

MASS: Motion-Aligned Selective Scan for Refinement in Flow-Based Video Frame Interpolation

Jun-Sang Yoo[✉] and Seung-Won Jung^{✉*}

Department of Electrical Engineering, Korea University, Seoul, Korea
{junsang7777, swjung83}@korea.ac.kr

Abstract. Video frame interpolation (VFI) remains a challenging task, particularly when dealing with large, non-linear motions and complex occlusions. While flow-based methods are prevalent, they often struggle with ambiguous correspondences. Recent VFI methods based on selective State Space Models (SSMs) are still limited by static grid-based scanning that misaligns with physical motion. In this paper, we propose Motion-Aligned Selective Scan (MASS), a novel framework that reformulates feature scanning from static spatial grids to dynamic motion trajectories. MASS builds a feature sequence along each pixel’s flow-guided trajectory and aggregates it with an SSM. Specifically, we introduce a learnable non-linear path integration to approximate complex curved trajectories via residual velocity updates, and a velocity-aware SSM that dynamically adjusts the sampling budget and step size based on motion magnitude. This adaptive strategy allocates denser sampling to fast-motion regions while keeping static regions efficient. Furthermore, the aggregated states guide a refinement module to rectify intermediate flows and masks in an end-to-end manner. Extensive experiments indicate that MASS achieves highly competitive overall performance on standard benchmarks, establishing state-of-the-art results particularly in challenging scenarios with large displacements and complex dynamics.

Keywords: Video Frame Interpolation · Selective State Space Models · Trajectory-wise Modeling · Motion-Guided Refinement

1 Introduction

Video frame interpolation (VFI) aims to synthesize an intermediate frame I_t at time $t \in (0, 1)$ from two consecutive frames I_0 and I_1 , enabling applications such as slow-motion generation [12, 29, 37], video compression [33, 36], and novel view synthesis [5, 14, 22]. While recent works [4, 39, 42] also explore diffusion or higher-level object/foreground cues to improve the perceptual fidelity, flow-based approaches [2, 10, 15, 16, 28] remain a prevalent paradigm for enforcing temporal alignment by explicitly modeling inter-frame motion. They estimate

* Corresponding author.

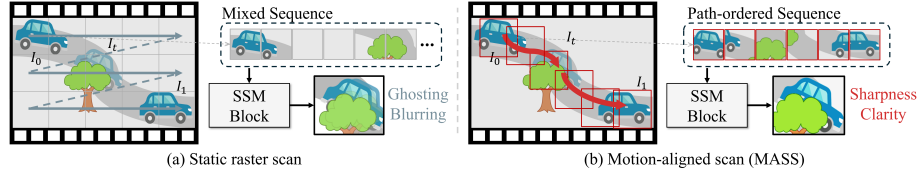


Fig. 1: Comparison of feature serialization strategies for VFI. (a) Static raster scanning flattens unaligned grids into mixed sequences, causing ghosting artifacts. (b) MASS extracts features along surrogate trajectories forming motion-consistent sequences and reducing misalignment in dynamic regions.

intermediate flows toward time t , warp the input frames, and reconstruct I_t by blending the warped frames with a synthesis network.

Despite their success, estimating intermediate flow is inherently challenging because the intermediate frame is unobserved. Even with multi-scale or iterative estimators [32, 34], large displacement, occlusion/disocclusion, thin structures, and repetitive textures often lead to ambiguous correspondences, which degrade warping-based reconstruction. To address this, recent methods [19, 29] such as SGM-VFI explicitly strengthen the correspondence stage by introducing global matching cues. By establishing sparse global correspondences between I_0 and I_1 and converting them into intermediate-flow compensation, these methods improve robustness under large motion. However, correspondence-centric approaches remain insufficient in regions where reliable matches are fundamentally unavailable, such as disocclusions and motion boundaries, forcing the interpolation pipeline to operate with only approximate flow estimates.

In parallel, another line of work explores global modeling architectures, including Transformers [35] and selective state space models (SSMs) [6] for VFI [11, 17, 23, 40, 41]. These models offer an efficient mechanism for long-range context aggregation. However, such designs still depend on the reliability of the underlying motion hypothesis and typically constrain aggregation to spatially defined windows for efficiency, which can be challenged by occlusion and highly non-linear motion. Furthermore, most existing SSM-based VFI methods [11, 40] serialize features using scan orders defined on static grids (e.g., raster or local windows). For VFI, the most informative context for a target pixel often lies along its physical motion path rather than in a static spatial neighborhood. This mismatch motivates us to replace grid-defined serialization with flow-guided trajectory serialization, better matching the continuous, motion-driven nature of the task.

In this paper, we introduce **Motion-Aligned Selective Scan (MASS)**, which shifts the scan order from static spatial grids to dynamic motion trajectories, turning per-pixel interpolation into trajectory-wise sequence modeling as illustrated in Fig. 1. Given coarse intermediate flows, MASS constructs a flow-guided surrogate trajectory for each pixel. Features sampled along this trajectory are flattened into a one-dimensional sequence. By aligning serialization with

the motion hypothesis, MASS aggregates context along flow-guided trajectories rather than a fixed spatial grid, reducing interference from motion-inconsistent neighbors.

Specifically, MASS incorporates two mechanisms to overcome the limitations of linear interpolation: learnable velocity and adaptive sequence length. Real-world motion is rarely perfectly linear. To address this, MASS updates the trajectory state at each step using a learnable residual, enabling the model to approximate complex, curved paths. We vary the number of sampled points with motion magnitude, so longer trajectories are sampled more densely. To prevent temporal aliasing and preserve transient details in fast-moving regions, our model assigns longer sequence lengths (i.e., denser sampling along the trajectory) to pixels with large displacements, while static areas are processed with shorter sequences for efficiency.

To demonstrate the effectiveness of our framework, we evaluate its performance on various standard benchmarks. The results show that MASS achieves state-of-the-art performance, particularly exhibiting superior robustness in complex scenarios with large displacements and occlusions without incurring significant computational overhead.

Our contributions are summarized as follows:

- We propose Motion-Aligned Selective Scan (MASS), which replaces static grid serialization with flow-guided, learnable non-linear trajectories to better capture the continuous dynamics of frame interpolation.
- We introduce a motion-adaptive discretization strategy that dynamically adjusts sequence sampling and SSM step sizes, alongside a refinement module that uses forward-backward context discrepancies to correct intermediate motions.
- Extensive experiments show MASS is highly competitive across standard benchmarks, achieving the most significant gains in challenging scenarios with large displacements and high-resolution inputs.

2 Related Work

2.1 Flow-based Video Frame Interpolation

Flow-based VFI synthesizes an intermediate frame by estimating intermediate-time correspondences and visibility cues (e.g., blending/occlusion masks), and reconstructing I_t via warping and fusion [2, 12]. A major line of work derives these flows from bidirectional estimations [20, 26], while another line directly predicts them via dedicated supervision [10, 13, 15]. Representative models like RIFE [10] establish highly efficient warp-and-merge paradigms, whereas IFR-Net [15] emphasizes the intrinsic difficulty of intermediate flow estimation due to the absence of the target frame.

To strengthen inter-frame interaction and capture broader context, attention-augmented methods [23, 41] have been widely adopted. VFIFormer [23] and EMA-VFI [41] demonstrate that cross-frame attention effectively balances detail

preservation and global reasoning. TTVFI [18] introduces endpoint-conditioned linear motion modeling in Transformers to mitigate warping inconsistencies; however, it models trajectories simply as linear displacements between two endpoints and constrains its attention mechanism to static spatial windows.

Nevertheless, large displacement and high-resolution scenarios remain challenging, as correspondence errors can be amplified by warping and become difficult to correct. Several works address this through scalable training strategies (e.g., XVFI [30]), scale-robust estimation (e.g., FILM [29]), or stronger correlation-and-refinement schemes (e.g., AMT [17]). Other approaches expand the effective matching range by incorporating bilateral correlation modeling [27, 28] or sparse global matching compensation [19].

Rather than introducing additional matching branches or relying on fixed interaction patterns, we revisit refinement from the perspective of motion-consistent aggregation. Our goal is to construct representations that remain informative even under local correspondence uncertainties by explicitly aggregating features along non-linear motion trajectories.

2.2 State Space Models for Vision and VFI

SSMs [6, 7] have emerged as an efficient, linear-time alternative to attention for sequence modeling. To apply SSMs to images, Vision Mamba variants serialize 2D feature maps into 1D token sequences. Early designs explore bidirectional and cross-axis scanning to mitigate the loss of 2D spatial structure [21, 43], while subsequent works further optimize scan-order designs to better preserve locality and continuity [8, 25].

SSM-based approaches for VFI remain relatively underexplored. VFIMamba [40] adapts selective scanning to two-frame modeling by arranging tokens from I_0 and I_1 into an interleaved sequence for efficient multi-directional mixing. LC-Mamba [11] emphasizes locality by introducing shifted-window and Hilbert curve-based scan orders, further extending scanning to spatiotemporal tokens. While these methods demonstrate that SSMs can serve as highly efficient feature mixers for VFI, their scan orders remain defined over predetermined token traversals on static grids (spatial or spatiotemporal).

Unlike prior SSM-based VFI methods, we avoid serializing tokens along fixed spatial traversals. Instead, we sample features along flow-guided trajectories and use the resulting forward and backward contexts for both frame synthesis and intermediate motion refinement. This trajectory-aligned aggregation helps mitigate alignment errors in large-displacement and occlusion-heavy scenarios.

3 Method

3.1 Overview

Our goal is to synthesize an intermediate frame $\hat{I}_t$ at time $t \in (0, 1)$ (default $t = 0.5$) from two consecutive frames I_0 and I_1 . As illustrated in Fig. 2, the

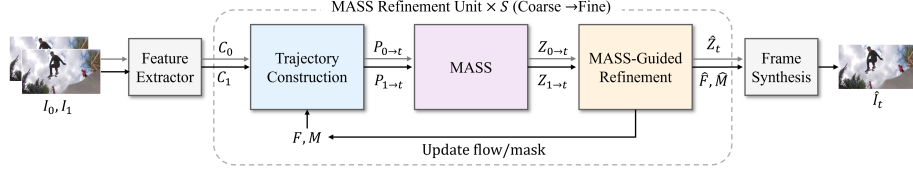


Fig. 2: Overall pipeline of our framework. Features extracted from I_0 and I_1 are processed through a cascade of MASS refinement units in a coarse-to-fine manner, producing refined motion and context to synthesize the target frame $\hat{I}_t$.

overall pipeline follows a coarse-to-fine design to robustly handle large non-linear motions and complex occlusions. We first extract feature maps C_0 and C_1 from I_0 and I_1 using a shared feature extractor. At each scale, the model repeats three steps: trajectory construction, trajectory-wise scan, and motion refinement.

Specifically, we start from coarse bidirectional flows $F = \{F_{t \rightarrow 0}, F_{t \rightarrow 1}\}$ (and an occlusion/blending mask M) estimated by a lightweight estimator [15]. These coarse motions provide a baseline for building pixel-wise trajectories $P_{0 \rightarrow t}$ and $P_{1 \rightarrow t}$. Unlike standard VFI methods that aggregate features on a static grid (e.g., via regular convolutions or window attention), our MASS aggregates features along these trajectories. As a result, MASS produces forward and backward motion-aligned contexts $Z_{0 \rightarrow t}$ and $Z_{1 \rightarrow t}$, which are later fused into $\hat{Z}_t$ for frame synthesis.

Crucially, the aggregated features serve two roles in our framework. First, they provide rich semantic context for frame synthesis. By gathering information along the motion trajectory, they preserve complex structures and fine textures that pure flow-based warping often misses. For synthesis, the forward and backward contexts are fused into $\hat{Z}_t$ and combined with the refined motion in the synthesis module to generate the target frame $\hat{I}_t$.

Second, the aggregated features provide a cue for motion reliability. In non-occluded regions, forward and backward trajectories should trace the same physical content, so their aggregated contexts should be consistent. We use the discrepancy between these contexts in the refinement module to identify trajectory errors and occlusions, and to update the coarse flow and mask, producing refined estimates $\hat{F}$ and $\hat{M}$. These updated motion estimates are then fed into the next stage to reconstruct more accurate trajectories, forming an iterative coarse-to-fine update loop.

3.2 Motion-Aligned Selective Scan

To overcome the limitations of static grid aggregation, MASS reformulates VFI as a trajectory-wise sequence modeling task. For a target pixel p , instead of restricting the receptive field to a spatial window, we utilize the initial coarse motion to trace a continuous flow-guided path. We then sequentially sample features along this trajectory and aggregate them using an SSM, explicitly align-

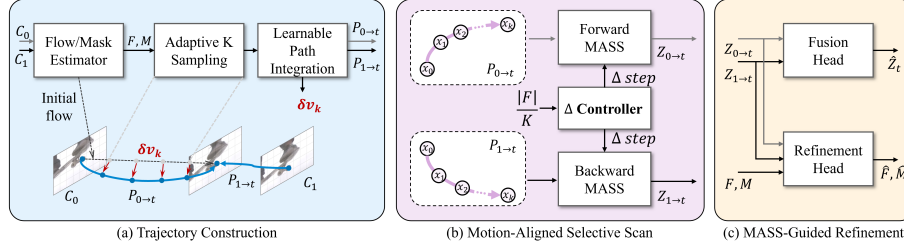


Fig. 3: Detailed illustration of the MASS refinement unit. (a) Non-linear trajectories are constructed from coarse flows through motion-adaptive sampling and learnable residual updates ($\delta \mathbf{v}_k$). (b) Trajectory-aligned feature sequences are aggregated with a velocity-aware scan, which adaptively modulates the discretization step according to the trajectory velocity ($\|F\|/K$). (c) The resulting forward and backward contexts are fused into $\hat{Z}_t$ for frame synthesis, while their discrepancy is used to refine the intermediate motion estimates ($\hat{F}, \hat{M}$).

ing the context with physical motion and avoiding interference from misaligned neighbors.

This process involves two coupled mechanisms. First, we construct non-linear trajectories from coarse motion estimates, allowing the sampling process to follow complex motion paths rather than relying on a straight-line assumption. Second, we aggregate the resulting trajectory-aligned feature sequences using a velocity-aware selective SSM (VA-SSM) to obtain motion-aligned contexts. We apply this process symmetrically to both forward ($0 \rightarrow t$) and backward ($1 \rightarrow t$) directions, producing $Z_{0 \rightarrow t}$ and $Z_{1 \rightarrow t}$. These contexts are then used not only for synthesis but also to guide motion refinement, as summarized in Fig. 5.

3.2.1 Learnable Non-Linear Path Integration. Linear interpolation assumes constant velocity, approximating motion as a straight line. However, complex real-world dynamics (e.g., rotation, acceleration) require curved trajectories. We construct such trajectories via an iterative integration process that progressively refines sampling coordinates starting from the coarse flow.

We first determine how densely to sample the trajectory. We assign a dynamic sampling budget $K(p)$ based on the motion magnitude. Intuitively, pixels with larger flow magnitude require denser sampling to avoid undersampling long trajectories. If we sample these long paths with too few points, sparse sampling can under-approximate curved motion and may lead to aliasing artifacts. Conversely, static regions require fewer steps.

Therefore, we dynamically set the sampling budget $K(p)$ by scaling the average magnitude of the bidirectional flows, defined as $\|F(p)\| = \frac{1}{2}(\|F_{t \rightarrow 0}(p)\| + \|F_{t \rightarrow 1}(p)\|)$. Using a scaling factor α , we round the scaled magnitude to the nearest integer and restrict it to a predefined range $[K_{\min}, K_{\max}]$:

$$K(p) = \text{clip}(\lfloor \alpha \cdot \|F(p)\| \rfloor, K_{\min}, K_{\max}) \quad (1)$$

Given the sampling budget $K(p)$, we define the trajectory by iteratively updating sampling coordinates relative to the fixed target pixel. Let $\mathbf{u}$ denote the coordinate of pixel p in the target frame. We aim to trace a sequence of points $\{\mathbf{u} + \mathbf{d}_k\}_{k=0}^{K(p)}$ towards the source frame, so that sampling follows the underlying motion. We initialize the accumulated displacement as zero ($\mathbf{d}_0 = \mathbf{0}$). At each step k , we first sample a local feature vector from the source feature map at the current traced coordinate $\mathbf{u} + \mathbf{d}_k$ via the backward warping operation [15], denoted as $\overleftarrow{\mathcal{W}}(C_0, \mathbf{u} + \mathbf{d}_k)$. We then feed this motion-aligned feature together with the coarse flow $F_{t \rightarrow 0}(p)$ and the normalized progress indicator $r_k = k/K(p)$ into two convolutional layers $\phi_{0 \rightarrow t}$, which predict a residual velocity $\delta \mathbf{v}_k$. This residual term serves as a learnable correction to the constant-velocity assumption; it lets the trajectory deviate from linear interpolation when local appearance indicates curved motion. For the forward direction ($0 \rightarrow t$), this is written as:

$$\delta \mathbf{v}_k = \phi_{0 \rightarrow t} \left(\overleftarrow{\mathcal{W}}(C_0, \mathbf{u} + \mathbf{d}_k), F_{t \rightarrow 0}(p), r_k \right), \quad \text{with } r_k = \frac{k}{K(p)}. \quad (2)$$

The displacement is then updated by taking one coarse-flow step (a constant increment of $F_{t \rightarrow 0}(p)/K(p)$) and adding the predicted residual:

$$\mathbf{d}_{k+1} = \mathbf{d}_k + \frac{F_{t \rightarrow 0}(p)}{K(p)} + \delta \mathbf{v}_k. \quad (3)$$

In other words, each step advances the trajectory by a coarse-flow increment plus a learned residual offset, allowing the path to bend when local appearance deviates from the straight-line assumption. Repeating this process yields a non-linear trajectory and a trajectory-aligned feature sequence. We apply the same construction to the backward direction ($1 \rightarrow t$) using C_1 and $F_{t \rightarrow 1}$, producing $P_{1 \rightarrow t}$. To enable efficient batched computation, we implement variable-length trajectories using a fixed-length buffer with a masking strategy. More details are provided in the supplementary material.

3.2.2 Velocity-Aware Selective Scan. Given the extracted feature sequence, the subsequent step is to aggregate these observations into a unified context vector $Z(p)$. Rather than relying on an aggregation mechanism, we introduce a Velocity-Aware SSM (VA-SSM) to adaptively modulate the internal state transition dynamics based on the underlying per-step motion along the trajectory. Intuitively, larger displacements per step correspond to faster appearance changes along the sequence. A fixed discretization may under-resolve rapid appearance variation along high-velocity trajectories, which can reduce the fidelity of fine textures.

To address this, we modulate the discretization step size according to the trajectory’s average velocity. In discretized SSMs, the system evolution depends on a step size Δ . While $K(p)$ controls how densely we sample points in space along the trajectory, Δ controls how finely the SSM evolves along the resulting token sequence. Since the trajectory spans an approximate displacement of

$\|F(p)\|$ with $K(p)$ steps in total, we define the average trajectory velocity as $v(p) = \|F(p)\|/K(p)$, where $\|F(p)\|$ is the average flow magnitude introduced above. Because $K(p)$ is discretized to a bounded range, the resulting step velocity still varies across pixels. We then adjust the trajectory-specific step size $\Delta(p)$ inversely with $v(p)$, which changes the effective transition used in the scan:

$$\bar{A}(p) = \exp(\Delta(p)A), \quad \text{with} \quad \Delta(p) = \frac{1}{1 + \beta \cdot v(p)}, \quad v(p) = \frac{\|F(p)\|}{K(p)} \quad (4)$$

By reducing $\Delta(p)$ for trajectories with large average velocities, the model updates its internal state with finer temporal granularity, preserving high-frequency details such as edges and textures during context aggregation. Applying the selective scan on the forward and backward trajectories yields the motion-aligned contexts $Z_{0 \rightarrow t}$ and $Z_{1 \rightarrow t}$.

3.3 MASS-Guided Refinement and Synthesis.

The MASS operation yields two motion-aligned contexts, $Z_{0 \rightarrow t}$ and $Z_{1 \rightarrow t}$. We utilize these features simultaneously to correct geometric errors and to synthesize the final frame.

We first identify trajectory errors by checking the consistency between the forward and backward contexts. Ideally, for non-occluded regions, the semantic context aggregated along the underlying motion path should be similar regardless of the scanning direction. We compute a discrepancy map $D(p) = |Z_{0 \rightarrow t} - Z_{1 \rightarrow t}|$, where a large difference indicates trajectory misalignment or occlusion. We use the discrepancy between forward and backward contexts as a refinement cue. A two convolutional layer head takes this cue, together with the aggregated contexts and the current motion estimates, and predicts residual updates to produce the refined flow $\hat{F}$ and mask $\hat{M}$. In this way, visual consistency directly guides geometric correction, and the refined motion is reused in the subsequent iteration to improve trajectory construction.

Meanwhile, frame synthesis requires a unified semantic representation. A two convolutional layer fusion head merges $Z_{0 \rightarrow t}$ and $Z_{1 \rightarrow t}$ into a single context $\hat{Z}_t$. This head employs a softmax-based gating operation to effectively down-weight unreliable information from occluded regions. Using the refined motion and fused context, we generate the target frame by combining motion-compensated warping with residual synthesis. We first compute a structural baseline I_{base} by warping the input frames using the refined flow $\hat{F}$ and mask $\hat{M}$:

$$I_{\text{base}} = \hat{M} \odot \overleftarrow{\mathcal{W}}(I_0, \hat{F}_{t \rightarrow 0}) + (1 - \hat{M}) \odot \overleftarrow{\mathcal{W}}(I_1, \hat{F}_{t \rightarrow 1}). \quad (5)$$

Here, $\odot$ represents the Hadamard product. Subsequently, the generator $\mathcal{G}$ synthesizes the final image by integrating the global motion context $\hat{Z}_t$ with the local source features (C_0, C_1) to restore fine-grained details:

$$\hat{I}_t = I_{\text{base}} + \mathcal{G}(\hat{Z}_t, C_0, C_1). \quad (6)$$

In this formulation, I_{base} provides the structural foundation based on corrected motion, handling large displacements. For the generator $\mathcal{G}$, we adopt a U-Net-shaped architecture [10]. By feeding both the aligned MASS context and raw local features into this architecture, $\mathcal{G}$ effectively recovers complex textures and edges that simple warping might miss. More details are provided in the supplementary material.

4 Experimental Results

4.1 Datasets

We evaluate our proposed method on comprehensive benchmark datasets covering a wide range of resolutions and motion complexities. **Vimeo90K** [38] is our primary training set. It contains 51,312 triplets for training and 3,782 frame triplets for testing, with a resolution of 448×256 . This dataset is widely recognized for its diversity in scene content and realistic motion patterns. **UCF101** [31] includes a test set of 379 frame triplets at 256×256 . **Middlebury** [1] uses the OTHER set at about 640×480 resolution. **SNU-FILM** [3] includes 1,240 frame triplets at 1280×720 , categorized by motion difficulty (Easy, Medium, Hard, Extreme). Lastly, **Xiph** [24] is tested in two versions to evaluate high-resolution performance: Xiph-2K, with images down-sampled to 2K, and Xiph-4K, with centrally cropped 4K images.

4.2 Implementation Details

Our framework is implemented in PyTorch. We train our model using the AdamW optimizer with a weight decay of 1×10^{-4} . The training is conducted on two NVIDIA RTX 3090 GPUs with a total batch size of 32. The feature extractor adopts the SGM-VFI [19] architecture, where the deepest features are at $H/8 \times W/8$ resolution. The refinement is performed progressively, with the $1/8$ and $1/4$ scales serving as the coarse and fine stages, respectively, via upsampling. For the MASS module, we set the dynamic sampling budget range $[K_{\min}, K_{\max}]$ to $[2, 8]$. The discretization step size Δ for the Velocity-Aware SSM is adaptively modulated within the range of $[0.25, 1.0]$ based on motion magnitude. The scaling parameters α and β in Eq. 1 and Eq. 4 are dynamically computed by normalizing local motion magnitude and velocity against their frame-level averages. For the SSM configuration, we balance efficiency and performance by setting the internal state dimension to 16 and the expansion factor to 2. For the objective function, we utilize the Laplacian pyramid loss [26] to guide the network in reconstructing multi-scale structures and fine details. Further details are provided in the supplementary material.

4.3 Comparison with Previous Methods

To evaluate the proposed MASS framework, we compare it against recent VFI methods across five standard benchmarks, as summarized in Table 1. For a fair

Table 1: Quantitative comparison across benchmarks (IE for Middlebury (M.B.); PSNR/SSIM for Vimeo90K, UCF101, Xiph, and SNU-FILM). The best and second-best results are in **bold** and underlined, respectively. † indicates models originally trained with additional datasets, which we re-trained exclusively on Vimeo90K for fair comparison. Other results are derived from [9–11, 15, 17, 23, 29]. Evaluation procedures follow [31] for Vimeo90K, UCF101, and Middlebury, [26] for Xiph, and [15] for SNU-FILM, with test-time augmentation disabled. FLOPs are measured at a resolution of 1280×720.

Method	M.B.	Vimeo90K	UCF101	Xiph		SNU-FILM				FLOPs (T)
				2K	4K	Easy	Medium	Hard	Extreme	
IFRNet-B [15]	1.95	35.80/0.9794	35.29/0.9693	36.00/0.936	33.99/0.893	40.03/0.9905	35.94/0.9793	30.41/0.9358	25.05/0.8587	0.21
RIFE [10]	1.96	35.61/0.9779	35.28/0.9690	36.19/0.938	33.76/0.894	39.80/0.9903	35.76/0.9787	30.36/0.9351	25.27/0.8601	0.20
AdaCoF [16]	2.24	34.47/0.9730	34.90/0.9680	34.86/0.928	31.68/0.870	39.80/0.9900	35.05/0.9754	29.46/0.9244	24.31/0.8439	0.36
ABME [28]	2.01	36.18/0.9805	35.38/0.9698	36.53/0.944	33.73/0.901	39.59/0.9901	35.77/0.9789	30.58/0.9364	25.42/0.8639	1.30
DAIN [2]	2.04	34.71/0.9756	34.99/0.9683	35.95/0.940	33.49/0.895	39.73/0.9902	35.46/0.9780	30.17/0.9335	25.09/0.8584	5.51
VFIFormer [23]	<u>1.82</u>	36.50/0.9815	35.42/0.9699	36.89/0.945	34.50/0.906	40.12/0.9907	36.09/0.9798	30.67/0.9378	25.43/0.8643	47.71
AMT-L [17]	1.86	36.34/0.9813	35.39/0.9697	36.27/0.940	34.48/0.903	39.95/0.9904	36.09/0.9797	30.75/0.9377	25.41/0.8633	0.68
AMT-G [17]	1.83	36.53 0.9816	35.41/0.9698	36.38/0.940	34.63/0.903	39.88/0.9912	36.12/0.9805	30.77/0.9384	25.42/0.8644	2.2
EMA-S [41]	1.94	36.07/0.9794	35.34/0.9696	36.54/0.942	34.24/0.902	39.81/0.9903	35.88/0.9792	30.68/0.9371	25.47/0.8627	0.39
EMA [41]	1.84	36.50/0.9814	35.38/0.9697	36.74/0.944	34.54/0.905	39.57/0.9905	35.85/0.9797	30.80/0.9389	25.59/0.8650	1.64
SGM-VFI-S-1/2† [19]	1.96	35.98/0.9796	35.32/0.9695	36.58/0.942	34.23/0.901	40.06/0.9908	36.02/0.9796	30.58/0.9364	25.25/0.8590	1.96
VFIMamba-S† [40]	1.98	35.87/0.9793	35.32/0.9697	36.26/0.940	33.52/0.890	40.00/0.9907	35.81/0.9791	30.58/0.9378	25.30/0.8620	0.39
VFIMamba† [40]	1.90	36.46/0.9814	35.42/0.9700	36.82/0.944	34.12/0.898	40.20 <u>0.9911</u>	36.11/0.9800	30.79/0.9389	25.66/0.8657	1.54
LC-Mamba-B [11]	1.89	36.43/0.9813	35.39/0.9698	36.90/0.945	34.26/0.904	40.07/0.9909	36.08/0.9801	30.59/0.9375	25.35/0.8630	1.06
LC-Mamba-P [11]	1.92	36.53 0.9816	35.42/0.9699	<u>36.99</u> 0.946	34.49/0.906	40.16/0.9909	<u>36.17</u> <u>0.9802</u>	30.72/0.9382	25.48/0.8645	2.70
Ours-S	1.90	36.32/0.9805	<u>35.44</u> /0.9699	36.84/0.944	34.65/0.907	40.05/0.9908	36.10/0.9800	<u>30.90</u> /0.9387	<u>25.73</u> /0.8668	0.50
Ours		1.79 36.53 0.9816	35.48 / 0.9703	37.05 0.946 34.83 0.908	<u>40.17</u> /0.9909	36.23 <u>0.9802</u>	31.09 0.9407	26.04 0.8681		2.06

comparison, we evaluate all models under the same training data constraint. Since some recent methods utilize additional datasets (e.g., X4K1000FPS [30]) for fine-tuning, we re-trained them from scratch exclusively on the standard Vimeo90K dataset using their official codebases (denoted by †). More detailed analyses are provided in the supplementary material.

While MASS delivers highly competitive performance on standard benchmarks such as Vimeo90K and UCF101, it shows the clearest gains in challenging scenarios characterized by large, non-linear motions and complex occlusions. This is particularly noticeable on the SNU-FILM dataset, which categorizes videos by motion magnitude. As motion complexity increases, performance degrades as large motion and occlusion make alignment increasingly difficult. In contrast, MASS demonstrates clear performance improvements on both the Hard and Extreme subsets. Notably, on the Extreme split, our method yields 26.04 dB in PSNR, outperforming the previous best approach, VFIMamba [40], by 0.38 dB. This gain is consistent with the benefit of modeling non-linear, trajectory-aligned context under complex motion.

Furthermore, the proposed model maintains consistent performance in high-resolution scenarios. On the Xiph-2K and Xiph-4K benchmarks, MASS compares favorably against all competing methods, recording 37.05 dB and 34.83 dB in PSNR, respectively, outperforming recent state-of-the-art methods such as LC-Mamba. These results suggest that the velocity-aware scan helps preserve details in demanding high-resolution, large-displacement scenarios. Additionally, our lightweight variant, MASS-S, maintains competitive restoration quality with a reduced computational cost of 0.50 TFLOPs, demonstrating its applicability in resource-constrained environments.



Fig. 4: Visual comparison on SNU-FILM [3]. The red arrows highlight regions with large motions and fine details.

Finally, visual comparisons in Fig. 4 support the quantitative results. While prior methods frequently exhibit ghosting or severe blurring under large displacements, MASS reconstructs structurally coherent frames with sharper edges. This highlights the advantage of our trajectory-wise modeling. By tracking non-linear motion paths, MASS explicitly ensures semantic continuity, successfully mitigating the feature misalignment that typically degrades performance in dynamic scenes.

4.4 Ablation Study

To validate the effectiveness of the proposed MASS framework, we conduct component-wise ablation studies using the lightweight variant of our model (Ours-S). We report PSNR and SSIM on the SNU-FILM and Xiph datasets, alongside FLOPs measured at 720p resolution, computed using the mean adaptive K on the SNU-FILM Hard split. Additional ablation results are provided in the supplementary material.

4.4.1 MASS Components and Integrator Variants. Table 2 investigates the contribution of each component within our MASS framework, alongside an ablation of the integrator design. Since MASS constructs token sequences along motion trajectories rather than a predefined spatial order, directly replacing the trajectory construction with a static grid scan is not straightforward within the same architecture. To validate the effectiveness of our motion-aligned design, the upper part of Table 2 begins with a trajectory-based baseline using a linear path,

Table 2: Ablation study on different configurations of the MASS module and integrator variants. Performance is reported in PSNR/SSIM.

Settings	Learnable Path	Adpt. K	Vel. Δ	SNU-FILM		Xiph		FLOPs (T)
				Hard	Extreme	2K	4K	
MASS components								
Baseline (Linear + Vanilla SSM)	–	–	–	30.66/0.937	25.49/0.863	36.54/0.941	34.40/0.904	0.53
+ Non-linear Traj.	✓	–	–	30.81/0.938	25.60/0.865	36.69/0.942	34.54/0.905	0.54
+ Adpt. Sampling	✓	✓	–	30.85/0.938	25.65/0.865	36.76/0.943	34.58/0.905	0.49
+ MASS (Ours)	✓	✓	✓	30.90/0.939	25.73/0.867	36.84/0.944	34.65/0.907	0.50
Integrator component								
1D Conv	✓	✓	–	30.68/0.938	25.53/0.864	36.39/0.941	34.07/0.900	0.50
LSTM	✓	✓	–	30.80/0.938	25.56/0.865	36.30/0.940	34.10/0.900	0.54
Self-Attention	✓	✓	–	30.83/0.938	25.68/0.866	36.60/0.938	34.20/0.902	0.52
VA-SSM (Ours)	✓	✓	✓	30.90/0.939	25.73/0.867	36.84/0.944	34.65/0.907	0.50

a fixed sampling budget ($K(p) = 8$), and a vanilla SSM, and then incrementally adds each proposed component. Transitioning to a learnable non-linear path yields consistent gains, as learnable residual offsets effectively model complex, continuous motion dynamics. Furthermore, our adaptive sampling strategy dynamically allocates points based on motion magnitude. Beyond lowering FLOPs, it yields higher accuracy than using a fixed maximum sampling budget, suggesting that sequence simplification is actively beneficial for static or small motions. Finally, integrating the velocity-aware step size (Vel. Δ) converts the baseline SSM into our VA-SSM. By dynamically modulating the discretization step, the model adjusts its internal state transitions based on the trajectory velocity. This suggests that fast-moving trajectories are aggregated with appropriately fine granularity, while static or slow-moving regions maintain stable state representations.

Next, we evaluate the integrator component using the optimized trajectory sampling. As shown in the lower section of Table 2, standard sequence models such as 1D convolutions, LSTMs, and self-attention yield sub-optimal performance or incur higher computational costs. In contrast, our VA-SSM adaptively modulates its internal state transitions based on trajectory velocity. This dynamic aggregation achieves the highest accuracy across all benchmarks while matching the low FLOPs of a simple 1D convolution, demonstrating an optimal accuracy-efficiency balance uniquely suited for trajectory-wise modeling.

4.4.2 Scale Iterations and MASS-Guided Refinement. Table 3 analyzes the trade-off between restoration accuracy and computational efficiency regarding the number of scale iterations (S). A single MASS refinement unit ($S = 1$) provides insufficient refinement and feature aggregation, resulting in limited performance. Increasing the iterations to $S = 2$ significantly improves accuracy, suggesting that iterative processing enables finer refinement and boosts performance in scenarios with complex motion. However, adding a third iteration ($S = 3$) yields only marginal performance gains while substantially increasing the computational overhead. Therefore, we adopt $S = 2$ as the default configuration to maximize performance while maintaining practical efficiency.

Table 3: Ablation study on the number of scale iterations and the MASS-guided refinement heads. Performance is reported in PSNR/SSIM.

Settings	SNU-FILM		Xiph		Params (M)	FLOPs (T)
	Hard	Extreme	2K	4K		
Scale iterations						
$S = 1$	30.82/0.938	25.64/0.864	36.60/0.942	34.40/0.905	15.7	0.45
$S = 2$ (Ours)	30.90/0.938	25.73/0.866	36.84/0.944	34.65/0.907	16.5	0.50
$S = 3$	30.92/0.939	25.75/0.867	36.84/0.944	34.69/0.907	17.2	0.68
MASS-guided refinement						
w/o Fusion Head	30.63/0.937	25.50/0.860	36.49/0.941	34.26/0.904	16.4	0.49
w/o Refinement Head	30.56/0.936	25.44/0.864	36.46/0.940	34.20/0.903	16.4	0.49
Full Model (Ours)	30.90/0.938	25.73/0.866	36.84/0.944	34.65/0.907	16.5	0.50

In the lower section of Table 3, we further validate the necessity of the MASS-guided fusion and refinement heads. First, when the fusion head is removed, we observe that a naive aggregation of MASS features (e.g., simple element-wise addition) is insufficient. This result shows that our gating mechanism is necessary to effectively integrate the results from bidirectional contexts. Furthermore, the absence of the refinement head leads to a significant performance degradation. This suggests that leveraging the discrepancies between bidirectional contexts is essential for correcting geometric errors in the intermediate flow and occlusion mask.

4.5 In-Depth Analysis

4.5.1 Motion-Aligned Trajectory Controls. To examine the role of motion-aligned trajectory serialization in MASS, we conduct controlled experiments (Table 4) while keeping network weights and inference settings fixed. We first evaluate the impact of trajectory direction. The default aligned trajectory consistently achieves the best performance. Rotating the flow by $\pm 90^\circ$ (perpendicular to the motion) causes the most significant performance drop. This suggests that the model benefits from gathering context along the actual motion path rather than merely expanding the spatial sampling area. Negating or swapping flows results in smaller drops, likely because these paths still align with the underlying motion axis. Furthermore, altering the token serialization order via random permutation or reversal degrades quality, particularly on high-resolution sequences. Replacing the trajectory-guided scan with a static 3×3 spatial grid also leads to a large performance drop. This indicates that MASS relies on the sequential, motion-consistent ordering of features rather than just an unordered set of local points. We also analyze the role of residual-velocity integration (δv_k). While disabling it causes a modest performance drop, the gap between aligned and perpendicular trajectories persists. This implies that motion alignment itself is a key factor in the observed improvements.

Finally, Fig. 5 visualizes the learned trajectory correction. The overlays show how residual velocity updates bend the initial linear flow, allowing the sampling path to flexibly follow complex motions such as acceleration, rotation, and boundary movements.

Table 4: Controlled perturbations of motion alignment in MASS. (A) the trajectory direction, (B) the serialization order, and (C) the residual-velocity integration while keeping all other settings fixed.

Settings	SNU-FILM		Xiph	
	Hard	Extreme	2K	4K
(A) Trajectory direction perturbation				
Aligned (default)	30.90 / 0.939	25.73 / 0.867	36.84 / 0.944	34.65 / 0.907
Rotated +90° (perpendicular)	30.61 / 0.937	25.52 / 0.866	36.44 / 0.941	34.34 / 0.902
Rotated -90° (perpendicular)	30.63 / 0.937	25.51 / 0.866	36.45 / 0.941	34.38 / 0.903
Negated flow ($-F_{t \rightarrow 0}$, $-F_{t \rightarrow 1}$)	30.69 / 0.938	25.55 / 0.866	36.54 / 0.942	34.40 / 0.903
Swapped flows ($F_{t \rightarrow 1}$, $F_{t \rightarrow 0}$)	30.68 / 0.938	25.54 / 0.866	36.52 / 0.942	34.39 / 0.903
(B) Serialization order perturbation				
Trajectory order (default)	30.90 / 0.939	25.73 / 0.867	36.84 / 0.944	34.65 / 0.907
Random permutation	29.77 / 0.927	25.43 / 0.865	34.86 / 0.928	32.91 / 0.880
Reverse order	29.30 / 0.921	25.23 / 0.860	33.44 / 0.916	32.02 / 0.872
Static grid order (3×3)	26.75 / 0.885	24.06 / 0.837	28.92 / 0.826	29.44 / 0.855
(C) Residual-velocity integration (δv_k)				
Aligned (δv_k ON)	30.90 / 0.939	25.73 / 0.867	36.84 / 0.944	34.65 / 0.907
Aligned (δv_k OFF)	30.86 / 0.938	25.69 / 0.866	36.76 / 0.943	34.59 / 0.905
Rotated +90° (δv_k ON)	30.61 / 0.937	25.52 / 0.866	36.44 / 0.941	34.34 / 0.902
Rotated +90° (δv_k OFF)	30.50 / 0.937	25.44 / 0.865	36.34 / 0.940	34.30 / 0.902

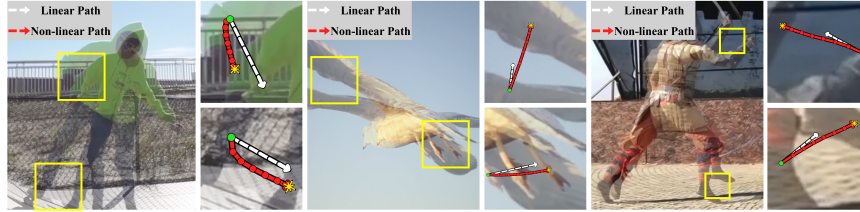


Fig. 5: Visualization of the learned trajectory correction.

4.5.2 MASS-Guided Refinement. Fig. 6 illustrates the internal mechanisms of MASS-guided refinement. Quantitatively, as shown in Fig. 6(a), the refinement consistently reduces initial coarse warping errors across progressive feature scales (from the coarse $1/8$ scale to the fine $1/4$ scale). The most substantial corrections occur at the $1/8$ scale, where severe initial motion errors are resolved, while finer geometric adjustments are left to the $1/4$ scale. To further investigate performance in regions with complex motion, we further isolate the top 20% of pixels with the highest motion magnitude. In these challenging regions, the error reduction is greater at the $1/8$ scale, whereas it drops at the $1/4$ scale. These results indicate that the $1/8$ scale effectively resolves the majority of large-displacement errors, leaving less residual misalignment for the $1/4$ scale to correct.

Qualitatively, Fig. 6(b) visualizes this refinement process at the coarse $1/8$ scale. A larger sampling budget K is allocated to the fast-moving foreground subject based on its motion magnitude. Correspondingly, the velocity-aware step size Δ is adjusted to smaller values in these regions, enabling finer state updates along the dynamic path. The discrepancy map D highlights local geometric misalignments. Guided by this map, the model refines the initial coarse motion, resulting in reduced errors compared to the coarse warping result. For a better

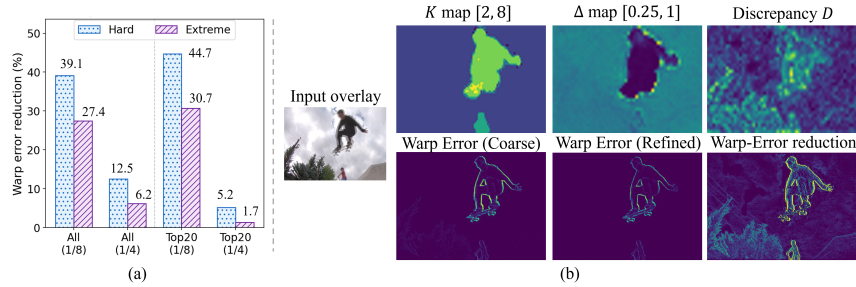


Fig. 6: In-depth analysis of MASS-guided refinement. (a) Average warp error reduction at the coarse 1/8 scale and the fine 1/4 scale for all pixels versus the top 20% high-motion regions on the SNU-FILM [3] Hard and Extreme splits. (b) Visualizations of internal representations at the coarse 1/8 scale: The input overlay is shown for spatial reference.

comparison between the two warp error maps, we visualize the error reduction. The observed reduction suggests that the overall MASS mechanism effectively handles dynamic motions, leading to notable performance improvements.

5 Conclusion

In this paper, we presented MASS, a novel framework that redefines the application of SSMs for VFI. By shifting the feature scanning paradigm from static spatial grids to dynamic motion trajectories, MASS effectively captures complex, non-linear motion dynamics that conventional serialization fails to resolve. Our key innovations—learnable non-linear path integration and velocity-aware adaptive sampling—efficiently allocate computational resources while preserving high-frequency details in fast-moving regions. Moreover, by exploiting forward/backward trajectory consistency, MASS refines intermediate flows and occlusion masks in a robust coarse-to-fine manner. Extensive experiments demonstrate that MASS establishes new state-of-the-art results, exhibiting particular superiority in challenging scenarios with large displacements and complex dynamics. Despite its robustness, MASS inherently relies on the guidance of the initial coarse flow. In extreme cases where this initial estimation fails catastrophically (e.g., due to chaotic motion of tiny objects or severe motion blur), the trajectory-guided scanning may struggle to recover the correct semantic context. Future work will address this dependency by exploring joint optimization strategies, training the flow estimator and the SSM in a fully coupled manner to allow the SSM to correct flow guidance dynamically from the earliest stages.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-16068196).

References

1. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. *International Journal of Computer Vision* **92**(1), 1–31 (2011)
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3703–3712 (2019)
3. Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M.: Channel attention is all you need for video frame interpolation. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 34, pp. 10663–10671 (2020)
4. Danier, D., Zhang, F., Bull, D.: Ldmvfi: Video frame interpolation with latent diffusion models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 1472–1480 (2024)
5. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deep stereo: Learning to predict new views from the world’s imagery. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5515–5524 (2016). <https://doi.org/10.1109/CVPR.2016.595>
6. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: *First Conference on Language Modeling* (2024)
7. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396* (2021)
8. Hatamizadeh, A., Kautz, J.: Mambavision: A hybrid mamba-transformer vision backbone. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 25261–25270 (2025)
9. Hu, P., Niklaus, S., Sclaroff, S., Saenko, K.: Many-to-many splatting for efficient video frame interpolation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3553–3562 (2022)
10. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: *European Conference on Computer Vision*. pp. 624–642. Springer (2022)
11. Jeong, M.W., Rhee, C.E.: Lc-mamba: Local and continuous mamba with shifted windows for frame interpolation. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 17671–17681 (2025)
12. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slo-mo: High quality estimation of multiple intermediate frames for video interpolation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 9000–9008 (2018)
13. Jin, X., Wu, L., Chen, J., Chen, Y., Koo, J., Hahm, C.h.: A unified pyramid recurrent network for video frame interpolation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1578–1587 (2023)
14. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics* **35**(6) (2016)
15. Kong, L., Jiang, B., Luo, D., Chu, W., Huang, X., Tai, Y., Wang, C., Yang, J.: Ifrnet: Intermediate feature refine network for efficient frame interpolation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1969–1978 (2022)
16. Lee, H., Kim, T., Chung, T.y., Pak, D., Ban, Y., Lee, S.: Adacof: Adaptive collaboration of flows for video frame interpolation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5316–5325 (2020)

17. Li, Z., Zhu, Z.L., Han, L.H., Hou, Q., Guo, C.L., Cheng, M.M.: Amt: All-pairs multi-field transforms for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9801–9810 (2023)
18. Liu, C., Yang, H., Fu, J., Qian, X.: Ttvfi: Learning trajectory-aware transformer for video frame interpolation. *IEEE Transactions on Image Processing* **32**, 4728–4741 (2023). <https://doi.org/10.1109/TIP.2023.3302990>
19. Liu, C., Zhang, G., Zhao, R., Wang, L.: Sparse global matching for video frame interpolation with large motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19125–19134 (2024)
20. Liu, Y., Xie, L., Siyao, L., Sun, W., Qiao, Y., Dong, C.: Enhanced quadratic video interpolation. In: European Conference on Computer Vision. pp. 41–56. Springer (2020)
21. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y.: Vmamba: Visual state space model. *Advances in Neural Information Processing Systems* **37**, 103031–103063 (2024)
22. Liu, Y., Deng, Y., Chen, H., Yang, Z.: Video frame interpolation via direct synthesis with the event-based reference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8477–8487 (2024)
23. Lu, L., Wu, R., Lin, H., Lu, J., Jia, J.: Video frame interpolation with transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3532–3542 (2022)
24. Montgomery, C., Lars, H.: Xiph.org video test media (derf’s collection). Online, <https://media.xiph.org/video/derf> (1994)
25. Munir, M., Zhang, A., Marculescu, R.: Vcmamba: Bridging convolutions with multi-directional mamba for efficient visual representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3037–3046 (2025)
26. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5437–5446 (2020)
27. Park, J., Kim, J., Kim, C.S.: Biformer: Learning bilateral motion estimation via bilateral transformer for 4k video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1568–1577 (2023)
28. Park, J., Lee, C., Kim, C.S.: Asymmetric bilateral motion estimation for video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14539–14548 (2021)
29. Reda, F., Kontkanen, J., Tabellion, E., Sun, D., Pantofaru, C., Curless, B.: Film: Frame interpolation for large motion. In: European Conference on Computer Vision. pp. 250–266. Springer (2022)
30. Sim, H., Oh, J., Kim, M.: Xvfi: extreme video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14489–14498 (2021)
31. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012)
32. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8934–8943 (2018)
33. Tan, P., Feng, W.c.: Hint-guided video frame interpolation for video compression. In: Proceedings of the 7th ACM International Conference on Multimedia in Asia. pp. 1–7 (2025)

34. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. pp. 402–419. Springer (2020)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in Neural Information Processing Systems* **30** (2017)
36. Wu, C.Y., Singhal, N., Krähenbühl, P.: Video compression through image interpolation. In: Proceedings of the European Conference on Computer Vision (2018)
37. Xiang, X., Tian, Y., Zhang, Y., Fu, Y., Allebach, J.P., Xu, C.: 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. pp. 3370–3379 (2020)
38. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. *International Journal of Computer Vision* **127**(8), 1106–1125 (2019)
39. Yoo, J.S., Lee, H., Jung, S.W.: Video object segmentation-aware video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12322–12333 (2023)
40. Zhang, G., Liu, C., Cui, Y., Zhao, X., Ma, K., Wang, L.: Vfimamba: Video frame interpolation with state space models. *Advances in Neural Information Processing Systems* **37**, 107225–107248 (2024)
41. Zhang, G., Zhu, Y., Wang, H., Chen, Y., Wu, G., Wang, L.: Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5682–5692 (2023)
42. Zhang, Z., Chen, H., Zhao, H., Lu, G., Fu, Y., Xu, H., Wu, Z.: Eden: Enhanced diffusion for high-quality large-motion video frame interpolation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2105–2115 (2025)
43. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417* (2024)