, 2021. 2

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1

[13] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 2

[14] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019. 2

[15] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2020. 1, 3

[16] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2020. 8

[17] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *European Conference on Computer Vision*, pages 645–660. Springer, 2020. 2

[18] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 2, 5, 6, 8, 12

[19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3224–3232, 2018. 2

[20] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021. 2

[21] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Fisr: deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11278–11286, 2020. 1, 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 12

[24] Ce Liu and Deqing Sun. A bayesian approach to adaptive video super resolution. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 209–216. IEEE, 2011. 2, 5, 6

[25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[26] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1418, 2015. 2

[27] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019. 2

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 2

[29] Uma Mudenagudi, Subhashis Banerjee, and Prem Kumar Kalra. Space-time super-resolution using graph-cut optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):995–1008, 2010. 1, 3

[30] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017. 2, 5, 6, 7

[31] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 2

[32] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 2

[33] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 2

[34] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 2

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[36] Fitsum A Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 892–900, 2019. 2

[37] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020. 2

[38] Oded Shahar, Alon Faktor, and Michal Irani. *Space-time super-resolution from a single video*. IEEE, 2011. 1, 3

[39] Eli Shechtman, Yaron Caspi, and Michal Irani. Increasing space-time resolution in video. In *European Conference on Computer Vision*, pages 753–768. Springer, 2002. 1, 3

[40] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 5

[41] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021. 2

[42] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 2, 5, 6, 12

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 2

[44] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4472–4480, 2017. 2

[45] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3360–3369, 2020. 2

[46] 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, 5, 6

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[48] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3370–3379, 2020. 1, 3, 5, 6, 12

[49] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2021. 1, 3, 5, 6, 12

[50] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances*

in Neural Information Processing Systems, 32:1647–1656, 2019. 2, 5, 12

[51] Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *arXiv preprint arXiv:2103.12716*, 2021. 2

[52] 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, 5, 12

[53] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 2

[54] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 1

A. Implementation Details

All models included in experiments are trained from scratch to perform fair comparisons.

For video frame interpolation methods incorporated in the experiments (*i.e*. Super SloMo [18], QVI [50], and DAIN [2]), we train them on Adobe240 dataset [42]. We keep all the training settings the same as proposed in their original papers, including the optimizer, initial learning rate, learning rate decay strategy, and the number of training epochs. For data settings, 9 consecutive frames are selected from video clips for training in every iteration. Networks take the first and last frames as inputs and generate intermediate 7 frames. We calculate loss between generated frames and the original ground-truth frames. Each video frame is resized to have a shorter spatial dimension of 360, and a random crop of 352×352 is performed.

Zooming SlowMo [48] and TMNet [49] are two STVSR models included in our experiments. Zooming SlowMo only supports fixed frame interpolation, and the interpolation time is set to 0, 0.5, 1 in the original paper. Following their settings, we also train the model from scratch to interpolate the fixed time instances. To ensure that the input video frames of all models are of the same frame rate, we extract 9 consecutive frames from video clips and take the 1^{st} and 9^{th} frames as inputs. We then down-sample the input frames via Bicubic interpolation by a factor of 4 and use the network to predict the high-resolution versions of the 1^{st}, 5^{th} and 9^{th} frames. TMNet supports arbitrary frame interpolation. In its paper, the authors mention that TMNet needs a two-stage training process for convergence, and we follow their suggestions. In the first stage, we pre-train the network on the Vimeo90K dataset [52]. The Vimeo90K dataset consists of 7-frame video sequences. We use the 1^{st}, 3^{rd}, 5^{th}, and 7^{th} frames after down-sampling as the network inputs and predict the high-resolution results of all the 7 frames, which means that the interpolation time is set to 0, 0.5, 1 in this stage. In the second stage, we select 9 consecutive frames from video clips, and the 1^{st} and 9^{th} frames are taken as inputs. After down-sampling, we use the network to generate high-resolution predictions of all 9 frames and calculate the loss value with the original high-resolution frames. TMNet is trained with more data, which may lead to advantages in the experiments.

For the training of VideoINR, we select 9 consecutive frames and down-sample the 1^{st} and 9^{th} frames as model inputs. In each iteration, We randomly select three frames from the 9-frame video sequence and use the network to generate high-resolution predictions at the time instances of the three selected frames.

We keep the training settings unchanged for Zooming SlowMo, TMNet, and VideoINR. All three models are optimized with the Charbonnier loss function [23].

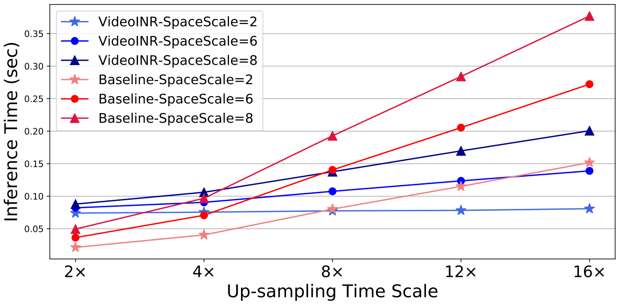

Figure 5. Inference time comparisons on different space scales.

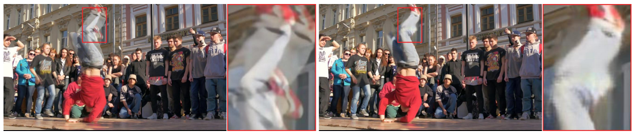

Figure 6. Failure case. Left is the overlay of frames at t=0 and t=1. Right is the interpolated frame at t=0.5

B. Efficiency on Different Scales

To evaluate the efficiency of VideoINR on different up-sampling space scales, we provide more inference time comparisons in Figure 5. We select the two-stage method composed of SuperSlomo and LIIF as the baseline, as it supports arbitrary up-sampling scales on both space and time.

C. Limitations

In some challenging cases, large motion and occlusion result in errors on the motion flow field, leading to blurred results with unclear boundaries. We show a failure cases of VideoINR in Figure 6.

D. Additional Qualitative Results

We provide more qualitative results in Figures 7,8,9,10. We compare VideoINR with two STVSR methods, DAIN [2] + BasicVSR [6] and TMNet [49]. The up-sampling space scale is set to 4 for all examples. In Figure 7, 8, we set the time scale for interpolation to 8, which is in our training distribution. We observe that DAIN + BasicVSR and TMNet tend to generate blurry regions or artifacts. In contrast, the results of VideoINR are consistent and aligned across two input frames, with sharp edges and clear details. In Figure 9, 10, we set the time scale to 12 and 16, which are out of the training distribution. We find that VideoINR well recovers objects with large motion and preserves better textural information compared with other methods. In summary, VideoINR shows the advantages of learning continuous representation for videos, and address the Space-Time Video Super-Resolution task. More visualization results can be found in the provided video.

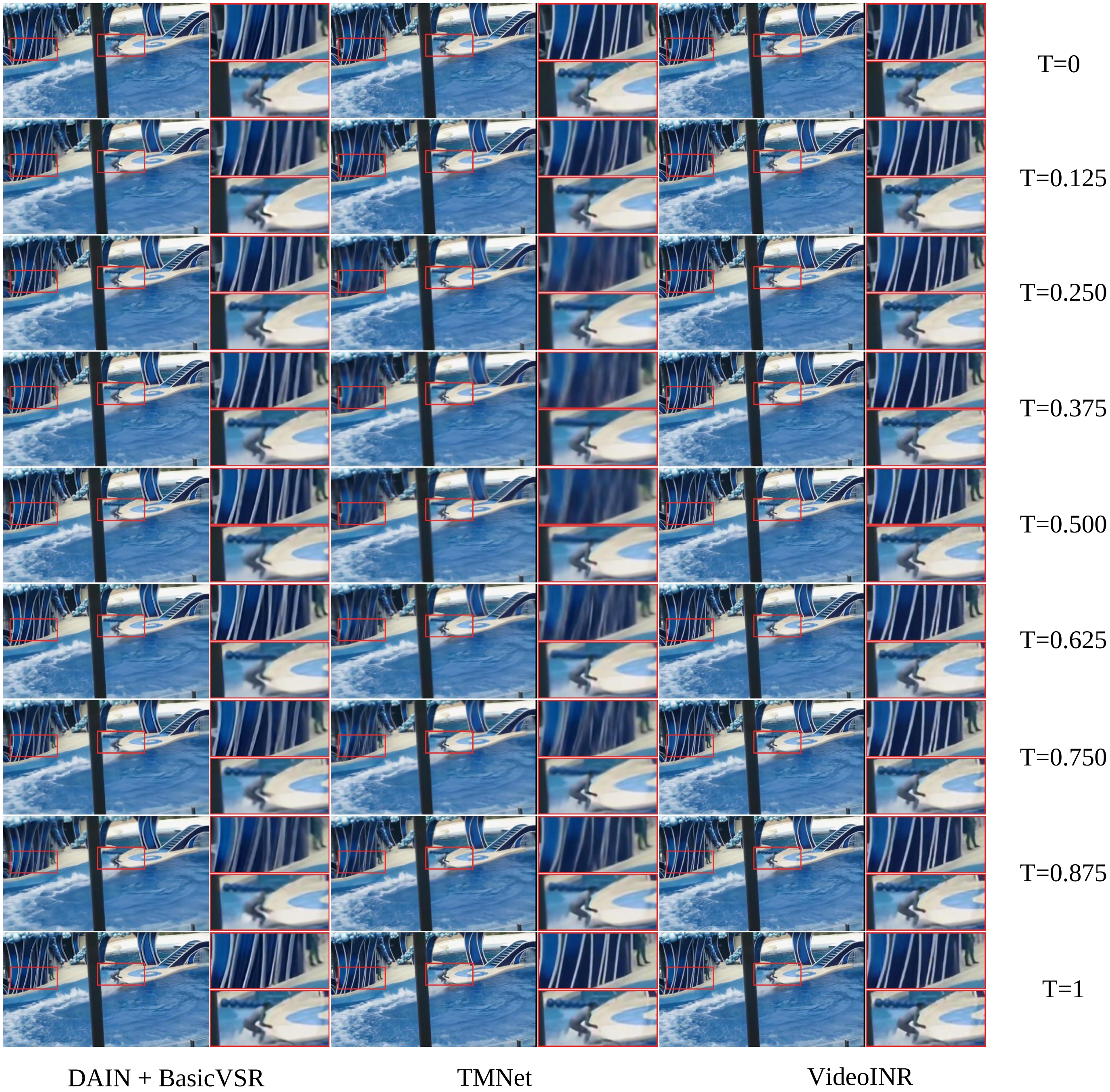

Figure 7. Qualitative comparisons of different STVSR methods on in-distribution time scale. Best zoom in for better visualization.

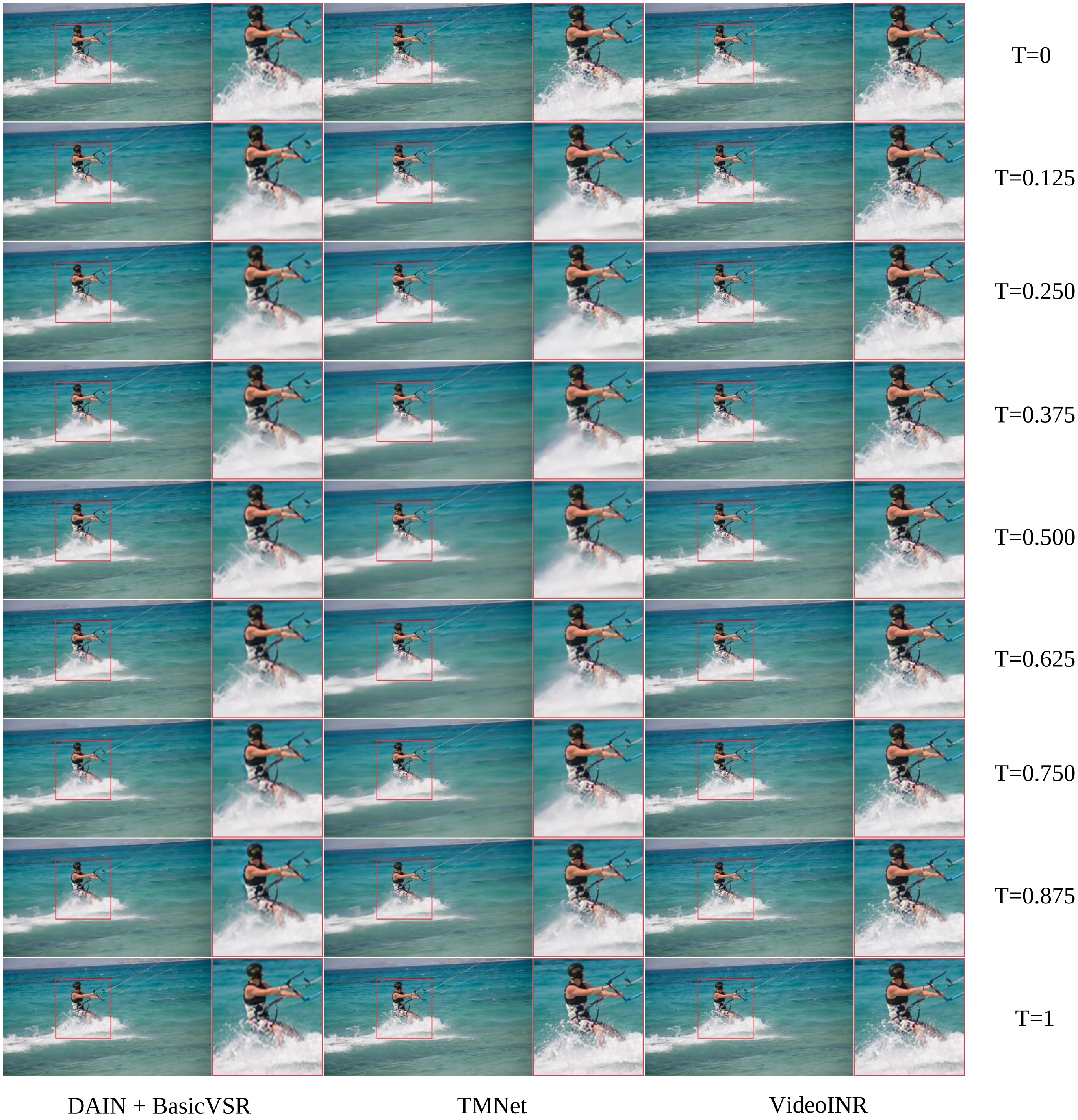

Figure 8. Qualitative comparisons of different STVSR methods on in-distribution time scale. Best zoom in for better visualization.

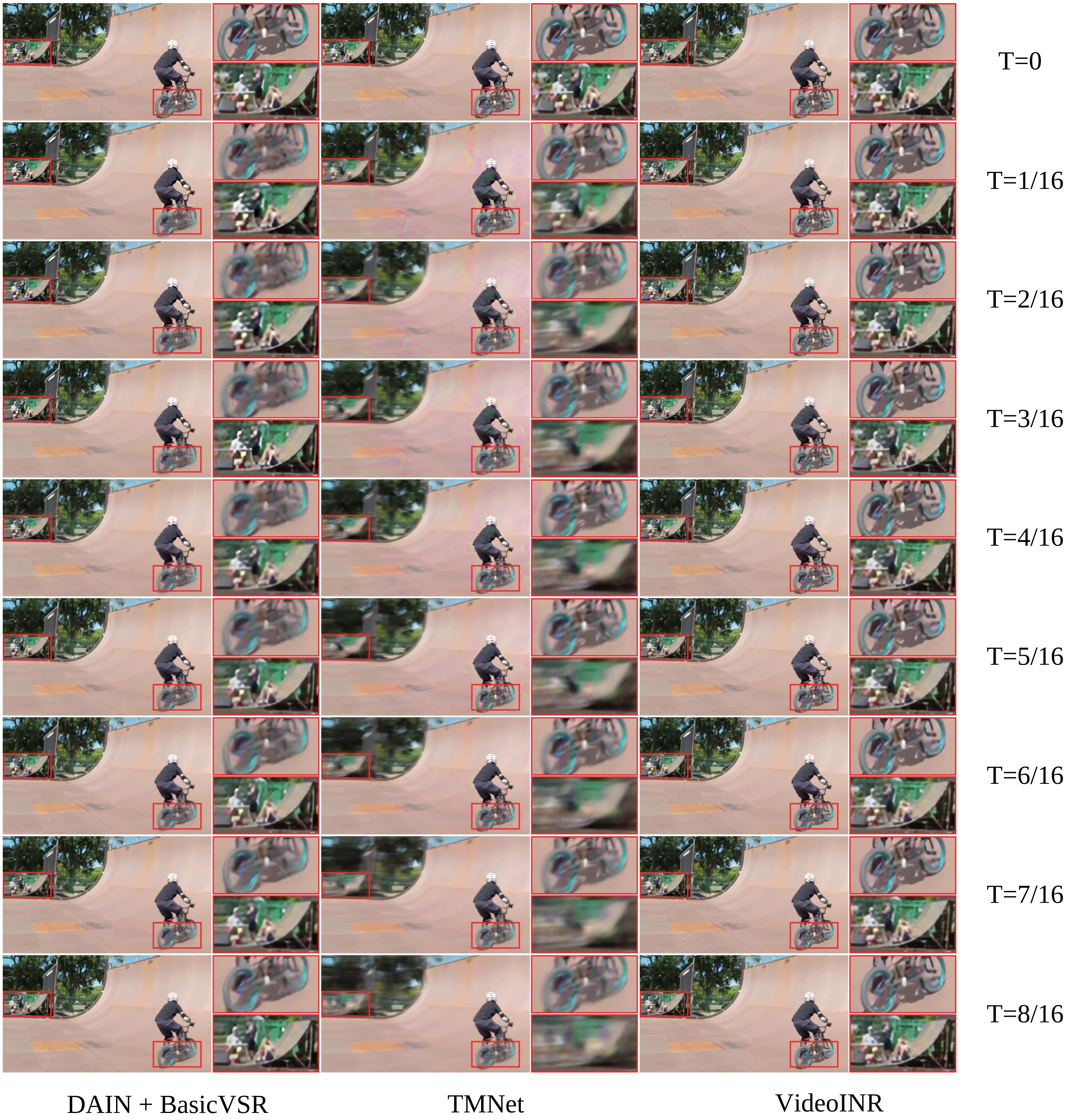

Figure 9. Qualitative comparisons of different STVSR methods on out-of-distribution time scale. Best zoom in for better visualization.

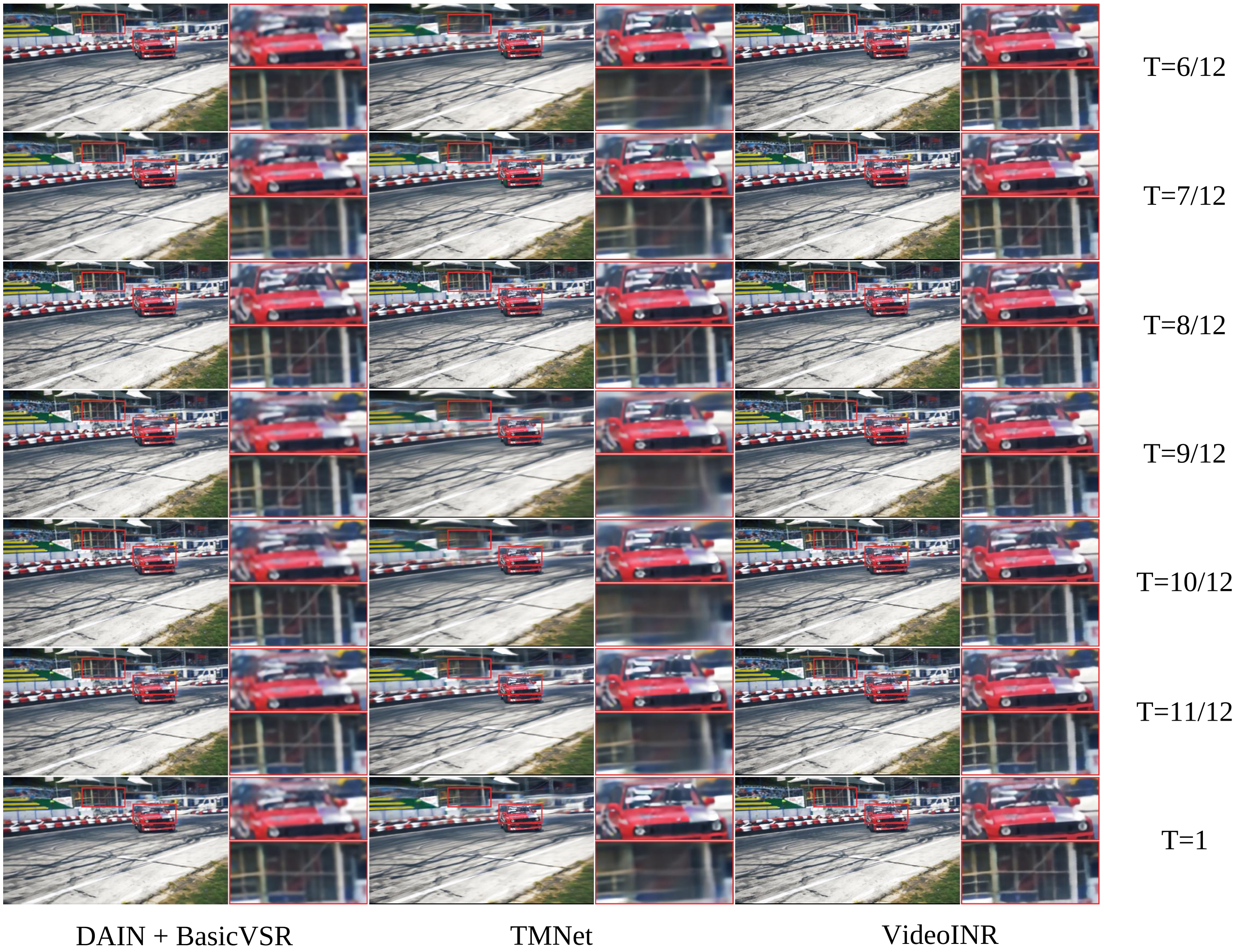

Figure 10. Qualitative comparisons of different STVSR methods on out-of-distribution time scale. Best zoom in for better visualization.