Following Motion for Sequential Modeling in Video Frame Interpolation

Jaehyun Park[1] and Nam Ik Cho[1,2,3]

[1] IPAI, Seoul National University
[2] Department of Electrical and Computer Engineering, INMC, Seoul National University
[3] SNUAILab
{jaep0805,nicho}@snu.ac.kr

Abstract. State Space Models (SSMs) have surfaced as a promising architecture in Video Frame Interpolation (VFI), as they can capture long-range dependencies with linear computational complexity. However, their predefined scanning order limits their effectiveness in modeling the dynamic motion trajectories inherent in VFI problems. To tackle this challenge, we propose **M**otion-**G**uided **M**amba for **V**ideo **F**rame **I**nterpolation (MGMVFI), an adaptation of the selective state space model tailored explicitly for VFI. MGMVFI introduces Motion-Guided Serialization (MGS), which leverages optical flow to define a motion-adaptive 1D input order for the SSM. This aligns the causal state updates with semantically related tokens, enabling motion-consistent feature propagation, particularly for large and dynamic motions. Additionally, to mitigate the unreliable feature representations caused by inaccurate optical flow estimates, we introduce contextual synthesis that utilizes the surrounding spatial context for robust inter-frame feature synthesis. These components are seamlessly integrated within our tailored Mamba architecture, which also employs a lightweight refinement block to enhance local detail reconstruction at a reduced computational cost. Extensive experiments on standard VFI benchmarks demonstrate that MGMVFI achieves state-of-the-art performance, particularly on complex and dynamic motions, thereby establishing a new direction for sequence modeling in video interpolation.

Keywords: Video Frame Interpolation · State Space Models · Mamba

1 Introduction

Video Frame Interpolation (VFI) is a long-standing task in computer vision that aims to synthesize intermediate frames between consecutive input frames in a video sequence. This task plays a crucial role in numerous applications, such as video frame rate up-conversion [1, 2, 4, 14], slow-motion video generation [1, 19], and novel view synthesis [6, 20, 52]. By increasing temporal resolution, VFI improves visual smoothness and performance of downstream video tasks.

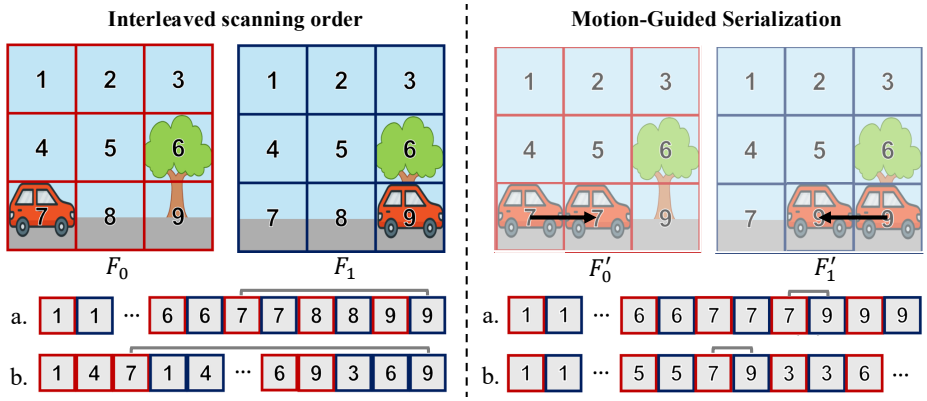

Fig. 1: We present Motion-Guided Serialization (MGS), which uses optical flow to sample from motion-aligned feature maps F_0' & F_1'. MGS results in *consecutive* semantically consistent tokens (Patch 7 & 9) for **a.** Horizontal and **b.** Vertical scans.

The effectiveness of deep learning-based VFI methods depends on how well the model captures the spatial and temporal dependencies in the input frame pair. Conventional approaches have primarily relied on convolutional neural networks (CNNs) [1,4,14,23] and Transformer-based architectures [31,41,48]. However, CNNs suffer from limited receptive fields, making them less suitable for modeling large motions that span an extensive range. Transformers, on the other hand, offer global attention mechanisms but suffer from quadratic computational complexity, limiting their scalability for high-resolution, real-time video processing.

Recently, state space models (SSMs), notably the Selective State Space (S6/ Mamba) family, have emerged as promising alternatives that combine the efficiency of convolutional operations with the dynamic modeling capability of recurrent structures [7–9]. S6-based architectures overcome the drawbacks of CNNs and Transformers by updating hidden states in a *selective* and *recurrent* manner, enabling the capture of long-range dependencies with linear complexity [7]. The inherent sequential nature of SSMs makes them highly dependent on the scanning order of the input features [16]. This sequence order is very pivotal, as it fundamentally dictates the flow of information through state updates, shaping the model's understanding of spatial and temporal relationships.

To process video inputs, prior works [7,10,11,25,28] have expanded 2D scanning strategies, such as horizontal, vertical, spiral scans, and Z-scans, by scanning each spatial coordinate sequentially for all images. Furthermore, 3D scans, such as the interleaved [47] and Hilbert curve-based scans [18], have also been specifically developed to learn features that cover both spatial and temporal dimensions. However, these pre-defined, motion-agnostic scanning paths are fundamentally misaligned with the dynamic nature of video. As shown in Figure 1, interleaved scanning [47] interleaves the columns of F_0 and F_1 to incorporate

the temporal dimension. However, this simple approach often aggregates semantically inconsistent features across time. For instance, Patches 7 and 9 represent features belonging to the same moving object. However, they have a large sequential distance (Gray line) in the interleaved scan's input sequence, forcing the SSM to both estimate the motion trajectories and model the inter-frame features. Moreover, the sequential distance depends on not only the magnitude of the motion but also the direction of the scan path, introducing additional variability that further hampers the estimation of motion trajectories.

Crucially, we reinterpret motion alignment as a *serialization* problem for sequence models. Classic flow-based VFI methods [1, 19, 37] estimate intermediate flows and visibility to warp features for spatial synthesis. In contrast, our goal is to define the *1D input sequence order* for the SSM: motion trajectories determine how tokens are serialized, so the causal state updates occur between semantically related patches. To this end, we propose a Motion-Guided Serialization (MGS) method that utilizes optical flow to dynamically rearrange input features along estimated motion paths. We combine this with patch-level interleaved scans to ensure that adjacent patches in the SSM input correspond to semantically-related and temporally-coherent content. The comparison is illustrated in Figure 1. Compared to interleaved scanning [47], our motion-aligned serialization method produces a semantically consistent input sequence, demonstrated by the adjacent features of the moving car in the serialized SSM input sequence (Patch 7 and 9). Additionally, because we interleave at patch-level, the semantically-related features stay adjacent regardless of scan direction.

However, MGS's dependence on optical flow estimation inherently creates inaccuracies that arise from occlusion/disocclusion, abrupt motion and lighting changes. We therefore further design a complementary masked adapter module to handle areas with inaccurate flow with contextual synthesis. Contextual synthesis identifies these information gaps and synthesizes robust features as replacements by leveraging the surrounding valid spatial context. The resulting motion-aligned sequence provides a rich, coherent input that vastly relaxes the motion estimation and inter-frame modeling objective for the SSM. This allows the downstream SSM to efficiently handle temporal propagation followed by the modified Efficient Discriminative Frequency-domain FFN (EDFFN) for enhanced local detail reconstruction at reduced computational cost [21].

Our contribution can be summarized in threefold:

1. **Motion-Guided Serialization (MGS)**: A novel serialization mechanism that leverages optical flow to dynamically define the SSM input sequence order along estimated motion paths, ensuring semantically consistent sequence modeling for complex and large-magnitude motions.
2. **Contextual Synthesis**: A robust strategy to restore corrupted feature representations in regions with inaccurate or missing flow by leveraging valid surrounding spatial context to synthesize replacement features.
3. **Adaptation of the Mamba architecture**: The integration of a modified Efficient Discriminative Frequency-domain FFN (EDFFN) into the Mamba framework to enhance local texture and detail reconstruction at a low cost.

2 Related Work

Video Frame Interpolation. Early deep learning-based VFI methods were predominantly built on CNNs. These approaches can be broadly categorized into flow-based and kernel-based methods. Flow-based VFI methods first estimate the optical flow between input frames and then warp these frames to the intermediate time step. Pioneering works, such as [30], introduced this concept, which has been followed by numerous refinements. These include direct estimation of intermediate flows [19, 29], bilateral motion estimation [39, 40], and the use of pre-trained contextual features to enhance the warping process [36]. More recent efforts have focused on efficiency, such as combining estimation and refinement into a single encoder [22] or using distillation [14]. Even in the modern landscape, advanced CNN architectures, such as SGM-VFI [27], continue to demonstrate competitive performance. Kernel-based VFI methods, pioneered by [38], bypass explicit flow estimation. Instead, they estimate spatially-adaptive convolution kernels that sample and weigh neighboring pixels from the input frames to directly synthesize the intermediate frame. To overcome the limitations of fixed local kernels, subsequent works explored adaptive filters [23] and deformable convolutions [2, 3] to better handle large motions and occlusions.

To address the limited receptive fields of CNNs and better model long-range dependencies, researchers began adopting the Transformer architecture. Initial works [31, 48] demonstrated the power of attention mechanisms for VFI. The Transformer was also applied to kernel-based methods [41] to address content-agnostic weights. Recently, EMA-VFI [48] has refined the Transformer architecture, achieving a balance of performance and efficiency that makes it a key state-of-the-art baseline.

A distinct line of research has focused on enhancing the perceptual quality of interpolated frames, particularly in challenging scenarios involving complex motion. Generative models, particularly Denoising Diffusion Probabilistic Models [12], have demonstrated remarkable potential. Works such as [17] and [5] have demonstrated that generative priors can yield more realistic results, albeit at a significant computational cost. More recently, EDEN [50] integrated a Transformer-based tokenizer with a DiT-style diffusion backbone, while Lyu et al. [32,33] adapted Brownian Bridge Diffusion Models [24] to set a high standard for perceptual quality and robustness. In this work, we set generative models aside and focus on deterministic models and their reconstruction performance.

State Space Models. State Space Models (SSMs) [9] were initially proposed for sequential signal modeling and later adapted for use in deep learning, utilizing HiPPO-based initialization and efficient structured representations. S4 [8] introduced parallelized updates to the hidden state, and S6 (Mamba) [7] further advanced the design with selective gating and hardware-aware implementations.

In vision applications, SSMs have been adapted to 2D and video domains [16]. Liu et al. [28] extended selective scanning to image patches, offering global receptive fields, while VideoMamba [25] incorporated hierarchical bidirectional state updates for efficient video sequence modeling. [49, 51] modeled motion-aware dynamic trajectories and temporal dependencies across frames to capture complex

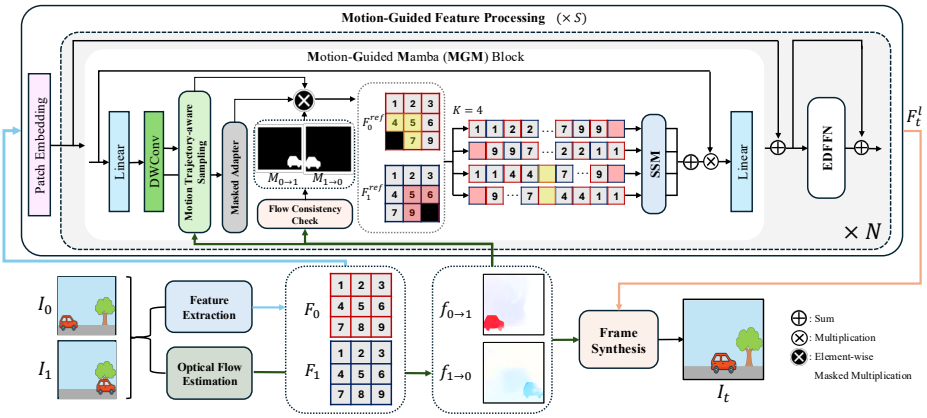

Fig. 2: Overall pipeline of our model. Our MGMVFI block first rearranges input features based on their motion trajectory. It then augments features in regions with unreliable optical flow using spatial context from the surrounding (highlighted) patches. The Contextual Synthesis is visualized for a 3×3 hollow kernel case.

temporal dynamics in video sequences. In image restoration, MambaIR [11] introduced local enhancement and channel attention to address pixel forgetting, followed by semantic-based reordering for local-global reasoning [10].

For frame interpolation, VFIMamba [47] was the first to adopt S6-based modeling, employing interleaved frame scanning and curriculum-based training to outperform both CNN- and transformer-based baselines. Afterward, LC-Mamba [18] adopted the Hilbert-curve scan method with shifted windows to maintain spatial continuity across window boundaries and extended this to a 3D voxel-level scan for better spatiotemporal modeling between frames. Our work builds on this line by introducing Motion-Guided Serialization. Unlike prior SSM-based models, which used fixed or heuristic scans, we leverage optical flow to dynamically rearrange scan paths along true motion trajectories, making the sequence modeling itself motion-adaptive.

3 Proposed Method

3.1 Overview

Our goal is to synthesize an intermediate frame $I_t \in \mathbb{R}^{H\times W\times 3}$ at an arbitrary timestep $t \in (0, 1)$ given two consecutive input frames, $I_0, I_1 \in \mathbb{R}^{H\times W\times 3}$. As illustrated in Figure 2, our model consists of three stages:

1. **Feature Extraction and Optical Flow Estimation:** A shared CNN encoder first extracts multi-scale deep feature maps, F_0 and F_1, from the input frames I_0 and I_1. Furthermore, a pre-trained optical flow model is used to estimate the corresponding flow maps: $f_{0\to1}$ and $f_{1\to0}$.

2. **Motion-Guided Feature Processing:** Our motion-guided pipeline processes the extracted features and optical flow with our proposed Motion-Guided Mamba for Video Frame Interpolation (MGMVFI) block. Within MGMVFI, the features are first processed using Motion-Guided Serialization and Contextual Synthesis to define a motion-adaptive serialization of the inputs. Then, the refined features are passed to our inter-frame feature modeling stage, which uses an SSM and the modified EDFFN block [21] to efficiently capture local details and high-frequency information.
3. **Frame Synthesis:** Finally, a frame synthesis module takes the motion-guided feature map F_t and additional flow priors to synthesize the final RGB output frame, I_t. We utilize the established generation modules presented in [48] and [14] to focus on the adaptation of SSMs.

3.2 Motion-Guided Serialization

The core of our MGMVFI block is **Motion-Guided Serialization (MGS)**, which reformulates how features are prepared for the SSM sequence model. Rather than using motion alignment for direct synthesis, MGS uses motion trajectories to define the serialization of features, i.e., the 1D token order fed into the SSM block. Concretely, we warp features to the intermediate grid so that standard raster scans over the resulting map correspond to motion-aligned sequences. This allows the causal flow of information to follow semantic motion, so SSM updates occur between related tokens and can focus on local residuals rather than global trajectory tracking. In practice, we serialize features by sampling along continuous motion trajectories, avoiding the limitations of fixed raster scans or linear blending that frequently produce ghosting artifacts near object boundaries.

To establish these trajectories, we must first estimate the intermediate flow fields originating from the target frame I_t. Given the bi-directional optical flow between the source frames, $f_{0\to1}$ and $f_{1\to0}$, we assume linear motion trajectories to scale the flow vectors. We employ forward splatting [37], denoted by $\mathcal{S}$, to push these scaled vectors onto the intermediate grid p_t:

$$f_{t\to0} = \mathcal{S}(-t \cdot f_{0\to1}), \qquad f_{t\to1} = \mathcal{S}(-(1-t) \cdot f_{1\to0}), \tag{1}$$

Then, calculate the source coordinates p_0 and p_1 by applying these intermediate flow fields to the target grid p_t:

$$p_0 = p_t + f_{t\to0}, \qquad p_1 = p_t + f_{t\to1} \tag{2}$$

Using these coordinates, we perform backward warping via bilinear sampling, denoted by $\mathcal{B}(\cdot,\cdot)$, to fetch the motion-aligned features from F_0 and F_1:

$$F_0' = \mathcal{B}(F_0, p_0), \qquad F_1' = \mathcal{B}(F_1, p_1) \tag{3}$$

Examples of the sampled F_0' and F_1' are illustrated in Figure 1. By sampling from these motion-aligned feature maps, semantically consistent patches (e.g.,

Patch 7 and 9) are always consecutively ordered in the 1D sequence, simplifying the learning objective of the SSM to focus solely on inter-frame feature modeling. To quantify this, we further define **Sequential Feature Similarity** on the 1D serialized sequence as

$$S = \frac{1}{N-1} \sum_{n=1}^{N-1} \exp\left(-\frac{\|\phi(z_{n+1}) - \phi(z_n)\|_2^2}{2\sigma^2} \right) \tag{4}$$

where N is the sequence length and $\phi(z_n)$ represents the feature representation of the n-th token in the sequence. We use this metric in our experiments (Table 5) to verify that MGS yields higher S, indicating smoother feature similarities between consecutive tokens and a simpler modeling target for the SSM.

3.3 Contextual Synthesis for Unreliable Regions

While backward warping populates the entire feature map, it fetches invalid or unreliable features in regions where accurate optical flow is unavailable. These failures occur not only due to physical occlusions and disocclusions, but also estimation inaccuracies caused by large motions or abrupt brightness changes. Consequently, the SSM receives invalid or corrupted tokens from these areas, resulting in substantial reconstruction errors. To address this, we introduce a novel **Contextual Synthesis** strategy designed to reconstruct these problematic regions by leveraging valid surrounding spatial information.

Flow Consistency Check for Reliability Masking. To identify such regions, we first identify unreliable motion paths using a standard forward-backward consistency check on the optical flow maps $f_{0\to1}$ and $f_{1\to0}$. A motion path is considered unreliable if a point p warped from I_0 to I_1 and back does not return to its original location:

$$\epsilon_{0\to1}(p) = \left|\left| f_{0\to1}(p) + f_{1\to0}\big(p + f_{0\to1}(p)\big) \right|\right|_2^2 \tag{5}$$

A binary consistency mask, which we denote as $M_{0\to1} \in \{0,1\}^{H'\times W'\times 1}$, is produced by thresholding this squared L2-norm against the average flow magnitude. This identifies pixels where the motion path has failed:

$$M_{0\to1}(p) = \begin{cases} 1 & \text{if } \epsilon_{0\to1}(p) > \alpha \cdot \overline{||f_{0\to1}||} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

This check is particularly crucial for handling algorithmic failures, such as those caused by abrupt changes in brightness. When lighting variations violate the brightness constancy assumption of the flow network, it often defaults to a zero-magnitude vector. If applied to a moving object, backward warping with this zero vector fetches incorrect spatial locations (e.g., the static background instead of the moving object), causing severe ghosting artifacts. Because our consistency check evaluates the bidirectional agreement of the flow, it successfully flags these zero-vector failures, ensuring unreliable regions are masked regardless of whether

they stem from physical occlusions or lighting variations. This bi-directional check produces two distinct binary masks:

- $M_{0\to1}$: A forward consistency mask identifying regions in I_0 whose features become unreliable when mapped to I_1 (e.g. occluded regions).
- $M_{1\to0}$: A backward consistency mask identifying regions in I_1 whose features are unreliable when mapped back to I_0 (e.g. disoccluded regions).

These masks remain in the coordinates of their respective source frames.

Contextual Synthesis via Masked Adapter. Synthesizing these unreliable regions is challenging because the corresponding features fetched from the opposite frame are corrupted. An intuitive example of this failure occurs during physical occlusion, as seen with Patch 9 in the interleaved scan of Figure 1. Because the tree trunk visible in F_0 (Patch 9) is covered by the moving car, features fetched from the same estimated location in F_1 (Patch 9) are incorrect. A straightforward solution is to use the mask $M_{0\to1}$ to simply zero out these incorrect features from F'_1. However, this forces the model to rely solely on the features from F_0, leading to low-quality synthesis characterized by blur and fading artifacts. To fill this information gap, we introduce a masked adapter module $\mathcal{A}_1$ that replaces the corrupted features of F'_1 with valid surrounding context. As shown by the highlighted patches in Figure 2, $\mathcal{A}_1$ uses a 7×7 hollow convolution, which convolves over the surrounding patches while explicitly excluding the unreliable center patch:

$$F_1^c = \mathcal{A}_1(F'_1) = Conv_{1\times1}(\text{GELU}(ConvHollow_{7\times7}(F'_1))) + F'_1 \tag{7}$$

By excluding the identified failure patch, we guide the adapter $\mathcal{A}_1$ to leverage only the robust, surrounding valid context to supplement the missing feature. A symmetric process is used to handle backward flow failures using a separate adapter $\mathcal{A}_0$ with the surrounding contextual features of F'_0.

Refined Feature Composition. The final feature maps for each stream, F^{ref}, are composed by selecting either the motion-sampled feature F' or the adapter's synthesized output F^c, guided by the appropriate mask:

$$F_1^{ref} = (1-M_{0\to1})\cdot F'_1 + M_{0\to1}\cdot F_1^c, \qquad F_0^{ref} = (1-M_{1\to0})\cdot F'_0 + M_{1\to0}\cdot F_0^c \tag{8}$$

Finally, these two refined feature maps from (8) are fused with the original features via an additive connection before being passed to the main SSM block:

$$F_1^{fused} = F_1 + F_1^{ref}, \quad F_0^{fused} = F_0 + F_0^{ref} \tag{9}$$

For each spatial location i in the target grid, we interleave the corresponding features from F_0^{fused} and F_1^{fused} to form a sequence token input z_i:

$$z_i = [F_0^{fused}(i) \parallel F_1^{fused}(i)] \tag{10}$$

where $\parallel$ denotes concatenation along the feature dimension. The final 1D sequence $Z = \{z_1, z_2, \ldots, z_N\}$ is then fed into the Mamba SSM block.

3.4 Inter-frame Feature Modeling

The feature modeling in our pipeline utilizes the following two key components. **SS2D Block.** We adopt the SS2D block [28] as our SSM. Following its design, we perform $K = 4$ different raster scans (horizontal, vertical, and their backward counterparts) on the motion-aligned input sequence Z. Outputs are then aggregated and multiplied by the original input features to modulate intensity according to the spatiotemporal context captured by the SSM. This gated representation then undergoes a final linear layer to produce the refined feature map. This adaptation allows robust estimation of motion trajectories and inter-frame feature modeling within a unified pipeline.
EDFFN Block. While our MGS simplifies the SSM's objective by semantically aligning features, it also introduces computational overhead. Furthermore, a recent study [34] suggests that the Mamba block tends to focus on low-frequency information, necessitating a complementary module to capture fine-grained details. To offset the computational cost and simultaneously enhance local feature modeling, we modify and adapt the Efficient Discriminative Frequency-domain FFN (EDFFN). EDFFN is uniquely suited to our aligned features, as its frequency-domain gating mechanism selectively refines *local* spatial details and enhances texture reconstruction, both of which are crucial for high-fidelity interpolation. On the other hand, the EDFFN's efficient design, which performs frequency operations on lower dimensions, provides significant computational savings.

4 Implementation Details

Training Datasets and Details. We train our model on the Vimeo-90K [46] triplet dataset, a large-scale collection known for its diverse motions. During training, the images are randomly cropped to 256×256 and augmented using horizontal, vertical, and temporal flipping, along with random rotations $(90°, 180°, 270°)$. We empirically set $S = 2$ and $N = 3$, repeating the feature processing for $\times 8$ and $\times 16$ downsampled features.

The model is trained for 150 epochs with a batch size of 8 on three NVIDIA RTX 3090 GPUs (approx. 3 days). We use the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 1×10^{-4}). The learning rate is initialized at 2×10^{-4} and gradually decreased to 2×10^{-5} using a cosine annealing schedule. To reduce computational overhead, we employ the pre-trained large RAFT [45] model and pre-compute the bi-directional flow maps. The flow consistency check threshold α is set to 1.1, which empirically yields the best results.
Objective Functions. Our proposed model is trained by minimizing a composite objective function, $\mathcal{L}_{total}$, which combines a primary reconstruction loss and an auxiliary warping loss:

$$\mathcal{L}_{total} = \mathcal{L}_{lap} + \lambda_1 \mathcal{L}_{warp}, \tag{11}$$

where $\mathcal{L}_{lap}$ is the $\mathcal{L}_1$ loss computed between the Laplacian pyramids of the final interpolated frame I_t and the ground-truth frame I_t^{GT}. The warping loss, $\mathcal{L}_{warp}$,

provides supervision on the image synthesized by the feature F_t^s of each s-th scale of our multiscale architecture. It is defined as the sum of Laplacian losses between the intermediate warped frames I_t^s and the ground truth:

$$\mathcal{L}_{\text{warp}} = \sum_s \mathcal{L}_{\text{lap}}(I_t^s, I_t^{\text{GT}}) \tag{12}$$

Based on empirical validation, the weighting coefficient λ_1 is set to 0.5.

5 Experiment Results

We compare our model against recent state-of-the-art (SotA) VFI models. As a deterministic model, we specifically choose to compare against CNN-based [1, 4, 13, 14, 22, 23, 26, 27, 37, 42], Transformer-based [48, 50], and S6-based [18, 47] models. The metrics reported for LC-Mamba are from its balanced model (Ours-B). For a more comprehensive qualitative assessment, we also include the diffusion-based generative model EDEN [50]. As a generative model that focuses on perceptual metrics, EDEN serves as a strong reference for qualitative performance.

Evaluation Datasets. Our model is evaluated on four test benchmarks. Vimeo90K [46] (448×256) and UCF101 [43] (256×256) consist of low-resolution real-world scenes, composed of varying motion magnitude. SNU-FILM [4] is a higher resolution dataset (1280×720) that is divided into four difficulty splits(Easy, Medium, Hard, and Extreme) depending on the motion magnitude. Xiph [35] consists of 4K-resolution frames, which we followed [37] to reproduce 2K-resolution frames and evaluate on both.

5.1 Quantitative Evaluation

To quantitatively evaluate our model, we report PSNR and SSIM, the standard reconstruction benchmarks in VFI literature. As shown in Table 1, our model achieves state-of-the-art (SotA) performance on the Vimeo-90K and UCF101 datasets across various motion magnitudes, exceeding prior methods. The primary advantage of MGMVFI is most evident on the challenging SNU-FILM benchmark. While it performs competitively on Easy and Medium splits, MGMVFI establishes new SotA results on the Hard and Extreme splits, which feature significant, complex motion. On the Extreme split, MGMVFI outperforms the next-best S6-based model, VFIMamba, by 0.07 dB, confirming the robustness of our motion-aware approach for high-fidelity synthesis.

The efficiency and high-resolution scalability of MGMVFI are further underscored by the Xiph benchmark results in Table 2. The performance on Xiph 4K highlights the importance of trajectory-aware serialization. While fixed scanning orders often fail to capture temporal dependencies at high resolutions where related features are spatially distant, MGMVFI dynamically aligns the sequence with the true motion path to effectively preserve structural integrity and fine textures. Furthermore, despite the motion-guided serialization step, our model

Table 1: Quantitative Evaluation across three benchmark datasets. The best and second best results are indicated by **bold** and underline.

	Train Dataset	Vimeo-90K [46]	UCF101 [43]	SNU-FILM [4]			
				Easy	Medium	Hard	Extreme
DAIN [1]	V	34.71/0.9756	34.99/0.9683	39.73/0.9902	35.46/0.9780	30.17/0.9335	25.09/0.8584
AdaCof [23]	V	34.47/0.9730	34.90/0.9680	39.80/0.9900	35.05/0.9754	29.46/0.9244	24.31/0.8439
CAIN [4]	V	34.65/0.9730	34.91/0.9690	39.89/0.9900	35.61/0.9776	29.90/0.9292	24.78/0.8507
Softsplat [37]	V	36.13/0.9805	35.39/0.9697	40.26/0.9911	36.07/0.9798	30.53/0.9365	25.16/0.8604
XVFI [42]	V	35.09/0.9759	35.17/0.9685	39.93/0.9907	35.37/0.9782	29.58/0.9276	24.17/0.8450
M2M-VFI [13]	V	35.47/0.9778	35.28/0.9694	39.66/0.9904	35.74/0.9794	30.30/0.9360	25.08/0.8604
RIFE [14]	V	35.61/0.9779	35.28/0.9690	39.80/0.9903	35.76/0.9787	30.36/0.9351	25.27/0.8601
IFRNet-L [22]	V	36.20/0.9808	35.42/0.9698	40.10/0.9906	36.12/0.9797	30.63/0.9368	25.26/0.8609
AMT-L [26]	V	36.35/0.9815	35.39/0.9698	39.95/**0.9913**	36.09/**0.9805**	30.75/0.9384	25.41/0.8638
AMT-G [26]	V	36.53/0.9817	35.41/0.9699	39.88/**0.9913**	36.12/**0.9805**	30.78/0.9385	25.43/0.8644
SGM-VFI [27]	V	35.81/0.9793	35.34/0.9693	40.14/0.9907	36.06/0.9795	30.81/0.9375	25.59/0.8646
EMA-VFI-S [48]	V	36.07/0.9797	35.34/0.9696	39.81/0.9906	35.88/0.9795	30.69/0.9375	25.47/0.8632
EMA-VFI [48]	V	36.64/0.9819	35.48/0.9701	39.98/0.9910	36.09/0.9801	30.94/0.9392	25.69/0.8661
EDEN [50]	LAVIB	32.66/0.9587	34.92/0.9676	38.69/0.9879	34.66/0.9744	29.59/0.9279	24.94/0.8536
VFIMamba [47]	V+X	36.64/0.9819	35.45/0.9702	**40.51**/0.9912	**36.40**/**0.9805**	30.99/0.9401	25.79/0.8682
LCMamba(B) [18]	V	36.43/0.9813	35.39/0.9698	40.07/0.9909	36.08/0.9801	30.59/0.9375	25.35/0.8630
Ours	V	**36.67**/**0.9820**	**35.53**/**0.9710**	40.45/0.9910	36.38/0.9804	**31.04**/**0.9411**	**25.86**/**0.8732**

Table 2: Quantitative evaluation on Xiph [35] dataset and efficiency comparison. Runtimes (RT) are measured for a 2048×1024 resolution input on an RTX A6000.

	Xiph 2K		Xiph 4K		Params(M)	RT(ms)
SGM-VFI	36.57	0.9424	34.23	0.9021	20.8	1168.9
EMA-VFI	36.74	0.9445	34.54	0.9054	65.6	332.3
EDEN	33.36	0.9043	26.84	0.7303	157.8	1304.4
VFIMamba	37.13	0.9451	34.62	**0.9059**	66.1	414.7
LCMamba(B)	36.90	**0.9456**	34.26	0.9040	16.2	785.6
Ours	**37.15**	0.9454	**34.65**	**0.9059**	78.4	427.5

maintains a competitive inference runtime of 427.5 ms. This efficiency stems from the EDFFN module, which selectively refines high-frequency details in the frequency domain, allowing for a streamlined sequence-modeling backbone.

5.2 Qualitative Evaluation

Figure 3 provides a qualitative comparison against leading VFI models using challenging scenes from the Vimeo90K and SNU-FILM Hard datasets. Scene 1 (columns 1–2) illustrates a rapid-motion example where the motion-agnostic scanning orders of prior S6-based models fail. Their fixed interleaved and Hilbert-curve scans provide semantically inconsistent patches, causing structural distortion. In contrast, our MGS module processes features along the motion path, providing a coherent sequence that simplifies the SSM's objective and produces structurally sound, sharp interpolations. Scene 2 (column 3) highlights a difficult disocclusion scenario. Baselines, including prior S6 models and EMA-VFI, exhibit prominent ghosting and background blur due to insufficient information in newly revealed regions. Conversely, our results demonstrate the effectiveness of the Contextual Synthesis mechanism; the masked adapter synthesizes robust

Fig. 3: Visual comparison of the interpolated results from the SNU-FILM [4] Hard and Vimeo90K [46] dataset. Best viewed zoomed in.

features from the surrounding valid spatial context. Combined with our modified EDFFN, the model handles information gaps to produce a crisp, artifact-free background. Scene 3 (column 4) demonstrates our model's superiority in reconstructing high-frequency details. While baselines suffer from blur or washout artifacts that sacrifice faithfulness—as seen in the generative EDEN model—our model renders the sharpest textures. This performance is attributable to the EDFFN module, whose frequency-domain gating is uniquely suited for texture reconstruction, as validated by the 0.14 dB gain in Table 3.

5.3 Ablation Study

Progressive Module Analysis. We conduct a progressive ablation study on the Vimeo-90K, UCF101, and SNU-FILM Hard datasets to validate the contributions of our proposed modules, with quantitative results and visual comparisons provided in Table 3 and Figure 4, respectively. We begin with a baseline model consisting of core SSM blocks, which struggles to handle dynamic move-

Fig. 4: Visual comparison of the progressive module analysis from Vimeo-90K [46].

ment, resulting in significant blur and ghosting artifacts. Integrating the modified EDFFN block yields a consistent improvement in sharpness across all benchmarks, raising the PSNR on the SNU-FILM Hard split to 29.80 dB. The addition of our core MGS method provides a substantial performance leap, increasing the SNU-FILM Hard PSNR by 0.44 dB to 30.24 dB. Visually, the +EDFFN and MGS configuration produces much clearer images with reduced structural artifacts, demonstrating the effectiveness of aligning the serialization order with estimated motion paths. Finally, introducing the Contextual Synthesis module yields the most significant gain, boosting the SNU-FILM Hard PSNR to 31.04 dB and the Vimeo-90K PSNR to 36.67 dB. This full model effectively reconstructs complex occlusions, such as a sheep's head, while correctly eliminating disoccluded background artifacts (hind leg) as guided by the consistency mask $M_{1\to0}$.

Table 3: Quantitative report of the progressive module analysis

Scan method	Vimeo-90K [46]		UCF101 [43]		Hard [4]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SSM	35.96	0.9793	35.37	0.9699	29.71	0.9293
+ EDFFN	36.10	0.9797	35.38	0.9701	29.80	0.9300
+ MGS	36.26	0.9802	35.56	0.9713	30.24	0.9345
+ Cont. Synth.	36.67	0.9820	35.53	0.9710	31.04	0.9411

Table 4: Performance comparison with different optical flow backbones on SNU-FILM Hard [4]

Flow Backbone	RT(ms)	Params(M)	PSNR	SSIM
LiteFlowNet	185.12	4.86	30.89	0.9405
PWC-Net	218.23	9.37	30.70	0.9388
RAFT - Small	277.88	0.99	30.92	0.9408
RAFT - Large	415.90	5.26	31.04	0.9411

Optical Flow Sensitivity Analysis. For our model, we employed the RAFT-Large model [45] to generate the flow maps $f_{0\to1}$ and $f_{1\to0}$. In this section, we examine changes in performance when using flow estimators [15,44,45] at varying levels of accuracy and computational cost. As shown in Table 4, while using RAFT-Large yields the highest fidelity, our model maintains robust performance even with lightweight estimators such as RAFT-Small. The performance drop is marginal (approx. 0.12dB), indicating that MGMVFI is not reliant on perfect flow estimation. This is further supported by the performance of our model across different scan methods presented below (Table 5), indicating that the SSM itself possesses inherent motion-modeling capabilities. Furthermore, utilizing a

smaller flow backbone significantly reduces the overall inference latency, offering a flexible trade-off for resource-constrained applications.

Table 5: Comparison of MGS against previous scanning methods

Scan method	$S \uparrow$	Vimeo-90K [46]		Hard [4]	
		PSNR	SSIM	PSNR	SSIM
Sequential	0.15	35.39	0.9791	29.45	0.9336
Spiral	0.12	35.17	0.9781	28.83	0.9302
Z-scan	0.29	35.50	0.9792	29.79	0.9335
Interleaved	0.35	36.56	0.9817	30.88	0.9390
Hilbert Curve	0.33	36.45	0.9813	30.57	0.9376
Ours (MGS)	0.46	36.67	0.9820	31.04	0.9411

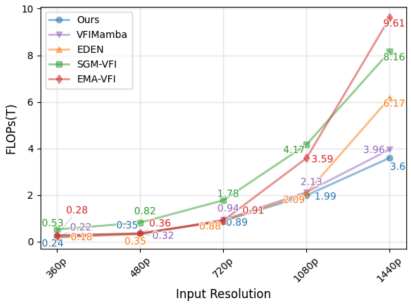

Fig. 5: Computation comparison on different input resolution

Scan methods. Table 5 evaluates our Motion-Guided Serialization (MGS) against conventional single-image (Sequential, Spiral, Z-scan) and video-specific (Interleaved, Hilbert Curve) scanning methods. To quantify the semantic continuity of the 1D serialized sequence, we introduce the Sequential Feature Similarity (S) metric (4) measured across 100 randomly sampled Vimeo90K images. Our results show that MGS significantly outperforms all baselines, particularly in complex motion scenarios. Specifically, MGS achieves a superior S of 0.46— a 31% improvement over the next best method (Interleaved). This semantic consistency directly translates into reconstruction accuracy, with MGS achieving a SotA PSNR of 31.04 dB on the SNU-FILM Hard benchmark. This highlights MGS's semantic consistency, simplifying the sequence modeling task for the S6 block and enabling more accurate local feature reconstruction.

Computation scalability. Finally, the scalability of the S6 architecture is evaluated by measuring computational load against input resolution in Figure 5. While competitive at lower scales, the MGMVFI block exhibits a clear advantage at high resolutions, where its linear scaling results in substantially lower resource consumption than other architectures.

6 Limitations

Despite MGMVFI's strong performance for SSM-based interpolation, several avenues for improvement remain. MGMVFI relies on external optical flow for MGS; flow errors can corrupt token ordering and propagate through the SSM. While contextual synthesis is beneficial, it does not address the core problem. Furthermore, MGS uses a single motion-adaptive serialization, which can underperform in scenes with multiple objects and small motions. We leave new research directions, such as joint flow refinement and multi-trajectory serialization, for future work.

7 Conclusion

We have proposed MGMVFI — Motion-Guided Mamba for Video Frame Interpolation — a motion-aware framework that redefines how S6 models handle VFI by moving beyond fixed scanning patterns. By implementing the MGS module, we leveraged optical flow to create a motion-adaptive serialization that preserved temporal coherence and streamlined interpolation. This was complemented by contextual synthesis to mitigate occlusions and an optimized EDFFN block that improved computational efficiency while restoring fine details. MGMVFI achieved state-of-the-art performance on challenging benchmarks, proving the efficacy of motion-guided serialization. We anticipate that this motion-guided serialization approach will provide a scalable foundation for broader video synthesis and understanding tasks.

Acknowledgements

This work was supported by Samsung Electronics Co., Ltd(IO251211-14315-01).

References

1. 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)
2. Cheng, X., Chen, Z.: Video frame interpolation via deformable separable convolution. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10607–10614 (2020)
3. Cheng, X., Chen, Z.: Multiple video frame interpolation via enhanced deformable separable convolution. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 7029–7045 (2021)
4. 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)
5. 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)
6. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5515–5524 (2016)
7. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: First conference on language modeling (2024)
8. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)
9. Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. Advances in neural information processing systems **34**, 572–585 (2021)
10. Guo, H., Guo, Y., Zha, Y., Zhang, Y., Li, W., Dai, T., Xia, S.T., Li, Y.: Mambairv2: Attentive state space restoration. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 28124–28133 (2025)

11. Guo, H., Li, J., Dai, T., Ouyang, Z., Ren, X., Xia, S.T.: Mambair: A simple baseline for image restoration with state-space model. In: European conference on computer vision. pp. 222–241. Springer (2024)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
13. 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)
14. 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)
15. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8981–8989 (2018)
16. Ibrahim, F., Liu, G., Wang, G.: A survey on mamba architecture for vision applications. arXiv preprint arXiv:2502.07161 (2025)
17. Jain, S., Watson, D., Tabellion, E., Poole, B., Kontkanen, J., et al.: Video interpolation with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7341–7351 (2024)
18. 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)
19. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: 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)
20. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. ACM Transactions on Graphics (TOG) **35**(6), 1–10 (2016)
21. Kong, L., Dong, J., Tang, J., Yang, M.H., Pan, J.: Efficient visual state space model for image deblurring. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12710–12719 (2025)
22. 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)
23. 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)
24. Li, B., Xue, K., Liu, B., Lai, Y.K.: Bbdm: Image-to-image translation with brownian bridge diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition. pp. 1952–1961 (2023)
25. Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., Qiao, Y.: Videomamba: State space model for efficient video understanding. In: European conference on computer vision. pp. 237–255. Springer (2024)
26. 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)
27. 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)

28. 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)
29. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: Proceedings of the IEEE international conference on computer vision. pp. 4463–4471 (2017)
30. Long, G., Kneip, L., Alvarez, J.M., Li, H.: Learning image matching by simply watching video (2016)
31. 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)
32. Lyu, Z., Chen, C.: Tlb-vfi: Temporal-aware latent brownian bridge diffusion for video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16260–16269 (2025)
33. Lyu, Z., Li, M., Jiao, J., Chen, C.: Frame interpolation with consecutive brownian bridge diffusion. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 3449–3458 (2024)
34. Ma, X., Ni, Z., Chen, X.: Tinyvim: Frequency decoupling for tiny hybrid vision mamba. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23519–23529 (2025)
35. Montgomery, C., Lars, H.: Xiph. org video test media (derf's collection). Online, https://media. xiph. org/video/derf **6**, 1 (1994)
36. Niklaus, S., Liu, F.: Context-aware synthesis for video frame interpolation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1701–1710 (2018)
37. 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)
38. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. In: Proceedings of the IEEE international conference on computer vision. pp. 261–270 (2017)
39. Park, J., Ko, K., Lee, C., Kim, C.S.: Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 109–125. Springer (2020)
40. 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)
41. Shi, Z., Xu, X., Liu, X., Chen, J., Yang, M.H.: Video frame interpolation transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17482–17491 (2022)
42. 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)
43. 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)
44. 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)
45. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)

46. 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)
47. 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)
48. 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)
49. Zhang, Z., Liu, A., Reid, I., Hartley, R., Zhuang, B., Tang, H.: Motion mamba: Efficient and long sequence motion generation. In: European Conference on Computer Vision. pp. 265–282. Springer (2024)
50. 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)
51. Zhou, C., Wu, T., Liu, W., Wu, X., Fu, Y.: Mvssm: Motion-aware visual state space model for efficient video deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4855–4865 (2026)
52. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 286–301. Springer (2016)