Flow-Guided Temporal Attention for Diffusion-Based Video Frame Interpolation

Ajay Vamsi Velamarthi
vamsiajay4@gmail.com
Somdyuti Paul
somdyuti@ai.iitkgp.ac.in

Department of Artificial Intelligence
Indian Institute of Technology
Kharagpur
Kharagpur, West Bengal, India

Abstract

Recent diffusion-based video frame interpolation (VFI) methods have demonstrated improved perceptual quality over traditional flow-based approaches. State-of-the-art results in VFI have recently been achieved by leveraging the temporal attention mechanism of diffusion transformers (DiTs). While temporal attention enables cross-frame token interactions, the motion information needs to be implicitly inferred from such token interactions, which can limit performance in the presence of large motion. Further, implicit motion modeling from token interactions becomes more challenging as the model's learning capacity reduces. To address these limitations, we introduce explicit optical flow guidance to the DiT used for VFI, building upon EDEN, a DiT-based VFI model. Specifically, we incorporate optical flow priors for providing local guidance through gated token modulation prior to temporal attention and global flow-aware conditioning with pooled flow embeddings to modulate input tokens of transformer blocks via adaptive layer normalization. Experimental results demonstrate improved interpolation quality across perceptual and temporal consistency metrics. Larger gains are observed under large-motion scenarios, with 7.57% and 10.02% FloLPIPS gains over the best competing model achieved on the DAVIS and SNU-FILM (Extreme) benchmarks. Notably, a smaller variant of our model with only 62.7M parameters trained with explicit motion priors outperforms the baseline EDEN with 118.3M parameters, highlighting the efficacy of explicit motion priors as an inductive bias for diffusion-based VFI. Code is available at https://github.com/AjayVamsiV/EDENFlow.

1 Introduction

Video frame interpolation (VFI) aims to synthesize intermediate frames between two input frames and plays an important role in applications such as frame rate upsampling, slow-motion generation, video restoration, and video editing. A key challenge in VFI is accurately modeling temporal correspondence in the presence of large motion, occlusions, motion blur, and nonlinear scene dynamics.

1.1 Limitations of Existing Approaches

Traditional VFI approaches primarily rely on optical flow estimation and frame warping [7, 13]. These methods explicitly estimate pixel-level motion between neighboring frames and

© 2026. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

synthesize intermediate frames through warping and refinement. However, flow-based methods often struggle with inaccurate motion estimation in the presence of occlusions, lighting changes, aperture effects and complex motion patterns, leading to ghosting artifacts and temporal inconsistencies. Recent advances in diffusion models [10, 27] and latent diffusion frameworks [25] have significantly improved image and video generation quality. Diffusion transformers (DiTs) [23] further enabled global context modeling and demonstrated strong scalability. Building upon these advances, recent VFI methods such as LDMVFI [6], LBBDM [22], and EDEN [35] generated intermediate frames directly in latent space. Despite these improvements, existing diffusion-based VFI methods still rely primarily on implicit motion modeling through feature similarity during temporal attention computation, which becomes challenging under large displacement, occlusions, and fast nonlinear motion. In particular, diffusion models with limited learning capacity struggle with implicit motion modeling for latent generation.

1.2 Our Contributions

We address the limitations of diffusion-based VFI by bridging the gap between flow-based interpolation and diffusion-based frame generation. Unlike traditional flow-based VFI that warps frames in pixel space, diffusion-based VFI operates in latent space where pixel-level warping is not meaningful. Thereby, we propose using flows as a motion prior to guide the diffusion process through token modulation. Specifically, building upon the diffusion-based VFI introduced in EDEN [35], we propose flow-guided temporal attention, where forward and backward optical flows modulate boundary frames at the local token-level via a learned gating mechanism prior to temporal attention. We further introduce global flow-aware adaptive layer normalization (AdaLN) conditioning for scene-level motion modulation. This dual-path flow conditioning acts as an effective inductive bias for the diffusion model responsible for latent generation.

Our main contributions are:

- Flow-guided temporal attention for diffusion-based VFI.
- A gated flow-conditioned token modulation for motion-aware temporal aggregation.
- Dual-path flow conditioning with local and global motion guidance.
- Improved interpolation quality, temporal consistency, and training convergence, especially in the presence of large motion.

2 Related Work

2.1 Optical Flow Estimation

Optical flow [11, 20] remains one of the most effective techniques for explicit motion estimation in video understanding and frame interpolation. The advantages of data-driven deep learning for overcoming issues of traditional flow estimation techniques such as occlusion handling, aperture problem, brightness constancy and small motion assumptions, etc. were successfully demonstrated by models such as FlowNet [8] and PWC-Net [29]. Subsequently, RAFT [30] employed a recurrent-CNN based optical flow estimator to learn multi-scale correlations between all pixel pairs, outperforming prior methods. Transformer self-attention was used to learn optical flows through global feature matching in GMFlow [32], while a

transformer-based cost volume encoder was introduced for flow estimation in FlowFormer [12].

2.2 Flow-Based Video Frame Interpolation

Traditional VFI approaches primarily rely on optical flow estimation and frame warping for temporal correspondence modeling. Super SloMo [13] estimated bidirectional optical flow and visibility maps for intermediate frame synthesis, while DAIN [3] used depth-aware flow projection to better handle occlusions and large motion. More recent flow-based VFI methods have further improved interpolation quality through advanced correspondence learning. VFIFormer [19] introduced transformer-based long-range pixel correlation modeling for improved temporal correspondence. ST-MFNet [5] proposed spatio-temporal multi-flow estimation for handling complex motion patterns. AMT [16] constructed bidirectional correlation volumes for all pixel pairs to interpolate frames. SGM-VFI [17] addressed large motion through sparse global matching. VFIMamba [33] explored state space models for efficient video frame interpolation.

2.3 Diffusion-Based Video Frame Interpolation

The technology for image and video generation has undergone a paradigm shift with the introduction of diffusion models [10], with subsequent advancements improving photorealism [7], high-resolution synthesis [25], and inference speed [27]. While initial diffusion models relied on a convolutional U-Net [26] backbone, diffusion transformers (DiTs) [23] demonstrated improved scalability between generation quality and model size.

Motivated by these advances, LDMVFI [6] and LBBDM [22] employed U-Net-based [26] diffusion models to generate intermediate frames directly in latent space, while EDEN [35] proposed a diffusion transformer using latent tokenization, temporal attention, and motion conditioning using boundary frame difference embedding for large-motion VFI. TLB-VFI [21] used 3D-wavelet gating and a temporal-aware autoencoder for implicit motion modeling; [14] adapted image-to-video diffusion models via cross-attention between boundary frame features and flow features; MoMo [15] employed a dedicated motion diffusion model to estimate bidirectional optical flow for pixel-space warping. Although [14] and [15] also use explicit optical flow, unlike existing approaches, the proposed approach uses flow to condition boundary-frame tokens via gated flow modulation.

3 Proposed Method

3.1 Overview

We build upon the EDEN [35] diffusion transformer framework for video frame interpolation and introduce explicit motion-aware temporal conditioning using optical flow. Given two boundary frames I_b and I_f, the objective is to synthesize the intermediate frame I_t through latent diffusion. While EDEN relies primarily on implicit motion reasoning through feature similarity and temporal attention, our method injects explicit motion priors into temporal attention using optical flow.

Specifically, we incorporate flow information in two complementary ways:

- global flow conditioning through adaptive layer normalization (AdaLN) modulation,

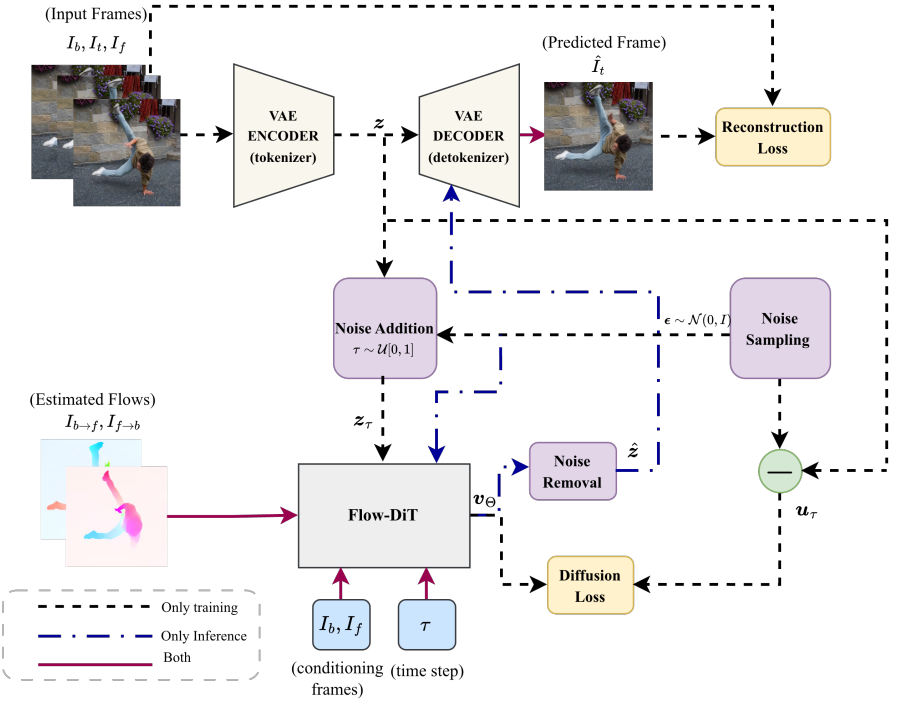

Figure 1: **Overview of EDENFlow.** Latent diffusion framework with flow-guided conditioning for video frame interpolation.

- local flow-guided temporal attention using gated token modulation.

The overall architecture, subsequently referred to as EDENFlow, is illustrated in Fig. 1. During training, a transformer-based variational autoencoder (VAE) takes the two boundary frames I_b, I_f and the intermediate frame I_t to produce a tokenized latent representation z, which it decodes and detokenizes to reconstruct the target intermediate frame $\hat{I}_t$ [33]. A combination of an L_1 reconstruction loss and a perceptual LPIPS [34] loss between the ground truth intermediate frame I_t and the reconstructed frame $\hat{I}_t$ is minimized to train the VAE. The latent is perturbed through the forward diffusion process to obtain a noisy latent z_τ. The DiT receives the noisy latent together with the boundary frames and estimated optical flow priors as its inputs to predict the target velocity representation at each diffusion step by minimizing the mean squared error between the predicted velocity, v_Θ (where Θ represents the DiT's parameters) and ground-truth velocity u_τ using a linear transport path formulation [9]. During inference, the model starts from Gaussian noise and progressively integrates the predicted velocity field to obtain the final denoised latent. The denoised latent is then passed through the VAE decoder to generate the intermediate frame as output. These steps are elaborated in the remainder of this section.

3.2 Optical Flow Estimation

We rely on learned optical flow estimation to obtain motion priors for conditioning the DiT model for producing the denoised latent during reverse diffusion. To minimize computa-

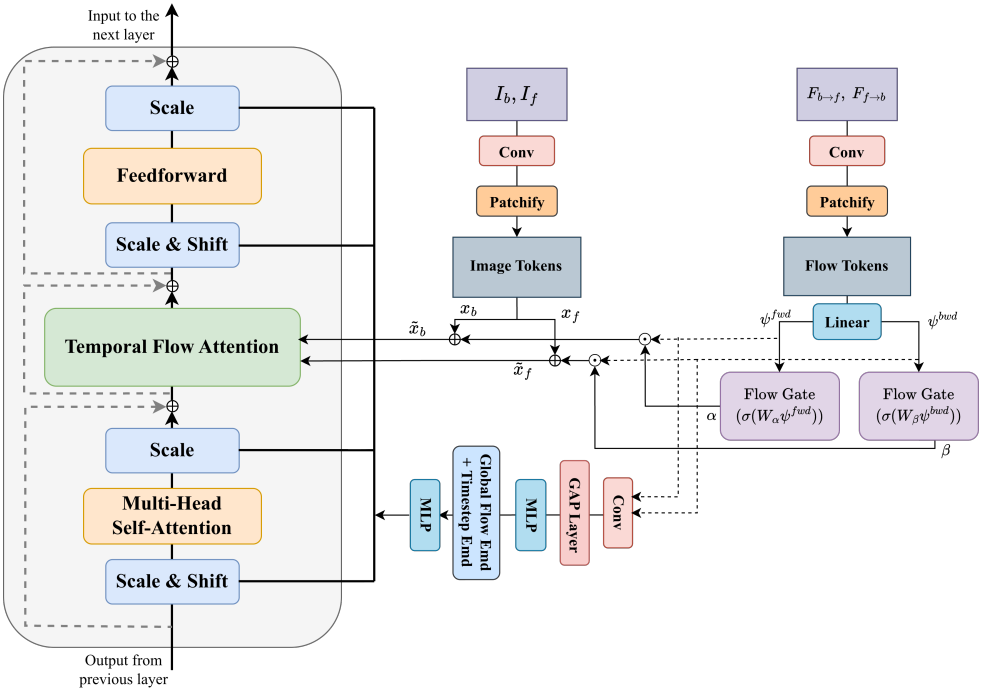

Figure 2: **Proposed flow-guided diffusion transformer block.** The proposed architecture introduces local token-level flow modulation and global flow-aware conditioning into temporal attention using forward and backward optical flow.

tional overhead, we employ RAFT [30], a fast deep neural network having 5M parameters designed for accurate optical flow estimation.

Given the two boundary frames, we estimate both forward and backward optical flows as follows:

$$F_{b\to f} = \Phi(I_b, I_f), \quad F_{f\to b} = \Phi(I_f, I_b), \tag{1}$$

where $\Phi(\cdot)$ denotes the pretrained RAFT estimator.

Using both forward and backward flow provides complementary motion information and improves robustness in challenging regions such as occlusions, disocclusions, and motion boundaries.

3.3 Input Tokenization

For computation of temporal attention, the two boundary frames are first tokenized as follows:

$$x_b = \mathrm{LN}\big(P(C(I_b)) + E_{\mathrm{pos}}\big), \quad x_f = \mathrm{LN}\big(P(C(I_f)) + E_{\mathrm{pos}}\big), \tag{2}$$

where $C(\cdot)$ denotes convolutional projection, $P(\cdot)$ denotes patch tokenization, E_{pos} denotes positional embedding and $\mathrm{LN}(\cdot)$ denotes layer normalization [1].

The positional embedding preserves spatial layout after flattening the image into tokens, which is important for motion correspondence learning inside attention layers. Layer nor-

malization further stabilizes feature distributions and reduces scale mismatch between image and flow tokens during temporal attention.

To enable interaction between the image tokens and flow tokens, the estimated optical flow maps are also tokenized using convolutional projection, patch tokenization, positional embedding and layer normalization. The tokenized representations are then subjected to a linear projection:

$$\psi^{\text{fwd}} = \text{LN}\left(W_p\left(\text{LN}\left(P(C(F_{b\to f}))+E_{\text{pos}}\right)\right)\right), \quad \psi^{\text{bwd}} = \text{LN}\left(W_p\left(\text{LN}\left(P(C(F_{f\to b}))+E_{\text{pos}}\right)\right)\right), \quad (3)$$

where W_p denotes a learnable linear projection layer. This pipeline produces flow tokens compatible with the image token embedding space to enable their effective interaction, while the additional linear projection further refines motion features before temporal attention.

3.4 Flow-Guided Gating Mechanism

Directly injecting flow features can be harmful when flow estimates are noisy or uncertain. To address this issue, we introduce two learnable gating functions that adaptively control the contribution of flow tokens:

$$\alpha = \sigma(W_\alpha \psi^{fwd}), \quad \beta = \sigma(W_\beta \psi^{bwd}), \quad (4)$$

where W_α and W_β are learnable weights, $\sigma(\cdot)$ denotes the sigmoid activation function and $\alpha,\ \beta \in [0,1]$ are the two learnable gating functions that control how much of the flow information is used to provide motion guidance for a particular token.

The motion-enhanced temporal context tokens are then computed as:

$$\tilde{x}_b = x_b + \alpha \odot \psi^{\text{fwd}}, \quad \tilde{x}_f = x_f + \beta \odot \psi^{\text{bwd}}, \quad (5)$$

where $\odot$ denotes element-wise multiplication. This gating mechanism enables the model to emphasize reliable motion cues while suppressing inconsistent or noisy flow regions.

3.5 Flow-Guided Temporal Attention

Following EDEN [33], temporal attention is performed using local temporal neighborhoods instead of global token interaction. Specifically, the motion enriched tokens $\tilde{x}_b$ and $\tilde{x}_f$ are reorganized into local 2×2 spatial neighborhoods. For each token $\bar{z}^i_\tau$, temporal attention is computed over:

$$\mathcal{T}_i = \left[\mathcal{N}_b(i), \bar{z}^i_\tau, \mathcal{N}_f(i)\right], \quad (6)$$

where $\bar{z}^i_\tau$ is the ith token of the intermediate latent representation $\bar{z}_\tau$ processed by prior transformer layers that is available as input to the temporal attention layer; $\mathcal{N}_b(i)$ and $\mathcal{N}_f(i)$ denote the corresponding 2×2 neighboring tokens from $\tilde{x}_b$ and $\tilde{x}_f$, respectively. Thus, each token attends to a compact set of nine tokens consisting of four neighboring tokens from $\tilde{x}_b$, the current token $\bar{z}^i_\tau$ and four neighboring tokens from $\tilde{x}_f$.

3.6 Global Flow Conditioning

EDEN [33] performs AdaLN using a difference embedding to provide global motion information between the two boundary frames. This difference embedding is calculated using the cosine similarity of the boundary frames, normalized using the mean and standard deviation statistics of the cosine similarity of all boundary frame pairs at the dataset level. However, such similarity-based embeddings provide only coarse motion cues and cannot explicitly capture motion direction or spatial correspondence, unlike optical flow which explicitly models pixel-level motion trajectories between frames. To provide richer scene-level motion information, we therefore construct a global flow descriptor using both forward and backward optical flow representations for transformer conditioning.

Spatially pooled forward and backward flow tokens are first aggregated as:

$$g^{\text{fwd}} = \text{GAP}(C(\psi^{\text{fwd}})), \quad g^{\text{bwd}} = \text{GAP}(C(\psi^{\text{bwd}})), \tag{7}$$

where $C(\cdot)$ denotes convolutional refinement and GAP denotes global average pooling.

The two directional embeddings are then combined as:

$$g = \frac{1}{2}\left(g^{\text{fwd}} + g^{\text{bwd}}\right). \tag{8}$$

A multilayer perceptron projects the global flow representation g into the same embedding space as the diffusion timestep embedding e_τ at time step τ:

$$c = \text{MLP}(g) + e_\tau. \tag{9}$$

The combined embedding c is used to condition transformer layers through AdaLN, following diffusion transformer architectures [23]. AdaLN applies learned scale and shift parameters to normalized transformer features conditioned on c, enabling the model to dynamically modulate feature activations according to both diffusion timestep and global motion information. Specifically, given an input feature x, AdaLN performs:

$$\text{AdaLN}(x,c) = \gamma(c)\cdot\text{LN}(x) + \delta(c), \tag{10}$$

where $\text{LN}(\cdot)$ denotes layer normalization, while $\gamma(\cdot)$ and $\delta(\cdot)$ are learned scale and shift functions generated from conditioning embedding c. The embedding c is also used to learn an additional scale function $\eta(c)$. This conditioning mechanism enables global motion-aware modulation of all transformer blocks.

3.7 Transformer Block

The first transformer block of the DiT receives the noisy latent z_τ as input at diffusion time step τ, which is obtained as:

$$z_\tau = (1-\tau)z + \tau\varepsilon, \quad \varepsilon \sim \mathcal{N}(0,I), \quad \tau \sim \mathcal{U}[0,1]. \tag{11}$$

Each diffusion transformer block consists of self-attention, flow-guided temporal attention, and a feed-forward network:

$$\bar{z}_\tau = \bar{z}_\tau + \eta(c)\cdot\text{SA}(\text{AdaLN}(\bar{z}_\tau, c)), \tag{12}$$

$$\bar{z}_\tau = \bar{z}_\tau + \text{TempAttn}(\tilde{x}_b, \bar{z}_\tau, \tilde{x}_f), \tag{13}$$

$$\bar{z}_\tau = \bar{z}_\tau + \eta(c)\cdot\text{MLP}(\text{AdaLN}(\bar{z}_\tau, c)), \tag{14}$$

where $\bar{z}_\tau$ denotes the intermediate latent representation within the transformer block, initialized as the noisy latent z_τ at the input to the first block, $\text{SA}(\cdot)$ denotes self-attention and $\text{TempAttn}(\cdot)$ denotes the proposed flow-guided temporal attention operating over the local temporal neighborhoods described in Eq. 6. MLP represents the positionwise feedforward neural network with shared parameters over all tokens.

The parameters of the DiT, denoted by Θ, are updated by minimizing $\|v_\Theta - u_\tau\|_2^2$, where $u_\tau = \varepsilon - z$ [9].

4 Experiments

4.1 Experimental Setup

Datasets. Following recent work on diffusion-based VFI [35], we trained on the large-scale LAVIB benchmark [28], which contains approximately 284,000 video clips with a resolution of 1296×1296. For evaluation, we used DAVIS [24], DAIN-HD [2], and SNU-FILM [6], which contain challenging

large-motion and nonlinear motion scenarios. DAVIS contains 2849 triplets at resolution 480×854, while DAIN-HD consists of high-resolution interpolation samples at 544×1280 resolution with large motion and complex scene dynamics. SNU-FILM further provides multiple motion subsets with varying difficulty levels ranging from Easy to Extreme at multiple resolutions.

Architecture. The proposed EDENFlow model differs from EDEN [35] only through the addition of flow-conditioning modules integrated into global temporal conditioning and local temporal attention, while the architectures of VAE and DiT as well as the diffusion formulation were kept identical to EDEN to ensure a fair comparison. The boundary frames as well as the forward and backward flow fields were projected using convolutional layers with kernel size of 16×16 and stride of 16 into the transformer embedding space of dimension d. The convolutional weights in the tokenization step are shared within each modality, i.e., the boundary frames x_b and x_f share one set of weights, while the flow priors $F_{b\to f}$ and $F_{f\to b}$ share another set of weights. The EDENFlow DiT model has 139.4M parameters, which is a modest increase over 118.3M parameters originally present in EDEN's DiT. To reduce learning capacity, EDENFlow was scaled down by reducing the embedding dimension d of the DiT from 768 to 576 and its MLP ratio from 4.0 to 2.0, yielding a smaller variant referred to as EDENFlow-Small having 62.7M parameters. Correspondingly, EDEN was scaled by reducing d from 768 to 576, while keeping the same MLP ratio (4.0) as the full capacity model to achieve a comparable number of parameters. The resulting model, called EDEN-Small, has 66.7M parameters. Detailed architecture configurations for the EDENFlow and EDENFlow-Small are summarized in Table 1.

Table 1: Architecture configuration of EDENFlow and EDENFlow-Small. (* denotes the configurations for EDENFlow-Small wherever they differ from EDENFlow.)

Component	EDENFlow
Latent dimension	16
Transformer dim (d)	768 / 576*
Transformer depth	12
Attention heads	12
MLP ratio	4.0 / 2.0*
VAE encoder depth	4
VAE decoder depth	4
VAE Parameters	77.37M
DiT Parameters	139.4M / 62.7M*

For optical flow estimation, we employed a pretrained RAFT [30] model with approximately 5M parameters to estimate both forward and backward optical flow. RAFT was chosen as it is one of the most widely used models for VFI in recent times, and provides a favorable latency vs. performance tradeoff, as shown in Section 4.8.3, enabling relatively low overhead motion estimation during training and inference. The weights of the RAFT model were not updated during training.

Training Details. We used the pretrained VAE weights provided by EDEN [35] and trained only the DiT, as the VAE architecture is identical for EDEN and EDENFlow. All models were trained using AdamW [18] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay 1×10^{-4}, and learning rate 1×10^{-4}. A cosine learning-rate scheduler with 100 warm-up steps was used during training.

Following EDEN, we employed joint multi-step and multi-resolution training to improve robustness across different motion scales and scene complexities, where multistep training used intervals of 1, 3 and 5 frames between boundary frames, while for multi-resolution training, frames were first cropped to maintain the aspect ratio of the target resolution, and then downsampled to the target resolution. Four target resolutions (256×448, 480×854, 544×1280 and 720×1280) were used. During training, each batch was composed of frames at a single resolution and a batch size of 16 was used for batches that comprised 544×1280 and 720×1280 samples, while a batch size of 32 was used for batches comprising the two lower resolution samples during training. All models were trained on NVIDIA H100 GPUs with 80 GB VRAM.

Evaluation Metrics. We evaluated reconstruction quality, perceptual fidelity, and temporal consistency using complementary metrics. PSNR↑ and SSIM↑ [31] measure pixel-level error and structural similarity, respectively. LPIPS↓ [34] evaluates perceptual quality using deep feature similarity, while FloLPIPS↓ [9] additionally incorporates optical flow alignment for temporal consistency evaluation. During inference, two denoising steps were used to produce the interpolated frames as in [35].

Table 2: Comparison of EDEN and EDENFlow (Ours).

Metric	DAVIS (EDEN / Ours)	DAIN-HD (EDEN / Ours)	SNU Easy (EDEN / Ours)	SNU Medium (EDEN / Ours)	SNU Hard (EDEN / Ours)	SNU Extreme (EDEN / Ours)
PSNR↑	25.9875 / **26.4870**	23.7178 / **24.5073**	38.1318 / **38.4868**	33.9740 / **34.5181**	28.9834 / **29.5783**	24.4467 / **24.9839**
SSIM↑	0.8509 / **0.8615**	0.9112 / **0.9197**	0.9867 / **0.9873**	0.9716 / **0.9734**	0.9222 / **0.9262**	0.8456 / **0.8525**
LPIPS↓	0.1052 / **0.0916**	0.1968 / **0.1711**	0.0154 / **0.0144**	0.0282 / **0.0260**	0.0583 / **0.0526**	0.1194 / **0.1043**
FloLPIPS↓	0.1446 / **0.1248**	0.2892 / **0.2736**	0.0268 / **0.0235**	0.0551 / **0.0457**	0.1084 / **0.0902**	0.1971 / **0.1689**

Table 3: Comparison of EDEN-Small (EDEN-S) and EDENFlow-Small (Ours-S).

Metric	DAVIS (EDEN-S / Ours-S)	DAIN-HD (EDEN-S / Ours-S)	SNU Easy (EDEN-S / Ours-S)	SNU Medium (EDEN-S / Ours-S)	SNU Hard (EDEN-S / Ours-S)	SNU Extreme (EDEN-S / Ours-S)
PSNR↑	25.8608 / **26.3745**	23.3263 / **24.1708**	37.9403 / **38.1712**	33.7243 / **34.3668**	28.8164 / **29.4392**	24.3438 / **24.8743**
SSIM↑	0.8480 / **0.8591**	0.9036 / **0.9156**	0.9864 / **0.9866**	0.9707 / **0.9729**	0.9207 / **0.9253**	0.8442 / **0.8507**
LPIPS↓	0.1089 / **0.0962**	0.2116 / **0.1870**	0.0159 / **0.0156**	0.0294 / **0.0270**	0.0612 / **0.0550**	0.1233 / **0.1096**
FloLPIPS↓	0.1486 / **0.1307**	0.3001 / **0.2861**	0.0273 / **0.0254**	0.0561 / **0.0482**	0.1116 / **0.0972**	0.2021 / **0.1792**

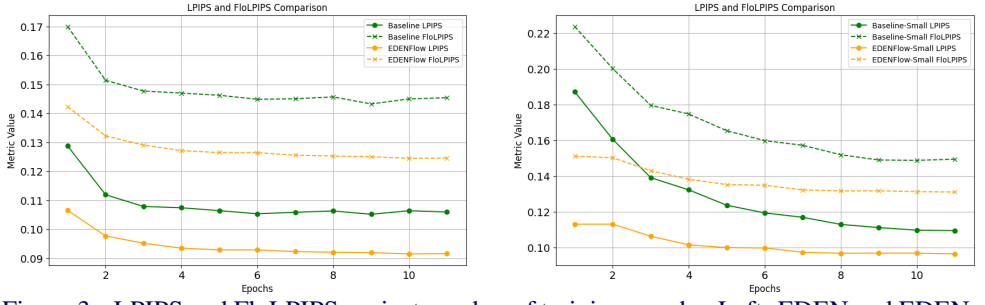

Figure 3: LPIPS and FloLPIPS against number of training epochs. Left: EDEN and EDENFlow. Right: EDEN-Small and EDENFlow-Small.

4.2 Comparison with EDEN Baseline

Table 2 presents the comparison between EDEN and the proposed EDENFlow method using full-capacity models trained under the same settings as EDEN for fair evaluation. The proposed flow-guided conditioning consistently improved PSNR, SSIM, LPIPS and FloLPIPS across all the benchmarks used for evaluation. The gains were particularly pronounced in challenging large-motion scenarios such as SNU-FILM Extreme, demonstrating the effectiveness of explicit motion priors for improved temporal correspondence modeling.

4.3 Comparison at Reduced Model Capacity

Table 3 presents the comprehensive evaluation of the EDEN-Small and EDENFlow-Small models trained under identical settings. These results reveal that EDENFlow-Small, with only 62.7M DiT parameters, not only outperformed the EDEN-Small model with 66.7M parameters, but also outperformed the full-capacity EDEN baseline with 118.3M DiT parameters as reported in Table 2 across all benchmarks and metrics. This confirms that the proposed flow guidance acts as an effective inductive bias for VFI, and that the observed gains are not solely attributable to the additional parameters introduced by EDENFlow. The integration of this inductive bias thus allows the model size to be reduced for applications that demand lightweight architectures, without sacrificing the quality of the results obtained.

We evaluated LPIPS and FloLPIPS on the DAVIS dataset periodically during training as shown in Fig. 3. Fig. 3 shows that both EDENFlow and EDENFlow-Small converged faster and consistently maintained lower LPIPS and FloLPIPS compared to their EDEN counterparts. The progression

of LPIPS and FloLPIPS across training epochs shows that while the baseline EDEN and EDEN-Small models gradually improve by implicitly inferring motion from token interactions, the flow-guided models leveraged explicit motion priors to achieve superior results from the outset of training. Since motion priors are available to the DiT, learning can focus on the generative denoising task rather than motion modeling, which accelerates convergence.

4.4 Effect of Motion Magnitude

We analyzed the impact of explicit flow guidance across varying motion magnitudes, by evaluating the performance gap between EDEN and EDENFlow as well as EDEN-Small and EDENFlow-Small on different subsets of the SNU-FILM benchmark. Fig. 4 plots the difference of the average LPIPS between EDEN and EDENFlow as well as between EDEN-Small and EDENFlow-Small against the interval between the boundary frames. As shown in Fig. 4, the performance gain provided by flow-guided conditioning increased consistently with the increase of motion magnitude from the Easy subset to the Extreme subset. The improvement for FloLPIPS was more significant than that for LPIPS, indicating that explicit optical flow guidance becomes increasingly beneficial for preserving temporal consistency under large motion and challenging frame transitions. These results further support the efficacy of flow-guided temporal conditioning for motion-aware diffusion-based VFI.

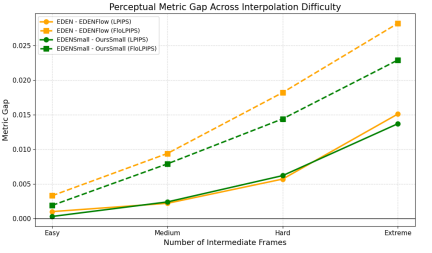

Figure 4: Performance gap between EDENFlow and EDEN (from Table 2) on SNU-FILM subsets with increasing motion difficulty.

4.5 Comparison with State-of-the-Art Methods

Table 4 compares our method against recent flow-based correspondence learning [5, 16, 17, 19, 33] and diffusion-based VFI approaches [6, 22, 35] using perceptual and temporal consistency metrics. The EDEN [35] results reported in Table 4 correspond to the end-to-end DiT and VAE decoder model weights provided by the official implementation of the model, which are obtained by first pre-training the complete VAE and the DiT individually, and then jointly fine-tuning the VAE decoder and the DiT, using random noise as input to the pretrained DiT rather than the latent from the VAE encoder combined with random noise. The minimization objective for this end-to-end fine-tuning is a combined loss term comprising L_1 loss, perceptual loss [34] and an L_2 regularization term on $\hat{z}$, where the L_2 regularization term was weighted by 0.001. EDENFlow was initialized using the same pretrained EDEN weights and further fine-tuned by jointly optimizing the DiT and the VAE decoder following the same protocol.

EDENFlow consistently improved LPIPS and FloLPIPS across multiple benchmarks and metrics, as reported in Table 4, demonstrating that explicit motion priors were effective even when starting from a strong end-to-end optimized diffusion baseline. The EDENFlow model using RAFT as the optical flow estimator improved average FloLPIPS by 7.57% and 10.02% on the challenging DAVIS and SNU-FILM (Extreme) benchmarks, respectively, over EDEN, the previous state-of-the-art diffusion-based VFI method. On DAIN-HD, EDENFlow achieved the second-lowest FloLPIPS, which degraded marginally by 0.73% compared to LBBDM [22] which achieved the lowest FloLPIPS on this benchmark. On SNU-FILM (Easy), a benchmark comprising nearly static scenes, where EDEN's large motion modeling is less suited, LBBDM achieved the best FloLPIPS performance. This performance gap persisted in EDENFlow, whose FloLPIPS was 30.86% higher compared to LBBDM on this dataset. Although EDENFlow's LPIPS was marginally higher than that of EDEN on SNU-FILM (Easy), its lower FloLPIPS indicates improved temporal consistency even in low-motion scenarios. Furthermore, EDENFlow consistently outperformed EDEN across all metrics on SNU-FILM (Easy) when trained from scratch, as shown in Tables 2 and 3. This suggests that the slight LPIPS regression observed in Table 4 is specific to the end-to-end fine-tuned setting, while the temporal-consistency benefit of

Table 4: Comparison with existing VFI methods on DAVIS, DAIN-HD544p, and SNU-FILM benchmarks.

	DAVIS		DAIN-HD544p		SNU-FILM (Easy)		SNU-FILM (Extreme)	
Method	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓
VFIFormer [19]	0.1370	0.1905	0.1604	0.2764	0.0176	0.0290	0.1463	0.2361
ST-MFNet [5]	0.1232	0.1674	0.1577	0.2507	0.0210	0.0336	0.1302	0.2192
AMT [16]	0.1091	0.1523	0.1479	0.2697	0.0191	0.0309	0.1205	0.2054
SGM-VFI [17]	0.1140	0.1571	0.1423	0.2471	0.0178	0.0292	0.1205	0.1936
VFIMamba [33]	0.1084	0.1486	0.1426	0.2471	0.0179	0.0299	0.1154	0.1892
LDMVFI [6]	0.1095	0.1497	0.1490	0.2378	0.0145	0.0207	0.1227	0.1894
LBBDM [23]	0.0963	0.1313	0.1471	0.2319	**0.0120**	**0.0162**	0.1101	0.1696
EDEN [35]	0.0854	0.1176	0.1444	0.2479	0.0128	0.0220	0.0968	0.1627
Ours (RAFT)	0.0799	0.1087	0.1372	0.2336	0.0135	0.0212	0.0912	0.1464
Ours (GMFlow)	**0.0795**	**0.1078**	**0.1351**	**0.2277**	0.0135	0.0212	**0.0903**	**0.1435**

Table 5: Computational complexity and inference time comparison across different resolutions. Rows 2 and 4 report RAFT time (ms) + EDENFlow/EDENFlow-Small's DiT time (ms).

	256×448		480×854		544×1280		720×1280	
Model	GFLOPs	Runtime (ms)	GFLOPs	Runtime (ms)	GFLOPs	Runtime (ms)	GFLOPs	Runtime (ms)
EDEN	32.67	15.72	120.30	61.49	205.44	112.67	282.01	157.58
EDENFlow	50.50	29.91+20.50	184.79	134.72+79.12	313.71	235.5+145.31	428.50	349.98+203.42
EDEN-Small	16.25	12.92	68.73	50.87	117.98	92.23	162.67	132.99
EDENFlow-Small	26.88	29.91+15.87	98.82	134.72+61.81	168.50	235.5+112.31	231.02	349.98+163.8

explicit flow guidance persists across different motion regimes. The experimental results in Table 4 thus suggest that the motion priors for conditioning diffusion models are most effective in the presence of large and complex motions as observed in DAVIS and SNU-FILM (Extreme). The gains attained by EDENFlow improve further when more advanced optical flow estimators are employed, as shown by the last row of Table 4. Using GMFlow as the optical flow estimator, EDENFlow improves over the best competing method by 8.33%, 1.81%, and 11.8% on DAVIS, DAIN-HD and SNU-FILM (Extreme), respectively, at the cost of higher computational overhead as discussed in Section 4.8.3. These gains affirm that EDENFlow provides robust performance irrespective of the optical flow estimator used, with larger improvements observed for flow estimators with more advanced motion modeling capabilities.

4.6 Computational Complexity

We compared the computational complexity and inference runtime of EDEN and the proposed flow-guided variants across multiple resolutions. Table 5 reports the GFLOPs per denoising step and the inference time per sample measured on an NVIDIA L40 GPU with 46 GB VRAM with a batch size of 16. The reported GFLOPs do not include the computations performed by the pretrained flow estimators to enable a fair comparison of the additional computational overhead introduced for flow-based dual conditioning in the DiT. However, to provide a practical estimate of the total inference times across different resolutions, we report the inference times of the optical flow estimators along with the total time taken by the proposed flow-guided models to perform two denoising steps to produce the interpolated frames in the second and fourth rows of Table 5.

EDENFlow and EDENFlow-Small require 3.2-3.5× and 3.5-3.9× higher total inference time than their corresponding EDEN baselines, respectively, across the evaluated resolutions, primarily due to

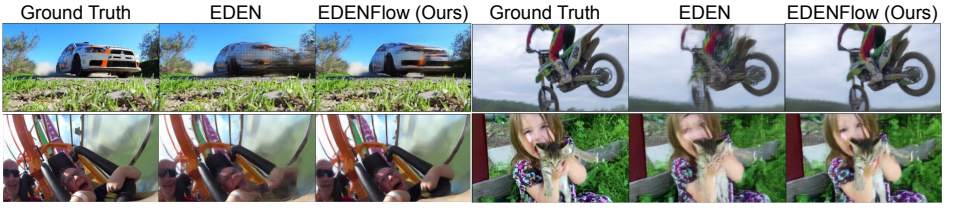

Figure 5: Qualitative comparison between EDEN and the proposed method on the DAVIS dataset.



Figure 6: Qualitative comparison between EDEN and the proposed method on the SNU-FILM (Extreme) dataset.

the additional RAFT flow estimation overhead. The additional computational overhead to integrate motion priors for local and global conditioning also increases the complexity of EDENFlow relative to the baseline. Although, the DiT of EDENFlow-Small has fewer GFLOPs than EDEN, it is marginally slower in terms of inference time due to the introduction of additional convolutional layers at the flow tokenization step. Thus, the improved VFI performance in terms of perceptual quality and temporal consistency is achieved at the cost of increased inference latency, which highlights the limitation of the proposed approach.

4.7 Qualitative Results

The interpolated frames obtained using the EDEN and EDENFlow models reported in Table 4 are shown in Figs. 5, 6 and 7 for samples from the DAVIS, SNU-FILM (Extreme) and DAIN-HD datasets, respectively. These qualitative results strongly suggest that the proposed flow-guided diffusion transformer produces sharper reconstructions, cleaner object boundaries, and improved temporal consistency under large-motion scenarios, thereby corroborating the quantitative gains reported.

4.8 Ablation Studies

We conducted ablation studies to assess the impact of the different components of the proposed EDENFlow model as described next. All ablation experiments were conducted by training and evaluating at the resolution of 256×448 (by downsampling the DAVIS dataset) for computational efficiency.

4.8.1 Flow Integration Strategy

Since motion information can be used to condition temporal attention in different ways, identifying an effective integration strategy is important for balancing temporal correspondence modeling and reconstruction quality. To better understand how motion guidance should be incorporated into the temporal attention of the diffusion transformer, we explored multiple architectural variants as summarized below:



Figure 7: Qualitative comparison between EDEN and the proposed method on the DAIN-HD dataset.

Table 6: Evaluation of different flow integration strategies on DAVIS.

Method	PSNR	SSIM	LPIPS	FloLPIPS	L_1
EDEN	28.129	0.8899	0.0594	0.0823	9416
Joint Temp-Attn	28.163	0.8912	0.0592	0.0811	9372
Two-Stage Temp-Attn	28.200	0.8916	0.0584	0.0796	9317
Flow-Gating	**28.371**	**0.8949**	**0.0555**	**0.0761**	**9069**

1. **Joint Temporal Attention.**
   Flow fields are tokenized using Eq. 3 and concatenated to the two boundary frames. Temporal attention is jointly computed over the image and flow tokens by expanding the neighborhood defined in Eq. 6 to 17 tokens, comprising four tokens each from x_b, ψ^{fwd}, x_f and ψ^{bwd}, respectively along with the current token from $\bar{z}_\tau$:

$$\bar{z}_\tau \to \text{SA}(z_\tau) \to \text{TempAttn}\left(x_b, \psi^{\text{fwd}}, \bar{z}_\tau, x_f, \psi^{\text{bwd}}\right) \to \text{MLP}(\bar{z}_\tau). \tag{15}$$

2. **Two-Stage Temporal Attention.**
   Motion aggregation and image aggregation are performed sequentially using two temporal attention stages:

$$\bar{z}_\tau \to \text{SA}(z_\tau) \to \text{TempAttn}\left(\psi^{\text{fwd}}, \bar{z}_\tau, \psi^{\text{bwd}}\right) \to \text{TempAttn}(x_b, \bar{z}_\tau, x_f) \to \text{MLP}(\bar{z}_\tau). \tag{16}$$

3. **Proposed Gated Flow-Guided Attention.**
   Forward and backward flow tokens are adaptively gated and fused with boundary frame tokens before temporal attention:

$$\bar{z}_\tau \to \text{SA}(\bar{z}_\tau) \to \text{TempAttn}\left(x_b + \alpha\psi^{\text{fwd}}, \bar{z}_\tau, x_f + \beta\psi^{\text{bwd}}\right) \to \text{MLP}(\bar{z}_\tau), \tag{17}$$

   where α and β denote learned gates as specified in Eq. 4.

Table 6 reveals that while all the above strategies for imparting motion guidance to the DiT resulted in improvements over the EDEN baseline without motion guidance, the most significant gains were achieved using the proposed gating mechanism for combining the flow tokens with image tokens prior to computing temporal attention.

4.8.2 Effect of Local and Global Flow Conditioning

We report the individual effects of local and global flow conditioning, as well as their combined effect, in Table 7, which reveals that both local and global flow conditioning contribute to the improvement in performance over the baseline, with local guidance accounting for most of the gains. Local flow conditioning introduces a new capability that is not present in EDEN, enabling explicit motion guidance at

the token level during temporal attention, thereby driving the majority of the reported improvements. In contrast, global flow conditioning enhances the existing global motion conditioning capability of EDEN by replacing its coarse cosine similarity-based difference embedding with a richer flow-based representation that provides more explicit scene-level motion information, yielding smaller but consistent gains. Combining both mechanisms achieves the best perceptual and temporal consistency performance.

4.8.3 Effect of Optical Flow Estimator

Table 8 summarizes the performance of EDENFlow-Small obtained with RAFT [30] and subsequently introduced transformer-based GMFlow [32] and FlowFormer [12] flow estimation models. Out of the three models evaluated, GMFlow achieved the best performance across all metrics, followed by RAFT. Table 9 reports the inference latency of these three models across all resolutions. GMFlow has a significantly higher latency than RAFT across all resolutions, while FlowFormer is the slowest of the three models. Thus, RAFT provides a favorable balance between interpolation quality and latency compared to the competing models.

Table 7: Evaluation of local and global flow conditioning on DAVIS.

Method	PSNR↑	LPIPS↓	FloLPIPS↓
Baseline (EDEN)	27.763	0.06607	0.09053
Local Only	**28.280**	0.05841	0.07966
Global Only	27.812	0.06478	0.08943
Ours (Both)	**28.280**	**0.0575**	**0.07840**

Table 8: Performance of EDENFlow-Small with different optical flow estimators on DAVIS.

Flow Model	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	L1↓	Params (M)
RAFT [30]	28.282	0.8928	0.0575	0.0784	9216	5.26
GMFlow [32]	**28.357**	**0.8946**	**0.0561**	**0.0767**	**9104**	**4.72**
FlowFormer [12]	28.278	0.8930	0.0578	0.0787	9216	16.15

Table 9: Optical flow estimation latency (ms) across different input resolutions.

Flow Model	256×448	480×854	544×1280	720×1280
RAFT [30]	**29.91**	**134.72**	**235.50**	**349.98**
GMFlow [32]	35.58	202.50	424.21	679.57
FlowFormer [12]	45.40	314.75	683.19	1135.59

5 Conclusion

We proposed EDENFlow, a diffusion-based VFI model that incorporates explicit optical flow priors to provide motion guidance at the local patch level as well as the global frame level. We showed that the proposed dual-path flow integration strategy employed to perform temporal attention and adaptive layer normalization in the DiT acts as an effective inductive bias for VFI, with more pronounced gains attained under large motion scenarios. Extensive experimental evaluation demonstrated consistent improvements over state-of-the-art VFI methods across multiple perceptual and temporal consistency metrics and benchmarks. Further, qualitative inspection of the interpolated frames revealed sharper object boundaries, and perceptually improved reconstructions over the EDEN baseline. A current limitation is the inference latency introduced by the external flow estimator. Our future work comprises addressing this limitation as well as jointly optimizing the entire VAE (encoder and decoder) and DiT for improved performance. We will also focus on synthesizing multiple intermediate frames, enabling an extension of EDENFlow to perform broader video generation tasks such as video prediction and inpainting.

Acknowledgment

This work was supported by the Prime Minister Early Career Research Grant by Anusandhan National Research Foundation (ANRF), Govt. of India, under grant No ANRF/ECRG/2024/003084/ENS.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xin Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3703–3712, 2019.

[3] Myungsub Choi et al. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10663–10671, 2020.

[4] Duolikun Danier, Fan Zhang, and David Bull. FloLPIPS: A bespoke video quality metric for frame interpolation. In *2022 Picture Coding Symposium (PCS)*, pages 283–287, 2022.

[5] Duolikun Danier, Fan Zhang, and David Bull. ST-MFNet: A spatio-temporal multi-flow network for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3521–3531, 2022.

[6] Duolikun Danier, Fan Zhang, and David Bull. LDMVFI: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1472–1480, 2024.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in neural information processing systems*, pages 8780–8794, 2021.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020.

[11] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17 (1-3):185–203, 1981.

[12] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *European conference on computer vision*, pages 668–685, 2022.

[13] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018.

[14] Luoxu Jin and Hiroshi Watanabe. Adapting image-to-video diffusion models for large-motion frame interpolation. *arXiv preprint arXiv:2412.17042*, 2024.

[15] Jaihyun Lew, Jooyoung Choi, Chaehun Shin, Dahuin Jung, and Sungroh Yoon. Disentangled motion modeling for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4607–4615, 2025.

[16] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. AMT: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023.

[17] Chunxu Liu, Guozhen Zhang, Rui Zhao, and Limin Wang. Sparse global matching for video frame interpolation with large motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19125–19134, 2024.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[19] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022.

[20] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.

[21] Zonglin Lyu and Chen Chen. TLB-VFI: Temporal-aware latent brownian bridge diffusion for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16260–16269, 2025.

[22] Zonglin Lyu, Ming Li, Jianbo Jiao, and Chen Chen. Frame interpolation with consecutive brownian bridge diffusion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3449–3458, 2024.

[23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[24] Jordi Pont-Tuset et al. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[28] Alexandros Stergiou. LAVIB: A large-scale video interpolation benchmark. In *38th Annual Conference on Neural Information Processing Systems, NeurIPS 2024*, 2024.

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[30] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419, 2020.

[31] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[32] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[33] Guozhen Zhang, Chunxu Liu, Yutao Cui, Xiaotong Zhao, Kai Ma, and Limin Wang. VFIMamba: Video frame interpolation with state space models. In *Advances in Neural Information Processing Systems*, volume 37, pages 107225–107248, 2024.

[34] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[35] Zihao Zhang, Haoran Chen, Haoyu Zhao, Guansong Lu, Yanwei Fu, Hang Xu, and Zuxuan Wu. EDEN: Enhanced diffusion for high-quality large-motion video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2105–2115, 2025.