Bi-directional Autoregressive Diffusion for Large Complex Motion Interpolation

Supplementary Material

We have two supplementary files, including this PDF file and **a video demo for dynamic comparisons**. In this PDF file, we provide more experimental results and discussions as follows:

- Visualization for the gain of DINOv3 features towards frame generation;
- More details about the diffusion scheduling matrices S and an ablation study on it;
- Full human evaluation results;
- More implementation details;
- More visual comparisons.

7. Gain of DINOv3 for Frame Generation

We visualize attention modules in Fig. 8. Let us recall the equation of the attention module:

$$A = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{4}$$

We query the first input frame with a selected patch i in an interpolated frame $softmax(\frac{q_i K_0^T}{\sqrt{d_k}})$, where q_i is the query of selected interpolated patch, and K_0 is the key computed from the first input frame. We visualize the results from attention modules with and without DINOv3 features to comapre how they extract patches from the first input frame for interpolation.

Our attention module that utilizes DINOv3 features can extract patches based on both pixel similarities (red boxes) and motions (orange boxes) by DINOv3 patch similarities. Thus, our ARVFI can interpolate frames with correct motions. In contrast, previous attention modules that do not utilize DINOv3 features extract patches only based on distance and pixel similarities (red boxes), cannot produce intermediate frames with accurate motions and thus producing unnatural motions and severe visual degradations.

8. Ablation Study on Diffusion Steps

As introduced in Sec. 3.3, ARVFI implements the bidirectional autoregressive interpolation by applying sequence frames increasing noise with their distance to input frames, controlled by two diffusion scheduling matrices S and S' for DINOv3 features and frame generation, respectively. More specifically, the diffusion scheduling matrices expand each diffusion timestep t into a temporally symmetric list $S_K = (S_K^1, S_K^2, ..., S_K^{\frac{N-1}{2}}, ..., S_K^2, S_K^1)$ where $S_K^1 \leq S_K^2 \leq ... \leq S_K^{\frac{N-1}{2}}$. Let us define the diffusion timestep gap as s and the total diffusion timestep number as T, then we can calculate $S_K^i = \max(\min(T-(K-i)\times s), T), 0)$. As

Figure 8. Visualization of attention maps. We visualize the responses of the first input frame to the a selected patch (yellow circle) in a interpolated frame. Previous attention modules extract patches only based on pixel similarities and distance, thus only response to the white backboard in the red boxes. In contrast, our attention module can extract patches based on motion information by DINOv3 patch similarities, enabling the model generate correct motions rather than simple pixel-wise approximation.

s	s'	LPIPS↓	FID↓	FVD↓	Runtime
100	150	0.251	17.39	105.32	1.451 sec.
200	150	0.246	17.62	101.38	0.937 sec.
1000	150	0.254	19.27	109.37	0.721 sec.
500	100	0.241	17.97	102.38	0.931 sec.
500	500	0.259	19.11	105.38	0.525 sec.
500	1000	0.267	20.31	109.31	0.467 sec.
500	**150**	**0.247**	**17.60**	**101.71**	**0.775 sec.**

Table 4. With different timestep gaps for DINOv3 features estimation (s) and frame generation s', interpolation accuracy changes accordingly. Because DINOv3 features mainly generate motion information by patch similarities, more sampling steps that add details to the generated DINOv3 features do not improve interpolation accuracy. In contrast, more diffusion sampling steps help the frame generation model produce fine-grained details, leading to superior LPIPS metrics. **Our setting** makes a great balance between processing time and interpolation quality.

an example, we assume $N = 5, s = T/3$, then we have:

$$S = \begin{pmatrix} T & T & T & T & T \\ \frac{2T}{3} & T & T & T & \frac{2T}{3} \\ \frac{T}{3} & \frac{2T}{3} & T & \frac{2T}{3} & \frac{T}{3} \\ 0 & \frac{T}{3} & \frac{2T}{3} & \frac{T}{3} & 0 \\ 0 & 0 & \frac{T}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

Correspondingly, the diffusion sampling step is related to the number of rows in the diffusion scheduling matrix S:

$$K = ceil(T/s) + floor(N/2). \tag{6}$$

In Sec. 4.2, we select the timestep gap $s = 500$ for DINOv3 features estimation and $s = 150$ for frame generation, resulting in 5-step and 10-step sampling, respectively. We generate DINOv3 features with fewer steps because we only require the DINOv3 features to provide object motions through patch similarities rather than fine-grained details. To further study the influence of sampling steps to interpolation, we additionally provide an ablation study as shown in Tab. 4, interpolating with different diffusion timesteps for DINOv3 feature generation and frame generation using the same sampled noise. Because DINOv3 features provide object motions by patch similarities, dense sampling that adds more details do not obviously improve interpolation quality, thus we generate DINOv3 features with diffusion timestep gap $s = 500$. In contrast, we utilize a smaller diffusion timestep gap $s' = 150$ for frame generation to ensure generated frames have high pixel fidelity.

9. Full Human Evaluation Results

We provide detailed human evaluation results in Fig. 9, where the proposed ARVFI consistently outperforms all baseline methods, including Wan [33], FCVG [42], LDMVFI [5], FILM [27], GIMM-VFI [9], and AMT [19], in terms of motion and overall interpolation quality.

Non-generative-based methods [9, 19, 27] assume pixels move from the first frame to the second along paths indicated by specific, predetermined motion models, which fail to large complex motions. Recent diffusion-based techniques [5, 33, 42] advance video frame interpolation by sampling intermediate frames from a possible distribution rather than simple regression, as the non-generative methods [9, 19, 27] do. Thus, they interpolate with the overall superior quality. However, all these methods utilize a simple pixel reconstruction training objective on generated frames to jointly estimate intermediate motions and frame appearances. This pixel reconstruction loss enforces the interpolation models to approximate the ground truth pixelwisely, ignoring the accuracy and consistency of generated motions; thus, we observe blurring, ghosting, and appearance transitions in generated frames.

In contrast to current solutions, our ARVFI generates intermediate motions and frame appearances in different data distributions separately. ARVFI first generates the intermediate motions in DINOv3 [32] data domain and then utilizes the patch similarities of DINOv3 features as motion representations to guide frame generation. By generating motions and frame appearances in different domains, we force the diffusion model to generate correct object motions first and then frames conditionally, leading to significant improvements in both generated motions and pixel fidelity, obtaining 85% more selections as the best-performed algorithm.

10. More Implementation Details

Diffusion Scheduler We utilize the default flow matching scheduler with a maximum diffusion timestep of $T = 1000$ in Wan [33] for both training and inference.

Two-Stage Training After the diffusion transformer G_{θ_d} for DINOv3 feature estimation converges, we regenerate the DINOv3 features with the pretrained G_{θ_d}. The diffusion transformer G_{θ_f} then trains with the generated DINOv3 features rather than the ground truth for another 200K iterations.

Feature Alignment We utilize the Wan-Fun-InP-1.3B model [33] to build all our diffusion transformers. We can inject generated DINOv3 features into the frame generation diffusion transformer G_{θ_f} because the DINOv3 features can align with frame embeddings in Wan [33] with several adaptations. A frame embedding in Wan [33] has 1536 channels and matches a $4 \times 3 \times 16 \times 16$ (frame length$\times$channel$\times$height$\times$width) sequence frame patch. The DINOv3 features only a squeeze in space. Each DINOv3 feature has 384 channels and matches a $3 \times 16 \times 16$ sequence frame patch. To align with frame embeddings, we fold four frames of DINOv3 features along the channel dimension, producing DINOv3 embeddings at 1536 channels and match the same $4 \times 3 \times 16 \times 16$ sequence frame patch indicated by the frame embedding.

11. More Visual Comparisons

We provide more intermediate and sequence frame interpolation results in Fig. 10 and Fig. 11, respectively. The results clearly show that the proposed ARVFI can effectively interpolate to challenging motions, resulting in consistent and accurate interpolation results.

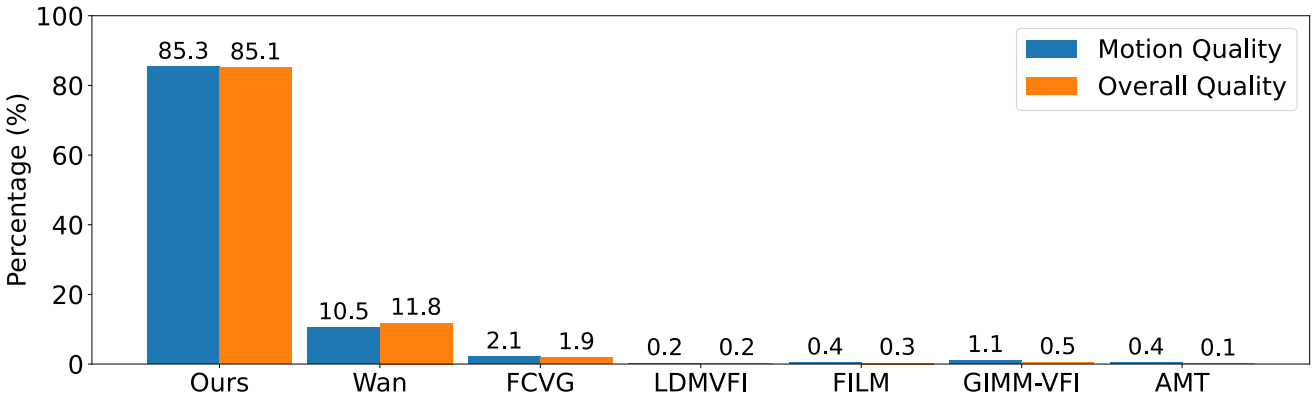

Figure 9. Human evaluation results. Our ARVFI is selected as the best-performed method for large complex motion interpolation in terms of motion and overall interpolation quality.

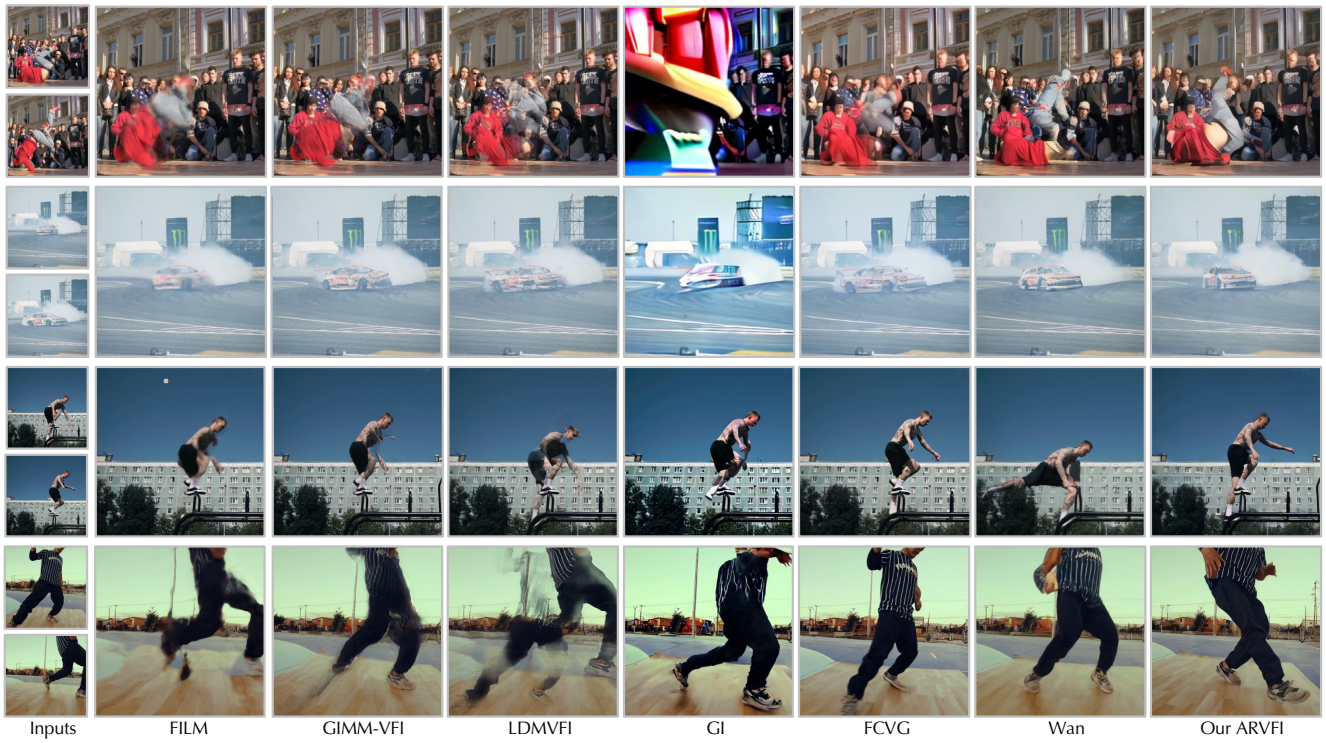

Figure 10. Interpolated middle frame. The first two rows are from the DAVIS-7 [16] dataset, and the last two rows are from our self-collected Pixels dataset. Conventional non-generative interpolation methods [9, 27] cannot interpolate to those challenging motions, resulting in severe artifacts like ghosting effects and fractions. Image diffusion-based interpolation method [5] improves small and simple interpolation, but still cannot solve large, complex motions. Recent video diffusion-based methods [33, 34, 42] jointly generate motions and appearances for all frames through each diffusion sampling, resulting in appearance approximation and unstable interpolation (GI produces undesired contents in the first row). In contrast, with the bidirectional autoregressive interpolation and DINOv3 features as motion representations, the proposed ARVFI can solve large complex motions better and produce more consistent and superior interpolation results.

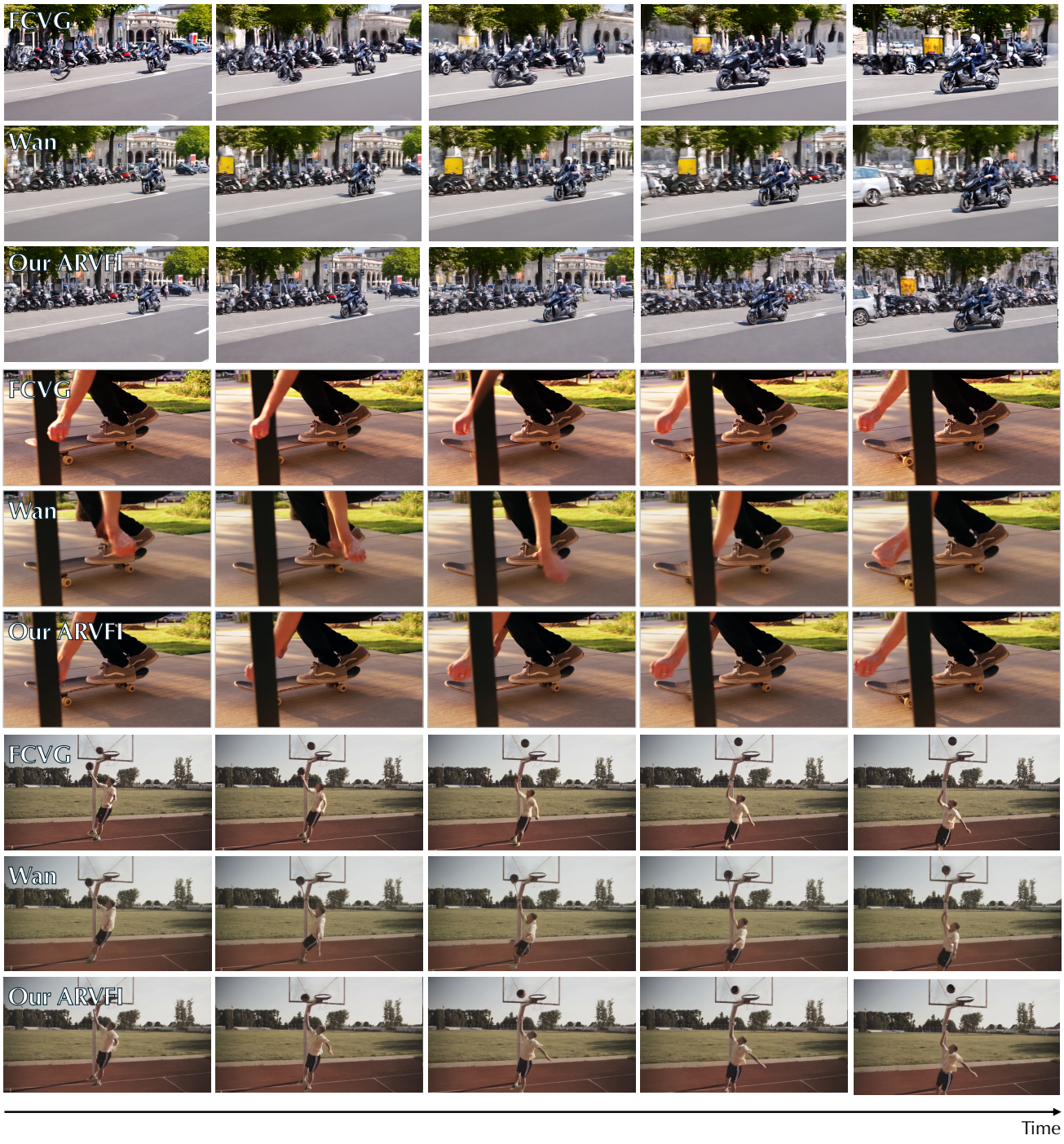

Figure 11. Sequence video frame interpolation results. The first row comes from the FCVG dataset [42], and the second and third rows are from our self-collected Pixels dataset. Compared with FCVG [42] and Wan [33], the proposed ARVFI generates correct object motions in DINOv3 data space and accordingly produces more consistent and accurate interpolation results.