This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Latent Diffusion for Stable Frame Interpolation

Jongmin Yu, Zhongtian Sun, Konstantinos Panagiotis Alexandridis, *Member, IEEE*, Angelica I. Aviles-Rivero, and Jinhong Yang, *Member, IEEE*

***Abstract*—This article presents a novel approach to video frame interpolation (VFI), called latent diffusion for stable frame interpolation (LD4SFI). LD4SFI leverages a latent diffusion model (LDM) enhanced by a vector-quantized spatiotemporal variational autoencoder (VQ-STVAE). Our method captures intrinsic orthogonal relationships in high-dimensional spatiotemporal data and seamlessly integrates complementary information across diverse video frame sequences. Given the robust sampling capabilities of LDMs, LD4SFI is conditioned on a disparity map that describes the motion dynamics between two neighboring frames. The disparity map is applied to feed explicit spatiotemporal differences into a diffusion model (DM), thereby improving the spatiotemporal smoothness of the interpolation results. With this, LD4SFI efficiently generates interpolated frames, significantly improving the continuity and visual quality of the video content. On UCF-101, densely annotated video segmentation (DAVIS), and SNU-FILM, LD4SFI achieves competitive or improved accuracy relative to existing state-of-the-art (SOTA) methods. LD4SFI produces 0.016 learned perceptual image patch similarity (LPIPS), 36.219 peak signal-to-noise ratio (PSNR), 0.974 structural similarity index (SSIM), 0.031 FloLPIPS, and 20.105 Fréchet inception distance (FID) for the UCF-101 dataset, and 0.072 LPIPS, 30.261 PSNR, 0.912 SSIM, 0.108 FloLPIPS, and 8.037 FID for the DAVIS dataset. Additionally, LD4SFI achieves the highest measured throughput among DMs on a single V100. Experimental results show that LD4SFI outperforms existing SOTA methods, demonstrating highly competitive SOTA performance across standard VFI benchmarks.**

***Index Terms*—Disparity conditioning, latent diffusion, spatiotemporal representation, vector-quantized VAE, video frame interpolation (VFI).**

Received 10 June 2025; revised 30 October 2025 and 27 March 2026; accepted 9 July 2026. This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization Program grant funded by Korean Government [Ministry of Science and ICT (MSIT)] under Grant IITP-2026-RS-2024-00436773. *(Jongmin Yu and Zhongtian Sun contributed equally to this work.) (Corresponding author: Jinhong Yang.)*

Jongmin Yu is with the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, CB3 0WA Cambridge, U.K., and also with PGAIR, ProjectG.AI, Yuseong-gu, Daejeon 34130, South Korea (e-mail: jy552@cam.ac.uk).

Zhongtian Sun is with the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, CB3 0WA Cambridge, U.K., and also with the School of Computing, University of Kent, Kent, CT2 7NZ Canterbury, U.K. (e-mail: z.sun-256@kent.ac.uk).

Konstantinos Panagiotis Alexandridis, Independent Researcher, resides in N1C 4AG London, U.K. (e-mail: alex.kostas@gmail.com).

Angelica I. Aviles-Rivero is with the Yau Mathematical Sciences Center, Tsinghua University, Haidian, Beijing 100084, China (e-mail: aviles-rivero@mail.tsinghua.edu.cn).

Jinhong Yang is with the Department of Medical Information Technology, Inje University, Gimhae 50834, Republic of Korea (e-mail: jinhong@inje.ac.kr).

Digital Object Identifier 10.1109/TNNLS.2026.3713915

I. Introduction

VIDEO frame interpolation (VFI) is a technique used to enhance the frame rate of a video by generating intermediate frames between existing ones. This process not only improves the temporal resolution of videos, thus enhancing the smoothness of motion, but also augments the visual quality of videos with low frame rates. The problem definition of VFI is as follows. Given two neighboring frames, x^0 (the earlier frame) and x^1 (the later frame), the goal of VFI is to generate an interpolated frame x^t, where $0 < t < 1$. The parameter t represents a sampling rate and is typically set to 0.5 for a $\times 2$ upsampled VFI, which generates a single frame at the midpoint between x^0 and x^t.

Functionally, VFI can improve motion smoothness and reduce visual artifacts, such as motion blur and stuttering, especially in low-frame-rate videos. This enhancement is vital for applications like slow-motion video generation [6], [7], frame rate upconversion [5], and video streaming, where seamless and fluid motion is essential for viewer satisfaction. Moreover, VFI plays a significant role in emerging technologies such as virtual and augmented reality, where high frame rates are essential for immersive, comfortable user experiences. Despite its importance, achieving high-quality VFI remains challenging, particularly in scenarios involving complex motion, occlusions, and significant brightness variations. Therefore, ongoing research focuses on developing advanced algorithms to address those challenges and further improve the performance and applicability of VFI in various multimedia applications.

The key challenge in developing VFI methods is accurately modeling the correspondence between the neighboring frames. Most deep learning-based approaches tackle this problem by extracting abstract features from high-dimensional data, such as image frames or motion flows, and using these features to predict the interpolated frame. The predicted frame is then refined by minimizing the error between the prediction and the ground truth [2], [3], [4], [8], [9], [10], [11], [12]. While these methods have significantly improved VFI performance due to the powerful representation capabilities of deep learning, their results still remain suboptimal in challenging scenarios involving dynamic textures and complex motion, as illustrated in Fig. 1.

Several studies have utilized adversarial learning [13] to enhance the high-frequency details of predicted frames, experimentally demonstrating that adversarial learning can improve the sharpness of results [1], [14]. However, adversarial learning often introduces optimization challenges, such as mode collapse [15] and vanishing gradients [16]. Joint training

2162-237X © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

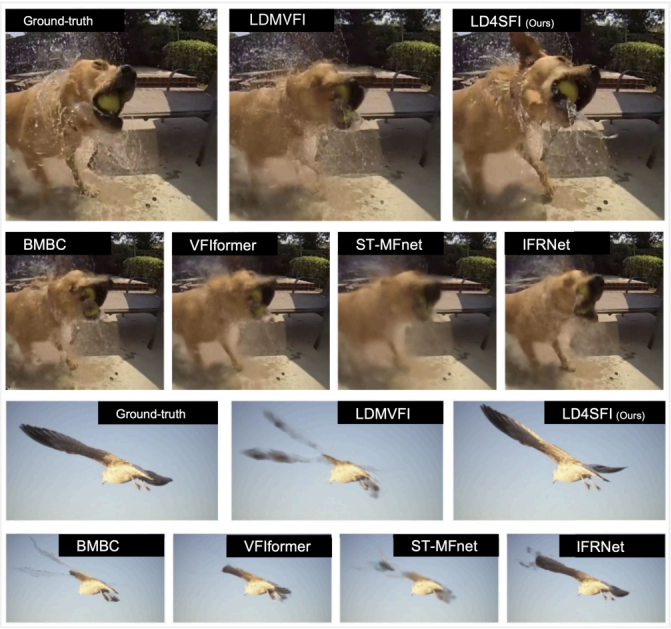


Fig. 1. VFI results of various methods, including the proposed LD4SFI. ST-MFnet [1] and IFRnet [2] use reconstruction errors based on interpolated frames. BMBC [3] and VFIformer [4] utilize reconstruction errors for both the frames and motion flows. LDMVFI [5] is a recently proposed VFI method based on the LDM. LD4SFI produces sharper interpolated frames with enhanced motion coherence and finer texture details, particularly in regions with fast motion (e.g., water splashes around the dog) and complex structures (e.g., feather edges in the bird image). Other methods exhibit motion artifacts, oversmoothing, or loss of details. Fig. 5 shows more detailed results.

with informative auxiliary features, such as motion flow, has been proposed as another solution [9], [17], [18], but these approaches tend to increase structural complexity and computational cost while introducing additional dependencies on the extra features.

To address those limitations, we propose utilizing a diffusion model (DM) [19] for VFI. DMs, in contrast to other generative models like GANs [13] and VAEs [20], have demonstrated superior image-generation quality [21], [22] and video synthesis [23]. Furthermore, conditioning tasks within DMs can be efficiently implemented using attention mechanisms, making them highly flexible. This suggests that, compared to other generative models, DMs are better suited to handle complex video data by deriving data distributions with greater precision when incorporating conditioning terms. As VFI can be formulated as the modeling of the conditional distribution between two frames, x^0 and x^1, and the interpolated frame x^t, represented by $p(x^t|x^0, x^1)$, DMs present significant potential for use in the VFI task.

Research on leveraging DMs for VFI is still in its early stages, with only a few methods published thus far [5], [24], [25], [26]. Jain et al. [24] proposed a two-stage VFI method using DMs, where the first DM generates videos with high temporal but low spatial resolution, and the second DM restores the spatial resolution to match the input. While this approach shows strong performance, it is computationally intensive. Huang et al. [25] combine a DM with an event flow simulator using optical flow. Danier et al. [5] introduced the LDMVFI, which combines a latent DM (LDM) [22] with a VFI-specific VQ-GAN [27], named vector quantized frame interpolate generative adversarial network

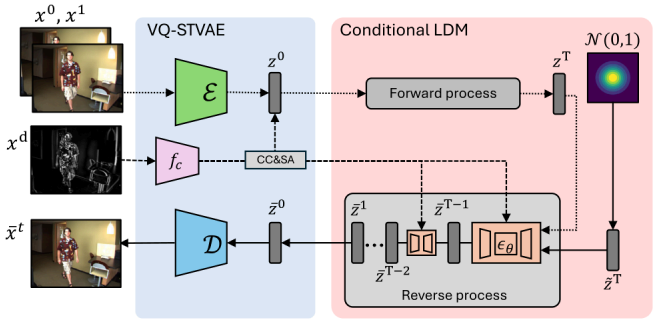


Fig. 2. Methodology overview of the proposed LD4SFI. The LD4SFI is composed of the VQ-STVAE (the blue box) and a conditional LDM (the red box). The densely dotted arrow lines denote feeding the extracted latent features from the two neighboring frames x^0 and x^1, and feeding them into the conditional LDM for training. The solid arrow lines indicate the inference of the interpolated frame in the test phase. The sparsely dotted arrow line defines the feature fusion and the conditioning based on the CCSA module.

(VQ-FIGAN). Although VQ-FIGAN improves interpolation by leveraging complementary latent features from x^0 and x^1, it does not explicitly model temporal properties, which can lead to occasional instability in the results (as seen in Fig. 1).

We propose an LDM for VFI called "latent diffusion for stable frame interpolation" (LD4SFI). Also, we present a vector-quantized spatiotemporal VAE (VQ-STVAE) that can explicitly learn a spatiotemporal representation via vector quantization [28]. Unlike other encoders for DM-based VFI methods, including LDMVFI [5], we instead of concatenating raw frames and latent features to enhance temporal properties, input a disparity map defined by the residual between the two neighboring frames. Fig. 2 shows the methodology of the proposed LD4SFI using the VQ-STVAE. Additionally, we formulate a disparity-balancing loss (DBL) to ensure temporal balance in the generated interpolated frame. We conducted an ablation study to demonstrate the effectiveness of those structural revisions for applying the DM to the VFI and loss function. The experimental results show that conditioning with the disparity map and the DBL improved VFI performance.

The key contributions of this article are as follows. First, we introduce LD4SFI, a publicly available LDM-based VFI framework comprising a VQ-STVAE and a conditional LDM. The LD4SFI framework is designed to harness the power of conditional LDM for interpolating video frames efficiently. Second, we introduce a variant of the traditional VQ-VAE, tailored to capture rich spatiotemporal information without reliance on external motion features. The VQ-STVAE is designed to provide a robust perceptual encoding of spatial-temporal data, facilitating more accurate and visually coherent frame interpolation directly from the latent space of video sequences. Third, a novel loss function is introduced to optimize the visual continuity and smoothness of interpolated frames. This balanced disparity loss strategically aligns the visual characteristics of interpolated frames with their neighboring frames in the sequence, effectively enhancing the fluidity and natural appearance of motion in upsampled videos.

II. Related Work

A major challenge in the VFI lies in accurately estimating object motion between frames, particularly in the presence of

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

complex and dynamic motion. At the same time, the interpolated frame should preserve high-resolution details, sharpness, and faithful textures. In essence, a successful VFI method must maintain temporal continuity whilst preserving, or improving, the spatial quality of the original frames.

The introduction of deep learning, particularly convolutional neural networks (CNNs) [29], has substantially advanced VFI by enabling models to learn effective spatial representations directly from frames through shared weights, cascaded structures, and hierarchical nonlinear kernels. CNN-based methods [2], [3], [9], [30] have generated sharp interpolated frames by learning stronger spatial representations than earlier approaches based on motion estimation and pixel color interpolation [31], [32]. More recently, transformer-based architectures have also been explored for VFI, with encouraging results [4].

However, VFI inherently involves more than spatial processing alone; it also requires modeling the temporal dynamics between neighboring frames. To capture these dependencies, some studies incorporate recurrent neural networks (RNNs) [10], [11], [12], while others exploit explicit motion cues such as optical flow [4], [9], [17], [18], [33]. Although such strategies can strengthen temporal reasoning, they may also increase computational cost and introduce sensitivity to the quality of the estimated motion. Recent efficiency-oriented work has therefore explored alternative temporal modeling schemes. VFIMamba [34] employs a state-space model to facilitate information transmission across frames with linear complexity and uses curriculum learning to model inter-frame dynamics across varying motion magnitudes. Likewise, *LC*-Mamba [35] investigates state-space modeling with shifted local windows and structure-preserving scan ordering, whilst BiM-VFI [36] revisits bidirectional motion-field guidance for nonuniform motion.

A further recent development in VFI is the adoption of DMs [21], [25], [26], [37], which are well suited to modeling complex conditional distributions such as $p(x^t|x^0, x^1)$. Video DMs [23], [38] have demonstrated the potential of this family for frame interpolation. In VFI-specific settings, Jain et al. [24] use two DMs, the first to improve temporal resolution and the second to recover spatial resolution. LDMVFI [5] combines VQ-FIGAN with an LDM to model the relationship between consecutive frames in latent space, whilst Huang et al. [25] propose MADIFF, a motion-aware LDM conditioned on motion flow. More recent diffusion-based studies have strengthened this direction for challenging motion and temporal reasoning. EDEN [39] enhances latent representations through a transformer-based tokenizer and augments a diffusion transformer with temporal attention and start–end difference conditioning, reporting improved perceptual quality on large-motion benchmarks. Hierarchical Flow Diffusion [40] adopts a different strategy by explicitly denoising optical flow in a coarse-to-fine hierarchy, with the aim of reducing the diffusion search space and improving efficiency, whilst reporting favorable perceptual metrics on densely annotated video segmentation (DAVIS) and SNU-FILM. TLB-VFI [26] adopts a Brownian-bridge diffusion objective and introduces a temporal-aware autoencoder with 3-D-wavelet feature gating, targeting stronger temporal cue utilization and improved runtime and parameter efficiency relative to earlier image-conditioned diffusion approaches.

These methods report competitive VFI performance, but several limitations remain. Using two DMs [24] can be computationally demanding. LDMVFI [5] mitigates this cost by operating in latent space, but, as shown in Fig. 1, its results can still be distorted in some cases. One possible reason is that its DM/LDM framework does not explicitly learn spatiotemporal representations. MADIFF [25] provides explicit motion flow as an alternative conditioning signal, but its encoder and decoder still process individual still images. More broadly, methods that strengthen diffusion with explicit flow [25], [40] or lightweight temporal modules [26] may improve conditioning, yet they can still depend on auxiliary motion estimation or only partially capture the joint spatiotemporal structure of neighboring frames.

Consequently, although hierarchical latent features extracted from two neighboring frames can improve the visual quality of the predicted frame, they may still provide insufficient information about temporal dynamics and the correspondence between the interpolated frame and its neighboring frames. In this work, we introduce LD4SFI, a latent diffusion framework designed to generate high-quality frames that are both spatially detailed and temporally consistent with their neighboring frames. To this end, we develop VQ-STVAE, which explicitly learns spatiotemporal representations within a VQ-VAE architecture. In addition, we propose the DBL, a complementary objective that enhances the temporal smoothness of the generated frames.

III. Preliminaries: DM

The goal of DMs [19], [41], [42] is to find the parameterized data distribution $p_\theta(\mathbf{x}_0)$ using a given data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. To do this, when a data $\mathbf{x}_0$ is given, the training of DMs conducts a *forward process* (*a.k.a.* diffusion process) $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, which adds Gaussian noise to the data and a *reverse process* (*a.k.a.* denoising process) $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, which denoises the given noised data by subtracting predicted noise.

The forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is a task to add Gaussian noise to data at a certain time step $t \le T$. The Gaussian noise is generated by a Gaussian probability $\mathcal{N}(\cdot)$ with scheduled variance: $\beta_1, \ldots, \beta_T$. The entire forward process to generate a completely noise-free sample $\mathbf{x}_T$ is represented by

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) \coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \mathcal{N}\left(\mathbf{x}_t;\ \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right). \quad (1)$$

The reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ can be considered as a denoising task. For each time step t, a DM predicts noise and subtracts it from the noised data. This task is represented by a Markov Chain so that it can be represented by a parametric conditional distribution, as follows:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \coloneqq \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right). \quad (2)$$

The learning of the DM is to find a suitable parametric distribution $p_\theta(\mathbf{x}_0)$ representing the given data. This distribution is central to reconstructing the original data from its

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

noisy state. However, directly optimizing the negative log-likelihood of $p_\theta(\mathbf{x}_0)$ poses computational challenges due to its complexity. Consequently, optimization is conducted via a variational lower bound, which is more computationally feasible and is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{DM}} &:= \mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \\ &= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right]. \end{aligned} \tag{3}$$

In DDPM [19], it was found that reparameterization tricks represent the forward and reverse processes; therefore, it can be replaced by the minimization of prediction error for Gaussian noise and noise prediction obtained by a neural network. As a result, (3) is more simplified to the $l2$-distance between the generated Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and predicted noise $\boldsymbol{\epsilon}_\theta$ at a certain time step. Using the notations, $\alpha_t := 1-\beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s$, the simplified loss is represented by

$$\begin{aligned}&\mathcal{L}_{\mathrm{DDPM}} \\ &= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right].\end{aligned} \tag{4}$$

The LDM applies the above noising and denoising processes to the latent feature space. Accordingly, the LDM assumes that there is an additional encoder $\mathcal{E}$ to encode raw data into a latent feature space and apply the noising and denoising process on the latent space. Equation (4) for the LDM can be rewritten as follows:

$$\mathcal{L}_{\mathrm{LDM}} = \mathbb{E}_{\mathcal{E}(x_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\left[\|\epsilon-\epsilon_\theta(z_t,t)\|^2\right]. \tag{5}$$

IV. Proposed Method: LD4SFI

A. Methodology Overview

VFI's formulation $x \sim p(x^t|x^0, x^1)$ can be considered as a sampling process for an interpolated frame from the conditional distribution. The visual quality and the temporal harmony of the generated interpolated frames can vary depending on how accurately this conditional distribution model is derived. We propose an LD4SFI to address this. The LD4SFI is composed of two main components. First, VQ-STVAE is a variation of the VQ-VAE designed to capture spatiotemporal representation explicitly. The second component is a denoising U-Net applied to operate the denoising process for using the LDM. Fig. 2 shows the VFI workflows of the LD4SFI consisting of the VQ-STVAE and the conditional LDM.

The VQ-STVAE's encoder $\mathcal{E}$ embeds the neighboring frames x^0 and x^1 into a spatiotemporal latent feature space z^0 based on the vector quantization manner [28]. After that, the conditional LDM derives the distribution of the latent features into a Gaussian distribution $\mathcal{N}(0,1)$ through the forward (*a.k.a.* noising) and reverse (*a.k.a.* denoising) processes. During the inference of the interpolated frames, a Gaussian noise $\tilde{z}^{\mathrm{T}}$ sampled from $\mathcal{N}(0,1)$ is repeatedly denoised by the UNet ϵ_θ for T steps until the latent feature $\bar{z}^0$ is unveiled. $\bar{z}^0$ is conveyed to the decoder $\mathcal{D}$ of the VQ-STVAE to generate the interpolated frame $\bar{x}^t$. When the VQ-STVAE and conditional LDM are being trained, we employ cross-channel and spatial attention (CCSA) modules using disparity map x^{d} as the feature fusion and conditioning mechanism.

B. VQ-STVAE

The role of VAE in the original LDM [22] is to encode the image's perceptual features into a latent feature space using an autoencoder and probabilistic modeling. However, the general VAEs take a static prior distribution, i.e., the VAE assumes that the latent feature distribution is Gaussian $\mathcal{N}(0,1)$. It may cause the posterior collapse, which is, regardless of the true data distribution [43], and this issue degrades the quality of generated samples. The VQ-VAE and VQ-GAN [27], which can improve the high-frequency details, have been proposed to handle this issue.

Danier et al. [5] present the VFI-specified VQ-GAN structure called the VQ-FIGAN. The VQ-FIGAN is an image-to-image autoencoder with a feature fusion based on attention mechanics to improve the visual quality of the results. The VQ-FIGAN can learn temporal properties by adding the conditioning module using x^0 and x^1 indirectly; however, this indirect way may not learn some strong correlation between the interpolated frame and the given two neighboring frames. Also, the VQ-FIGAN applies the conditioning to the decoder only, in which the codebook of the VQ modules cannot account for temporal properties.

The VQ-STVAE is designed to explicitly capture spatiotemporal representations using 3-D convolutional layers. Unlike the VQ-FIGAN proposed by Danier et al. [5], which is an image-to-image autoencoder that takes the target interpolated frame x^t as an input, the VQ-STVAE uses the two neighboring frames x^0 and x^1 as input. The latent features extracted by the 2-D convolutional layers in the VQ-FIGAN only capture the correlation within the spatial domain defined by the width and height axes; therefore, LDMVFI operates its encoder twice to extract latent features from the two neighboring frames x^0 and x^1. Consequently, the encoder of the VQ-FIGAN does not learn the temporal correlation between the two neighboring frames.

In this work, we embed 3-D convolutional layers that cover both spatial and temporal dimensions. The VQ-STVAE processes the two frames using 3-D convolutional layers to capture spatiotemporal representations explicitly. Instead of building a full 3-D convolutional encoder and decoder, we embed 3-D convolutional layers between the original 2-D convolutional layers, which is similar to Ho et al. [23]. This design enables spatiotemporal representation learning for the VQ-STVAE, thereby reducing network computational complexity. Fig. 3 illustrates the architectural details of the VQ-STVAE.

Additionally, we apply attention-based feature fusion using the latent features extracted from the disparity map x^{d}. The disparity map is defined by the subtraction results of the two frames represented by $x^{\mathrm{d}} = x^1 - x^0$. It is computationally efficient to obtain and directly model the motion dynamics.

Existing diffusion-based VFI methods [5], [25] employ feature concatenation on their decoder [5] or use highly advanced motion features [25], which trade off model complexities and VFI performances. Because conventional latent spaces are primarily learned utilizing isolated spatial representations, they necessitate complex supplementary mechanisms to extract temporal properties. The VQ-STVAE can explicitly learn spatial and temporal representations using 3-D convolutional layers, thereby mitigating this trade-off and making the spatial

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

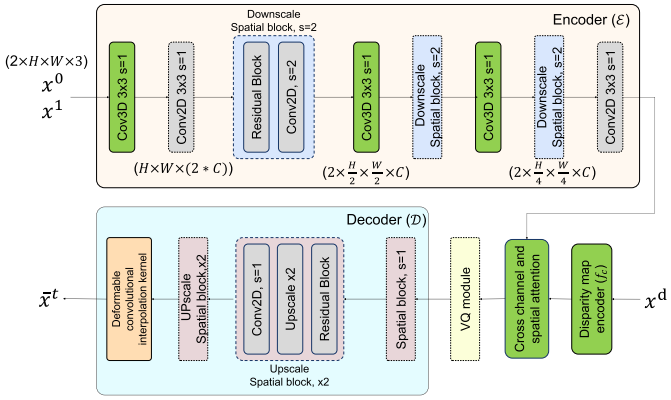


Fig. 3. Architectural details of the VQ-STVAE and the CCSA operator. The key components distinguishing the VQ-STVAE from the VQ-VAE, such as 3-D convolutional layers and the CCSA module, are marked in green.

and temporal properties of the latent feature more informative as the model complexity increases.

The attention-based feature fusion applied in the VQ-STVAE using the disparity map is conducted based on the CCSA. The feature fusion process using the CCSA is as follows. First, we extract the latent feature z^{d} of the disparity map x^{d} using a pretrained neural network f_c. Then, we compute the channel attention vector α_c^{d} based on z^{d}. We average- and max-pool z^{d} along the channel axis, then pass the resulting features through a convolutional layer to extract a combined feature. After that, we apply the sigmoid function σ to normalize the value of each term and sum those to generate the channel attention vector α_c^{d}. This process is represented as follows:

$$\alpha_c^{\mathrm{d}} = \sigma\left(W_d \cdot \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} z^{\mathrm{d}}_{h,w,*} + W_d \cdot \max_{h=1}^{H}\max_{w=1}^{W} z^{\mathrm{d}}_{h,w,*}\right) \quad (6)$$

where H and W denote the spatial axes which represent the height and width of z^{d}. W_d defines additional neural network to align the dimensionality of α_c^{d}. $*$ denotes axes that are kept during the computation of the attention vectors.

The spatial attention vector α_s^{d} is defined by applying a convolutional kernel (Conv2D) to the results of averaged pooled latent features across the channel axis. This process is represented as follows:

$$\alpha_s^{\mathrm{d}} = \sigma\left(\mathrm{Conv2D}\left(\left[\frac{1}{C}\sum_{c=1}^{C} z^{\mathrm{d}}_{*,c}\right] \Big|\Big| \max_{c=1}^{C} z^{d}_{*,c}\right)\right) \quad (7)$$

where C denote the depth of z^{d}. In computing the above attention vectors, an axis that should not be considered is frozen. For instance, in computing α_c^{d}, only spatial axes, which are height and width dimensions, are considered, and the depth axis is kept. It is described by $z^{\mathrm{d}}_{h,w,*}$ in (6). f_c can be defined by well-known pretrained neural networks such as VGG16 [44] or ResNet [45]. We employ the ResNet for our study. Fig. 3 shows the architectural details of the VQ-STVAE.

The feature fusion using the attention vectors α_c^{d} and α_s^{d} is conducted before the VQ operation, represented as follows:

$$z^0 = \frac{\left(z^0 \cdot \alpha_c^{\mathrm{d}} + \alpha_s^{\mathrm{d}} \cdot z^0\right)}{2}. \quad (8)$$

When the decoder $\mathcal{D}$ generates the interpolated frame $\bar{x}^t$, we apply a frame reconstruction approach based on the deformable convolution-based interpolation kernel [46] to improve the visual quality of the frame. Instead of directly generating the interpolated frame using the decoder, LD4SFI outputs local feature maps that can be arranged to create the frames. These local feature maps are robust to object occlusion [47]. This approach can reduce memory usage while maintaining VFI performance.

C. Conditional LDM

The VQ-STVAE encodes the intrinsic orthogonal relationships within high-dimensional spatiotemporal data into a compact latent space, integrating complementary information from diverse video frame sequences. In this compact space, a conditional LDM is applied to learn semantically important features from the spatiotemporal representation with low computational complexity. While the conditional LDM is being trained, we apply additional conditioning with the disparity map x^{d}.

Unlike LDMVFI [5] and MADIFF [25], which physically stack latent features extracted from neighboring frames [5] and motion flow [25] to the input noise before it apply to a denoising UNet, our conditioning is done by combining the latent features extracted from x^{d} to the latent feature of the UNet ϵ_θ during the denoising process using the CCSA operation (8). In computing α_c^{d} and α_s^{d}, we change the kernel dimensionalities of W_d and Conv2D to align with the dimensionality of latent feature of the denoising UNet ϵ_θ.

As demonstrated by Jain et al. [24] and some video DMs [23], [38], a DM [19] can also be used to predict the interpolated frame directly. However, it is computationally intensive and inefficient for learning representations [22]. The VFI's problem setting (i.e., requiring at least two frames for interpolation) is related not only to the spatial domain but also to the temporal domain. This problem can further worsen the high computational complexity of existing DMs. The conditional LDM alleviates this heavy computational load by compressing the image into a latent representation. Since the latent feature contains more abstract and semantic information, it helps in understanding perceptual information during frame interpolation.

D. Training and Inference

1) Training of VQ-STVAE: The objectives of the VQ-STVAE contain the objective of the VQ-VAE represented by

$$\mathcal{L}_{vq} = \left\|x^t - \bar{x}^t\right\|^2 + \left\|\mathrm{sg}\left[z_e\right] - e\right\|^2 + \beta\left\|z_e - \mathrm{sg}\left[e\right]\right\|^2 \quad (9)$$

where x^t and $\bar{x}^t$ indicate the input and the reconstructed output, respectively, and those are defined by a given interpolated frame x^t and the predicted one $\bar{x}^t$. e is the embedding vectors to z_e. sg and e indicate the stop-gradient operator and the set of the closest embedding vectors to z_e.

Also, we apply various complementary loss terms to improve the visual quality of VFI results. we combine a perceptual loss $\mathcal{L}_{per}$ [48] and a patch-level adversarial loss $\mathcal{L}_{adv}$ [49].

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

$\mathcal{L}_{per}$ is employed to improve the perceptual similarity between each consecutive frame, and $\mathcal{L}_{per}$ is defined as follows:

$$\mathcal{L}_{per} = \mathbb{E}\left[\sum_{i=1}^{N}\left\|\Phi_i^{\mathcal{E}}\left(\bar{x}^t, \bar{x}^t\right) - \frac{\Phi_i^{\mathcal{E}}\left(x^0, x^0\right) + \Phi_i^{\mathcal{E}}\left(x^1, x^1\right)}{2}\right\|^2\right] \tag{10}$$

where $\Phi_i^{\mathcal{E}}$ defines ith convolutional layers on the encoder $\mathcal{E}$ of the VQ-STVAE, respectively.

Additionally, we employ the adversarial learning loss $\mathcal{L}_{adv}$ to improve the quality of high-frequency visual features between the ground-truth frame x^t and the predicted one $\bar{x}^t$, which are represented by

$$\mathcal{L}_{adv} = -\mathbb{E}\left[\log D\left(x^t\right) - \log\left(1 - D\left(\bar{x}_t\right)\right)\right] \tag{11}$$

where D defines the discriminator used for path-level adversarial learning.

Additionally, we present the DBL $\mathcal{L}_{dbl}$ to improve temporal smoothness. The DBL is inspired by the fact that the disparity between two frames is usually proportional to the temporal distance between them. We formulate a loss that constrains the disparity between the interpolated frame and its two neighboring frames. The discrepancy between the generated interpolated frame $\bar{x}^t$ and the other two frames x^0 and x^1 should be less than or at least equal to the disparity between the two frames. Balanced disparity loss is formulated as follows:

$$\mathcal{L}_{dbl} = \max\left(\frac{\left|x^1 - \bar{x}^t\right| + \left|\bar{x}^t - x^0\right|}{2} - \left|x^1 - x^0\right|, 0\right). \tag{12}$$

The loss only works when the average of the disparities between $\bar{x}^t$ and the other two $\bar{x}^0$ and $\bar{x}^1$ are bigger than the disparity between $\bar{x}^0$ and $\bar{x}^1$. Suppose the averaged disparity is larger than the disparity between the two neighboring frames. In that case, it can be inferred that the interpolated frames cannot improve the smoothness between the two frames and may yield unstable results. In this case, the loss optimizes the lower-averaged disparity to improve temporal smoothness.

These principles are not always correct in real-world situations; depending on how complicated and dynamic the motions appear in a video, or on how long the two neighboring frames are, the disparity can differ. Nevertheless, the main target videos for VFI methods are usually 15 and 30 FPS, so based on this problem set, the above problem may not affect the disparity loss. We conduct an ablation study to demonstrate the effectiveness of the DBL.

At last, we apply regularization loss term $\mathcal{L}_{reg}$ to improve the generalization performance of LD4SFI, as follows:

$$\mathcal{L}_{reg} = \left[\frac{1}{2}\sum_i \|\mathbf{W}_i\|^2 + \sum_i |\mathbf{W}_i|\right] \tag{13}$$

where $\mathbf{W}_i$ indicates the kernel values of weight matrices of the ith layer of the VQ-STVAE for applying regularization. The VQ-STVAE encodes the intrinsic orthogonal relationships within high-dimensional spatiotemporal data into a compact latent space.

The total objective function to train the VQ-STVAE is as follows:

$$\mathcal{L}_{Total} = \mathcal{L}_{vq} + \mathcal{L}_{per} + \mathcal{L}_{adv} + \lambda_{dbl}\mathcal{L}_{dbl} + \lambda_{reg}\mathcal{L}_{reg} \tag{14}$$

where λ_{dbl} and λ_{reg} denote balancing weights for the DBL and the regularization loss. A detailed explanation of the hyperparameter settings is described in Section V.

2) Training of Conditional LDM: Training the conditional LDM optimizes the parameters of the denoising UNet ϵ_θ. Based the latent feature z^0 encoded from the two neighboring frames x^0 and x^1, it optimizes the parameters on ϵ_θ by repeating the forward and backward processes for T steps. For each step, it samples a noise version of z^0 by adding a certain amount of Gaussian noise. It predicts the noise using ϵ_θ with the conditioning using the latent feature z^{d} extracted from the disparity map x^{d}. The loss function to train the conditional LDM is defined by

$$\mathcal{L}_{\mathrm{CLDM}} = \mathbb{E}_{z^{\mathrm{T}}, z^{\mathrm{d}}, \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{1}),\mathrm{T}}\left[\left|\left|\epsilon - \epsilon_\theta\left(z^{\mathrm{T}}, \mathrm{T}\middle|z^{\mathrm{d}}\right)\right|\right|^2\right] \tag{15}$$

where T is the number of the diffusion process steps, and it is sampled from $\mathcal{U}(0, \mathrm{T})$.

z^{d} is the latent feature extracted from the disparity map: $f_c(x^{\mathrm{d}})$. z^{T} defines the t-times noised latent feature.

3) Inference: The VFI process using the LD4SFI from the given neighboring frames x^0 and x^1 starts with obtaining the disparity map x^{d} and the noise samples $\bar{z}^{\mathrm{T}}$ from Gaussian distribution $\mathcal{N}(0, 1)$. $\bar{z}^{\mathrm{T}}$ is iteratively denoised by the U-Net ϵ_θ until the latent features $\bar{z}^0$ are unveiled. During the denoising process, z^{d} is used for conditioning. $\bar{z}^0$ is applied to the decoder $\mathcal{D}$ to generate the interpolated frame.

V. Experimental Results

A. Implementation Details

The hyperparameter settings for training the VQ-STVAE are as follows: To improve computational efficiency, we use DDIM [37] as a sampler for all DMs. The number of sampling steps for the DM is set to 200. This sampling scale is equivalent to other diffusion-based VFI methods [5], [25]. The Adam optimizer is employed to optimize the VQ-STVAE, while AdamW is used for the conditional LDM. The learning rates for VQ-STVAE and the conditional LDM are set to 1×10^{-5} and 1×10^{-6}, respectively. Both VQ-STVAE and the conditional LDM are trained for 60 epochs. The downsampling factor f for the VQ-STVAE is 32, matching the downsampling factor used in LDMVFI [5] and MADIFF [25]. λ_{dbl} and λ_{reg} are set to 0.01. Two RTX A6000 GPUs are mainly used to train and evaluate the proposed LD4SFI.

B. Experimental Settings

1) Experimental Protocol: We utilize Vimeo90k [50], the most widely used training set for VFI. However, previous studies have highlighted the limitations of Vimeo90k in terms of the range and diversity of motion magnitudes [51]. To address these limitations and enhance the performance of VFI methods more comprehensively, we also include samples from the BVI-DVC dataset [52]. Our final training set comprises 64 612 frame triplets from Vimeo90k-septuplets and 17 600 from BVI-DVC. For data augmentation, we apply random cropping of 256 × 256 patches, random flipping, and temporal-order reversal. For model evaluation, we use three widely recognized VFI benchmarks: 1) UCF-101 [53]; 2) DAVIS [54]; and 3) SNU-FILM [55]. We follow each dataset's published terms. For example, the UCF-101 dataset is provided for research

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

use and comprises videos collected from YouTube; use is subject to the dataset terms and the YouTube Terms of Service. The DAVIS dataset is released for research/benchmarking. We perform no content-altering preprocessing on evaluation sets: only per-pixel normalization and resizing, as required by the model's input constraints. Those experimental protocols are based on Danier et al. [5] and Huang et al. [25]. The detailed information of the above datasets is as follows:

2) Vimeo90K Dataset: Vimeo-90k [50] is a large-scale, high-quality video dataset. This dataset consists of 89 800 video clips downloaded from vimeo.com, which cover a large variety of scenes and actions. It is designed for the following four video processing tasks: temporal frame interpolation, video denoising, video deblocking, and video super-resolution.

The triplet dataset consists of 73 171 3-frame sequences with a fixed resolution of 448 × 256, extracted from 15k selected video clips from Vimeo-90K. This dataset is designed for temporal frame interpolation.

3) BVI-DVC Dataset: BVI-DVC [52] is presented for training CNN-based video compression systems, with specific emphasis on machine learning tools that enhance conventional coding architectures, including spatial resolution and bit depth up-sampling, postprocessing, and in-loop filtering. BVI-DVC contains 800 sequences at various spatial resolutions from 270p to 2160p and has been evaluated on ten existing network architectures for four different coding tools. Experimental results show that this database produces significant improvements in terms of coding gains over three existing (commonly used) image/video training databases under the same training and evaluation configurations.

4) DAVIS Dataset: The DAVIS dataset [56] is a high-quality and high-resolution densely annotated video segmentation dataset under two resolutions, 480p and 1080p. There are 50 video sequences with 3455 densely annotated frames at the pixel level. 30 videos with 2079 frames are for training, and 20 videos with 1376 frames are for validation.

5) UCF-101 Dataset: UCF101 [53], which is related to human actions and contains 379 triplets with a resolution of 256 × 256.

6) SNU-FILM Dataset: SNU-FILM dataset [55] specifies a list of 310 triplets at four motion magnitude levels. As the original sequences are provided in the SNU-FILM dataset, we extended its predefined test triplets to quintuplets for evaluation here. The evaluation is divided into four settings: Easy, Medium, Hard, and Extreme. The average motion magnitude increases from Easy to Extreme.

7) Evaluation Metrics: We use five evaluation metrics: 1) structural similarity index (SSIM) [57]; 2) peak signal-to-noise ratio (PSNR) [57]; and 3) learned perceptual image patch similarity (LPIPS) [58]; 4) flow difference-weighted LPIPS (FLoLPIPS) [59]; and 5) Fréchet Inception Distance (FID) [60]. SSIM measures the similarity between two images, focusing on structural information such as luminance, contrast, and texture. PSNR quantifies the quality of a reconstructed or compressed image relative to the original by measuring the ratio of the maximum possible signal power to the power of corrupting noise, which affects the image's representation fidelity. LPIPS evaluates perceptual similarity between images using deep neural networks trained on perceptual tasks, considering human visual perception when comparing image patches. Unfortunately, those image-based metrics assess the fidelity of individual frames in isolation and therefore fail to capture spatial quality, temporal realism, temporal distributional similarity, or dynamical consistency across frames.

To provide a more comprehensive evaluation that encompasses spatial fidelity, temporal consistency, and perceptual stability, we also employ FloLPIPS and FID. Specifically, FloLPIPS is designed to quantify temporal distortions and motion-related artifacts by incorporating optical flow into perceptual similarity assessment, thereby reflecting frame-to-frame consistency. In contrast, FID evaluates the distributional alignment between generated interpolated frames and real video frames in a learned feature space, measuring how well the synthesized content adheres to the natural statistics of real-world data.

Furthermore, computational efficiency is assessed by measuring the inference speed of the proposed LD4SFI on a single NVIDIA V100 GPU. This hardware configuration was referred to from MADIFF [25]. The reported runtime is directly compared with that of MADIFF [25], which is evaluated under the same hardware configuration.

C. Ablation Study

The main differences between our LD4SFI and other existing LDM-based VFIs [5], [25], [26] are the inclusion of 3-D convolutional layers in the VQ-STVAE, the use of the disparity map, and the introduction of the DBL. Structurally, whereas the LDMVFI [5] relies upon independent 2-D spatial encodings and MADIFF [25] necessitates computationally heavy auxiliary event-to-flow simulators, LD4SFI employs VQ-STVAE. By integrating 3-D convolutional layers, the VQ-STVAE intrinsically learns a unified, coupled spatiotemporal representation in a single encoding pass without external heuristic dependencies such as motion flow or predefined even features. Furthermore, rather than relying on the raw concatenation of latent features (as in LDMVFI) or deterministic Brownian Bridge mapping (as in TLB-VFI [26]), LD4SFI conditions its denoising U-Net through the CCSA module, dynamically gating latent features based on a computationally efficient disparity map.

In terms of the objective function, the mathematical objectives of these frameworks also diverge; whilst LDMVFI and MADIFF utilize standard DDPM prediction errors combined with unconstrained adversarial and perceptual losses, LD4SFI introduces the DBL as a complementary term to improve motion smoothness. The DBL is formulated to provide a strict geometric boundary in the pixel domain, expected to explicitly penalize the generative process if the average visual disparity of the newly synthesized frame exceeds the inherent dynamic shift between the original temporal anchors. This mathematical constraint acts synergistically with the standard objectives to effectively suppress the temporally unstable hallucinations and sequence jitter that frequently compromise unregularized generative models, thereby providing rigid temporal coherence alongside robust spatial fidelity.

To demonstrate the effectiveness of these components, we conducted ablation studies. At the beginning, we derive two baseline LD4SFI models: $\text{LD4SFI}_{\text{VQ-VAE}}$ and $\text{LD4SFI}_{\text{VQ-STVAE}}$. $\text{LD4SFI}_{\text{VQ-VAE}}$ defines the LD4SFI with normal VQ-VAE, which does not have 3-D convolution and

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

VFI PERFORMANCES DEPENDING ON THE FEATURE FUSION ON VQ-STVAE AND THE DBL LOSS. SNU-FILM-AVERAGE CONTAINS THE AVERAGE RESULTS FOR "MEDIUM," "HARD," AND "EXTREME" RESULTS. THE BEST PERFORMANCE IS HIGHLIGHTED IN ORANGE. THE BALANCING WEIGHT λ_{dbl} FOR $\mathcal{L}_{dbl}$ IS SET TO 0.01

Model settings	UCF-101					DAVIS					SNU-FILM-Average				
	LPIPS(↓)	PSNR(↑)	SSIM(↑)	FloLPIPS(↓)	FID(↓)	LPIPS	PSNR	SSIM	FloLPIPS	FID	LPIPS	PSNR	SSIM	FloLPIPS	FID
$\text{LD4SFI}_{\text{VQ-VAE}-x^{\text{d}}}$	0.042	31.831	0.935	0.041	33.671	0.112	24.512	0.821	0.190	18.571	0.101	25.988	0.892	0.186	31.612
$\text{LD4SFI}_{\text{VQ-VAE}-x^{\text{d}}} + \mathcal{L}_{dbl}$	0.036	32.019	0.952	0.041	32.016	0.110	24.910	0.829	0.182	17.109	0.083	27.226	0.898	0.158	26.617
$\text{LD4SFI}_{\text{VQ-VAE}+x^{\text{d}}}$	0.027	32.032	0.943	0.037	29.526	0.109	25.053	0.827	0.175	14.295	0.081	28.265	0.899	0.126	20.056
$\text{LD4SFI}_{\text{VQ-VAE}+x^{\text{d}}} + \mathcal{L}_{dbl}$	0.027	32.189	0.961	0.034	24.051	0.104	25.807	0.859	0.156	11.950	0.063	28.982	0.907	0.103	18.371
$\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}}$	0.035	32.507	0.945	0.035	25.781	0.100	27.753	0.890	0.155	12.451	0.048	29.944	0.922	0.102	18.002
$\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}} + \mathcal{L}_{dbl}$	0.025	34.063	0.960	0.032	21.107	0.085	28.592	0.903	0.130	9.561	0.040	31.101	0.933	0.090	17.804
$\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}}$	0.020	35.520	0.971	0.033	21.096	0.080	29.408	0.908	0.111	9.172	**0.035**	32.653	0.943	0.083	16.035
$\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}} + \mathcal{L}_{dbl}$	**0.016**	**36.219**	**0.974**	**0.031**	**20.105**	**0.072**	**30.261**	**0.912**	**0.108**	**8.037**	0.036	**33.319**	**0.949**	**0.076**	**15.862**

is only composed of 2-D convolution. The VQ-VAE on this baseline is trained on a single frame, so it does not explicitly learn spatiotemporal representations. Methodologically, the input protocol of VQ-VAE is similar to that of VQ-FIGAN in LDMVFI. $\text{LD4SFI}_{\text{VQ-STVAE}}$ indicates the LD4SFI with the VQ-STVAE presented in Section IV-B. Based on those two baselines, we have established eight variations depending on the key components, which are the conditioning using x^{d} and the DBL. About the $\text{LD4SFI}_{\text{VQ-STVAE}}$, $\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}}$, and $\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}}$ define LD4SFI with or without the conditioning using x^{d}, respectively. This notation is also used to indicate the presence or absence of DBL. The same notation approach is used to denote the variation of $\text{LD4SFI}_{\text{VQ-VAE}}$.

By conducting experiments using those eight models, we aim to prove that learning explicit spatiotemporal representations with VQ-STVAE, feature fusion with x^{d}, and training with $\mathcal{L}_{dbl}$ can improve VFI performance. Additionally, we conduct the ablation study for $\mathcal{L}_{\text{dbl}}$. In this ablation study, we additionally try to show the sensitivity of $\mathcal{L}_{\text{dbl}}$ for the VFI performance on LD4SFI by training multiple LD4SFI models with various settings of the balancing weight λ_{dbl} for $\mathcal{L}_{\text{dbl}}$. The performances are quantitatively evaluated using LPIPS, PSNR, SSIM, FloLPIPS, and FID.

1) Performances Depending on the Dbl: Introducing the DBL loss ($\mathcal{L}_{dbl}$) consistently improves reconstruction fidelity and temporal consistency across datasets compared with identical settings without DBL. Table I represents the results of a comprehensive ablation study, which describes the performance of the LD4SFI with respect to the usage of the disparity map and the DBL. Relative to $\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}}$, adding DBL improves VFI performances. In UCF-101, PSNR is improved from 32.507 to 34.063, and SSIM is improved from 0.945 to 0.960. LPIPS is also improved from 0.035 to 0.025. Crucially, temporal consistency and perceptual quality are similarly enhanced, with FloLPIPS decreasing from 0.035 to 0.032 and FID reducing from 25.781 to 21.107; similar gains appear on DAVIS and the SNU-FILM-Average splits.

Incorporating disparity-feature fusion ($+x^{\text{d}}$) further elevates performance: $\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}}$ already attains strong results on UCF-101. It produces 0.020 LPIPS, 35.520 PSNR, 0.971 SSIM, 0.033 FloLPIPS, and 21.096 FID. Coupling it with DBL yields the best overall scores on UCF-101 and DAVIS. In UCF-101, it achieves 0.016 LPIPS, 36.219 PSNR, 0.974 SSIM, 0.031 FloLPIPS, and 20.105 FID. In the DAVIS dataset, it produces 0.072 LPIPS, 30.261 PSNR, 0.912 SSIM, 0.108 FloLPIPS, and 8.037 FID. On SNU-FILM-Average, $\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}+\mathcal{L}_{dbl}}$ achieves the highest performance in most evaluation metrics, recording 0.036 LPIPS, 33.319 PSNR, 0.949 SSIM, 0.076 FloLPIPS, and 15.862 FID. These results indicate that DBL and disparity-feature fusion are complementary: DBL primarily boosts spatial fidelity whilst simultaneously improving temporal coherence and perceptual realism, and disparity features provide a stronger representational substrate on which DBL can operate.

TABLE II

VFI PERFORMANCES DEPENDING ON THE BALANCING WEIGHT λ_{dbl} FOR $\mathcal{L}_{dbl}$ ON THE UCF-101 DATASET AND THE DAVIS DATASET. THE BASELINE MODEL IS THE LD4SFI TRAINED WITH DISPARITY MAP (I.E., $\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{D}}}$). THE FIRST-RANKED AND SECOND-RANKED PERFORMANCES ARE HIGHLIGHTED IN ORANGE AND YELLOW, RESPECTIVELY

Sensitivity to λ_{dbl}	UCF-101			DAVIS		
	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM
None ($\lambda_{dbl} = 0$)	0.020	35.520	0.971	0.080	29.408	0.908
$\lambda_{\text{dbl}} = 0.001$	0.020	35.928	0.972	**0.071**	29.125	0.910
$\lambda_{dbl} = 0.005$	0.018	**36.301**	**0.973**	0.075	**30.082**	0.909
$\lambda_{dbl} = 0.01$	**0.016**	**36.219**	**0.974**	**0.072**	**30.261**	**0.912**
$\lambda_{dbl} = 0.05$	0.017	36.107	0.972	0.073	29.990	0.910
$\lambda_{dbl} = 0.1$	**0.016**	36.113	0.973	0.073	30.017	**0.913**
$\lambda_{dbl} = 1.0$	0.017	35.802	0.971	0.076	29.592	0.910

Table II shows that the sensitivity of VFI performances depends on the balancing weight λ_{dbl} for the DBL loss $\mathcal{L}_{dbl}$. The sensitivity study in Table II shows a broad optimum for λ_{dbl} in the 10^{-2} – 10^{-1} range. On UCF-101, the best PSNR (36.301) occurs at $\lambda_{dbl} = 0.005$, the best SSIM (0.974) at (0.01), and the best LPIPS (0.016) at (0.01) and (0.1). On DAVIS, the best LPIPS (0.071) is at (0.001), whereas the best PSNR (30.261) is at (0.01) and the best SSIM (0.913) at (0.1). In practice, $\lambda_{dbl} = 0.01$ delivers a balanced trade-off across datasets. We set this value for further comparison with the existing state-of-the-art (SOTA) methods. Overall, the weakest variant is $\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}}$, confirming that removing both DBL and feature fusion limits quality. DBL alone yields sizeable gains; combining DBL with disparity-feature fusion delivers the strongest performance, establishing clear improvements across spatial fidelity, perceptual quality, and temporal efficiency.

2) VQ-STVAE and Conditional LDM With x^d: To isolate the structural advantages of explicit spatiotemporal modeling, we first evaluate the transition from a purely spatial 2D encoder ($\text{LD4SFI}_{\text{VQ-VAE}}$) to the proposed 3D-convolutional

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III

QUANTITATIVE COMPARISON RESULTS ON UCF-101 AND DAVIS DATASETS. THE LOWER (↓) LPIPS, FLOLPIPS, AND FID, AND THE HIGHER (↑) PSNR AND SSIM INDICATE BETTER PERFORMANCE. THE BEST PERFORMANCE IS HIGHLIGHTED IN ORANGE. THE SYMBOL "-" DENOTES THAT THE PERFORMANCES ARE NOT PROVIDED

Methods	Year	UCF-101					DAVIS				
		LPIPS(↓)	PSNR(↑)	SSIM(↑)	FloLPIPS(↓)	FID(↓)	LPIPS	PSNR	SSIM	FloLPIPS	FID
BMBC [3]	*'20*	0.034	35.152	0.969	0.045	33.171	0.125	26.835	0.869	0.185	15.354
AdaCoF [9]	*'20*	0.034	34.904	0.968	0.046	32.783	0.148	26.445	0.854	0.198	17.194
CDFI [61]	*'21*	0.036	32.653	0.968	0.049	33.742	0.157	26.471	0.857	0.211	18.098
XVFI [17]	*'21*	0.038	32.224	0.966	0.050	33.868	0.129	26.565	0.863	0.185	16.163
ABME [62]	*'21*	0.058	35.383	0.970	0.069	37.065	0.151	27.002	0.868	0.209	16.931
IFRNet [2]	*'22*	0.032	35.294	0.969	0.044	28.803	0.114	29.014	0.904	0.170	14.227
ST-MFNet [1]	*'23*	0.036	33.384	0.970	0.049	34.475	0.125	28.287	0.895	0.181	15.626
FLAVR [63]	*'23*	0.035	33.389	0.971	0.046	31.449	0.209	27.450	0.873	0.248	22.663
VFIformer [4]	*'22*	0.039	32.745	0.968	0.051	34.112	0.191	26.241	0.850	0.242	21.702
LDMVFI [5]	*'24*	0.026	32.186	0.963	0.035	26.301	0.107	25.541	0.833	0.153	12.554
MADIFF [25]	*'24*	0.024	32.159	0.966	0.033	24.289	0.096	26.069	0.853	0.142	11.089
VFIMamba [34]	*'24*	-	35.454	0.970	-	-	0.108	-	-	-	-
LC-Mamba [35]	*'25*	-	35.425	0.970	-	-	-	-	-	-	-
TLB-VFI [26]	*'25*	-	-	-	-	-	0.086	26.272	0.860	0.126	8.299
UPR-Net [64]	*'25*	0.032	35.470	0.970	0.039	21.970	0.134	-	-	0.172	15.002
LD4SFI (Ours)	-	**0.016**	**36.219**	**0.974**	**0.031**	**20.105**	**0.072**	**30.261**	**0.912**	**0.108**	**8.037**

architecture ($\text{LD4SFI}_{\text{VQ-STVAE}}$). As delineated in Table I, replacing the standard VQ-VAE with the VQ-STVAE yields consistent improvements across all datasets, confirming that native spatiotemporal representation learning is highly beneficial for VFI, particularly concerning temporal consistency. For instance, in the absence of disparity conditioning and DBL, $\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}}$ achieves a PSNR of 32.507, an SSIM of 0.945, and a FloLPIPS of 0.035 on the UCF-101 dataset, outperforming the $\text{LD4SFI}_{\text{VQ-VAE}-x^{\text{d}}}$ baseline, which records a PSNR of 31.831, an SSIM of 0.935, and a FloLPIPS of 0.041. This structural advantage becomes particularly pronounced under severe spatial displacement. On the challenging SNU-FILM-Average benchmark, the VQ-STVAE variant improves the PSNR from 25.988 to 29.944, the SSIM from 0.892 to 0.922, and reduces the FloLPIPS from 0.186 to 0.102. These empirical results suggest that integrating 3D convolutional layers effectively mitigates the spatial bottleneck inherent in 2D encodings, thereby reducing structural degradation and enhancing temporal stability in complex motion sequences.

Furthermore, the ablation results substantiate the efficacy of deterministic disparity map conditioning (x^{d}) in guiding the generative process. By employing CCSA to dynamically gate the diffusion U-Net, the model can efficiently allocate representational capacity to localized regions of motion. When applied to the standard VQ-VAE baseline, disparity conditioning ($\text{LD4SFI}_{\text{VQ-VAE}+x^{\text{d}}}$) marginally increases the PSNR from 31.831 to 32.032 on UCF-101. However, when fused with the explicit spatiotemporal encodings of the VQ-STVAE, the performance gains are substantially amplified. Specifically, comparing $\text{LD4SFI}_{\text{VQ-STVAE}-x^{\text{d}}}$ with $\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}}$ reveals a significant enhancement on the UCF-101 dataset, with the PSNR elevating from 32.507 to 35.520, the LPIPS decreasing from 0.035 to 0.020, and the FID improving from 25.781 to 21.096. Similar trends are observed on the DAVIS dataset, where the introduction of x^{d} into the VQ-STVAE architecture increases the PSNR by 1.655 dB (from 27.753 to 29.408) and decreases FloLPIPS from 0.155 to 0.111, confirming that disparity-guided attention synergizes with robust native spatiotemporal features to yield temporally coherent outputs.

Consequently, the complete architecture integrates 3-D convolution-based VQ-STVAE, disparity map conditioning, and the geometric temporal regularization provided by the disparity balancing loss ($\mathcal{L}_{dbl}$). The empirical data confirm that these components function harmoniously to enforce spatial fidelity, perceptual realism, and temporal coherence. The fully configured model, $\text{LD4SFI}_{\text{VQ-STVAE}+x^{\text{d}}+\mathcal{L}_{dbl}}$, converges to achieve the optimal performance across all evaluated metrics. On the UCF-101 dataset, it records 0.016 LPIPS, 36.219 PSNR, 0.974 SSIM, 0.031 FloLPIPS, and 20.105 FID. Moreover, it demonstrates exceptional robustness on the SNU-FILM-Average dataset, retaining high structural integrity and temporal continuity (LPIPS: 0.036, PSNR: 33.319, SSIM: 0.949, FloLPIPS: 0.076, FID: 15.862). Consequently, the ablation study firmly demonstrates that the architectural synthesis of a unified spatiotemporal representation and targeted disparity gating constitutes a highly advantageous framework for diffusion-based VFI.

D. Comparison With Existing SOTA Methods

We evaluate the quantitative and qualitative performance of LD4SFI in VFI against an extensive selection of SOTA methods. This comparative analysis includes BMBC [3], AdaCoF [9], CDFI [61], XVFI [17], ABME [62], IFRNet [2], ST-MFNet [1], FLAVR [63], VFIformer [4], LDMVFI [5], MADIFF [25], VFIMamba [34], *LC*-Mamba [35], TLB-VFI [26], and UPR-Net [64]. Tables III and IV detail the quantitative outcomes across the UCF-101 and DAVIS datasets and the SNU-FILM benchmarks, respectively. We refer to qualitative results reported in MADIFF [25] and TLB-VFI [26]. Following standard evaluation protocols, the "Easy" subset of the SNU-FILM dataset is omitted, as predictive accuracy across modern architectures on this split has largely saturated.

An assessment of the quantitative metrics indicates that LD4SFI yields favorable outcomes compared to contemporary deterministic and state-space baselines. On the UCF-101 dataset, highly efficient architectures such as VFIMamba and UPR-Net achieve competitive PSNRs of 35.454 and 35.470, respectively. LD4SFI records a PSNR of 36.219, an SSIM of 0.974, and an LPIPS of 0.016 under the same conditions.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV

QUANTITATIVE COMPARISON RESULTS ON SNU-FILM DATASETS. THE BEST PERFORMANCE IS HIGHLIGHTED IN ORANGE. THE SYMBOL "-" DENOTES THAT THE PERFORMANCES ARE NOT PROVIDED

Methods	SNU-FILM-Medium					SNU-FILM-Hard					SNU-FILM-Extreme				
	LPIPS	PSNR	SSIM	FloLPIPS	FID	LPIPS	PSNR	SSIM	FloLPIPS	FID	LPIPS	PSNR	SSIM	FloLPIPS	FID
BMBC [3]	0.034	35.313	0.977	0.059	12.272	0.068	29.333	0.927	0.118	25.773	0.145	23.921	0.843	0.237	49.519
AdaCoF [9]	0.039	35.054	0.975	0.066	14.173	0.080	29.464	0.924	0.131	27.982	0.152	24.311	0.844	0.234	52.848
CDFI [61]	0.036	35.224	0.977	0.066	12.906	0.081	29.660	0.929	0.141	29.087	0.163	24.645	0.854	0.255	53.916
XVFI [17]	0.039	34.497	0.975	0.072	16.000	0.075	29.381	0.929	0.138	29.483	0.142	24.677	0.855	0.233	54.449
ABME [62]	0.042	35.770	0.979	0.076	15.159	0.092	30.582	0.936	0.168	34.236	0.182	25.423	0.864	0.300	63.561
IFRNet [2]	0.033	35.943	0.979	0.058	12.084	0.065	30.413	0.936	0.122	25.436	0.136	25.050	0.859	0.229	50.047
ST-MFNet [1]	0.036	37.111	0.985	0.061	11.716	0.073	31.698	0.951	0.123	25.512	0.148	25.810	0.874	0.238	53.563
FLAVR [63]	0.049	35.988	0.979	0.077	15.006	0.112	30.541	0.937	0.169	34.746	0.217	25.188	0.860	0.303	72.673
VFIformer [4]	0.033	36.093	0.979	0.053	11.271	0.061	30.672	0.937	0.100	22.775	0.119	25.429	0.864	0.185	40.586
LDMVFI [5]	0.028	33.996	0.970	0.053	12.485	0.060	28.547	0.917	0.114	26.520	0.123	23.934	0.837	0.204	47.042
MADIFF [25]	0.027	34.255	0.973	0.049	11.022	0.058	28.961	0.923	0.107	22.707	0.118	24.150	0.847	0.198	44.923
VFIMamba [34]	-	36.403	0.981	-	-	-	30.992	0.940	-	-	0.115	25.792	0.868	-	-
LC-Mamba [35]	-	36.175	0.980	-	-	-	30.723	0.938	-	-	-	25.481	0.864	-	-
TLB-VFI [26]	0.021	35.308	0.977	0.036	8.518	0.044	29.592	0.929	0.085	17.470	0.095	24.513	0.847	0.151	29.868
UPR-Net [64]	0.034	36.264	0.980	0.052	10.983	0.062	30.861	0.937	0.097	22.127	0.112	25.634	0.864	0.176	40.098
LD4SFI (Ours)	**0.021**	**38.245**	**0.988**	**0.027**	**5.995**	**0.037**	**33.107**	**0.955**	**0.081**	**15.081**	**0.051**	**28.604**	**0.905**	**0.120**	**26.504**

Furthermore, LD4SFI establishes strong perceptual fidelity and temporal efficiency, achieving an FID of 20.105 and a FloLPIPS of 0.031. This performance differential is similarly observed on the DAVIS benchmark, where LD4SFI achieves an LPIPS of 0.072, a PSNR of 30.261, an FID of 8.037, and a FloLPIPS of 0.108. This suggests that the latent diffusion prior, when supported by a robust architectural backbone, aids in the preservation of high-frequency spatial details and temporal consistency across diverse video sequences.

When evaluating against recent latent diffusion-based frameworks, the structural differences in representation learning become more apparent. On the DAVIS dataset, LDMVFI and MADIFF record PSNRs of 25.541 and 26.069, alongside FIDs of 12.554 and 11.089, respectively, whereas LD4SFI achieves a PSNR of 30.261 and an FID of 8.037. In terms of temporal efficiency, LDMVFI and MADIFF yield FloLPIPS values of 0.153 and 0.142, compared to 0.108 for LD4SFI. This measurable gap can be linked to the underlying spatiotemporal encoding strategies. LDMVFI relies upon independent 2-D spatial encodings and the indirect concatenation of latent features, which may constrain its capacity to explicitly model temporal continuity. While MADIFF leverages auxiliary motion flow to address temporal dynamics, its core autoencoder operates primarily on static spatial frames. In contrast, the proposed architecture intrinsically couples these dimensions. By contrast, LD4SFI employs a vector-quantized spatiotemporal variational autoencoder (VQ-STVAE) integrated with 3-D convolutional layers. This configuration intrinsically couples spatial and temporal dimensions into a unified latent representation, which provides the diffusion U-Net with a more temporally coherent prior without necessitating computationally intensive external motion estimators.

Furthermore, LD4SFI demonstrates distinct representational strategies when compared to other advanced temporal diffusion formulations. For example, TLB-VFI utilizes a temporal-aware autoencoder mapped via a deterministic Brownian Bridge, yielding an LPIPS of 0.086, a PSNR of 26.272, an FID of 8.299, and a FloLPIPS of 0.126 on DAVIS. Rather than relying on deterministic temporal mapping, LD4SFI dynamically gates the denoising U-Net through a CCSA module conditioned on a disparity map. This explicit conditioning mechanism allows the network to allocate representational capacity specifically to localized regions exhibiting interframe motion.

TABLE V

EXECUTION SPEED OF THE PROPOSED LD4SFI AND OTHER METHODS [1], [3], [4], [5], [25]. EXECUTION SPEEDS (A.K.A. FPS) OF OTHER METHODS ARE REFERRED TO FROM MADIFF [25]. THOSE EXECUTION SPEEDS ARE MEASURED BY USING A SINGLE V100

Method	LD4SFI	BMBC	VFIformer	ST-MFNet	LDMVFI	MADIFF
FPS	0.08	1.04	0.30	**3.84**	0.06	0.02

The practical effects of these architectural choices are most observable under the severe motion magnitudes present in the SNU-FILM dataset. As temporal displacement increases from the "Medium" to "Extreme" splits, unconstrained generative models frequently encounter structural degradation and temporally unstable hallucinations. On the SNU-FILM-Extreme split, LDMVFI and MADIFF produce PSNRs of 23.934 and 24.150, alongside FloLPIPS scores of 0.204 and 0.198, respectively. Recent recurrent and state-space architectures like UPR-Net and VFIMamba similarly experience reductions, recording PSNRs of 25.634 and 25.792. Under these challenging conditions, LD4SFI retains a relatively higher degree of structural integrity and temporal stability, achieving an LPIPS of 0.051, a PSNR of 28.604, an SSIM of 0.905, an FID of 26.504, and a FloLPIPS of 0.120. This sustained accuracy in high-motion regimes underscores the robustness of the combined VQ-STVAE and CCSA architecture. By anchoring the generative process in explicitly coupled spatiotemporal features and dynamically attending to structural disparity, the framework effectively mitigates sequence jitter and maintains temporal coherence, even when interframe dynamics are highly complex.

The qualitative results further demonstrate LD4SFI's highly competitive perceptual quality compared to the listed SOTA methods. As shown in Fig. 4, the interpolated frames generated by LD4SFI exhibit significantly more detailed information than those produced by other methods, particularly in areas requiring fine-grained detail (e.g., facial expressions of moving individuals or object textures). Additionally, Fig. 5 explains

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



Fig. 4. Visual examples of frames interpolated by the SOTA methods [1], [2], [3], [5] and the proposed LD4SFI. Under large and complex motions, LD4SFI preserves thin structures and textures with good temporal consistency.

that LD4SFI's results are more robust to extreme and complex motion dynamics. Compared with the results of other methods [1], [2], [3], [5], the interpolated frames generated by LD4SFI contain more high-frequency components, such as feather patterns or water droplets, in scenes where a bird flaps its wings or a dog shakes off water. Table V summarizes the frames per second (FPS) for representative SOTA methods [1], [3], [4], [5], [25] using the figures collated by Huang et al. [25]. Among the nondiffusion baselines, STMF-Net [1] attains the highest throughput at 3.84 FPS. Diffusion-based approaches, including our LD4SFI, are markedly slower; LD4SFI processes at 0.08 FPS, whereas LDMVFI and MADIFF achieve 0.06 and 0.02 FPS, respectively. This gap is expected: DMs require iterative denoising, which imposes substantial computational overhead compared with some of the convolutional or transformer-based architectures [1], [3], [4].

Despite this inherent disadvantage, LD4SFI is faster than LDMVFI/MADIFF under the same number of steps. We

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



Fig. 5. VFI results of the proposed LD4SFI and other methods [1], [2], [3], [5]. The LD4SFI reproduces high-frequency details and considers motion blurs more naturally. Those results are an extended version of the Fig. 1.

attribute this to two design choices: First, a lightweight 3-D-convolutional stem that learns spatiotemporal features directly from short frame windows, avoiding separate per-frame encodings and late fusion as in LDMVFI; and second, the omission of auxiliary motion-estimation components—such as the image-to-event generator used by MADIFF—which increase both parameter count and memory traffic. Together, these choices reduce encoder redundancy and eliminate the need for additional motion-extraction passes, yielding a more favorable speed–quality trade-off without major architectural modifications.

In conclusion, LD4SFI is the leading method in this quantitative and qualitative comparison, consistently achieving lower LPIPS and higher PSNR and SSIM values across all datasets. This demonstrates its superior ability to produce high-quality video interpolations.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

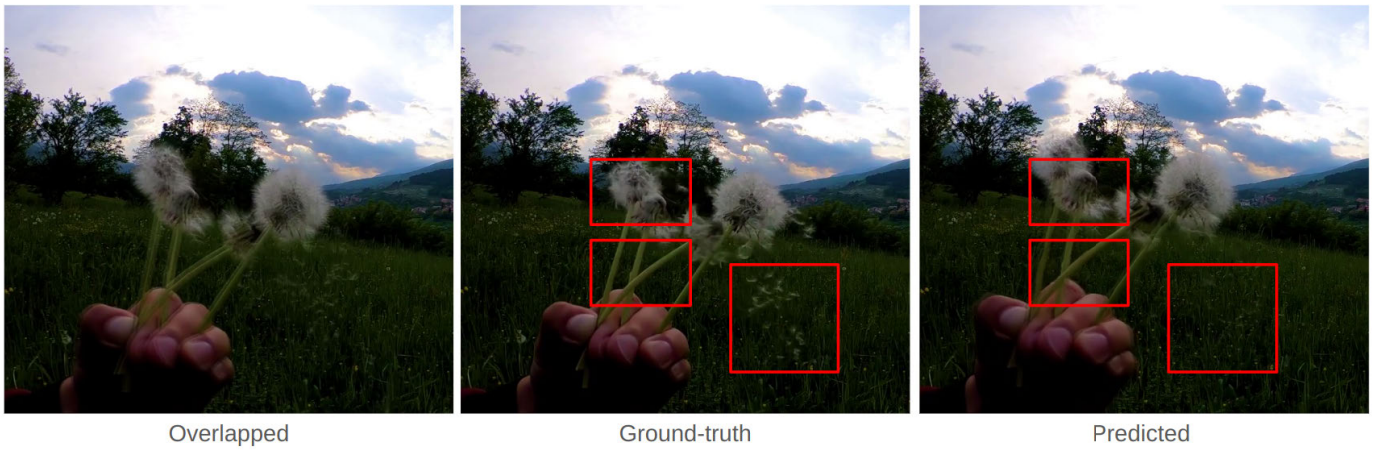


Fig. 6. Example results showing the limitations of LD4SFI. This video frames show dandelion seeds fluttering in the wind. It contains tiny movements of a slightly transparent object. The marked areas using the red-colored box highlight that the LD4SFI has failed to predict detailed visual information as suitable interpolation frames.

VI. Conclusion and Discussion

We introduced **LD4SFI**, a latent-diffusion framework for VFI that leverages a VQ-STVAE backbone, disparity-guided conditioning, and a simple DBL. Across standard benchmarks (UCF-101, DAVIS, and SNU-FILM), LD4SFI delivers *competitive or improved* accuracy relative to recent diffusion-based baselines, with robust results on UCF-101 and DAVIS, while achieving the highest measured throughput among the DMs evaluated on a single V100 in our setup. The proposed method produces 0.974 SSIM, 36.219 PSNR, and 0.016 LPIPS on the UCF-101 dataset and similarly strong results on the DAVIS dataset. On SNU-FILM, differences in LPIPS are small and, in some cases, comparable to strong baselines. Ablation studies with respect to DBL, disparity conditioning, and a sensitivity analysis over λ_{dbl} indicate that both the disparity signal and DBL contribute to quality and temporal smoothness within a reasonable range of weights.

While LD4SFI represents a significant step forward in diffusion-based VFI, we acknowledge certain limitations. Like other diffusion methods, LD4SFI remains far from real-time due to iterative sampling. Furthermore, while disparity-based conditioning improves performance, its current formulation remains relatively simplistic in handling highly nonlinear or extremely tiny motion dynamics. The disparity map is insufficient to describe motion dynamics in those situations; therefore, LD4SFI performance will degrade. Fig. 6 shows VFI results on tiny motion dynamics, in which the wind shakes a flower. Since the motion is very small, the disparity map cannot represent motion between two neighboring frames, so the VFI results do not contain enough detail. However, these limitations also highlight promising research directions. Future work will focus on developing more efficient inference strategies for diffusion-based VFI, including accelerated sampling techniques and adaptive conditioning mechanisms. Additionally, we will focus on developing cost-effective but enhanced motion representation.

References

[1] C. Morris, D. Danier, F. Zhang, N. Anantrasirichai, and D. R. Bull, "ST-MFNet mini: Knowledge distillation-driven frame interpolation," 2023, *arXiv:2302.08455*.

[2] L. Kong et al., "IFRNet: Intermediate feature refine network for efficient frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1959–1968.

[3] J. Park, K. Ko, C. Lee, and C. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. 16th Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 109–125.

[4] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3532–3542.

[5] D. Danier, F. Zhang, and D. Bull, "Ldmvfi: Video frame interpolation with latent diffusion models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, 2024, pp. 1472–1480.

[6] S. Dutta, A. Subramaniam, and A. Mittal, "Non-linear motion estimation for video frame interpolation using space-time convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1725–1730.

[7] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Blurry video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5113–5122.

[8] Q. Ai, P. Wang, L. He, L. Wen, L. Pan, and Z. Xu, "Generative oversampling for imbalanced data via majority-guided VAE," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2023, pp. 3315–3330.

[9] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5315–5324.

[10] X. Jin, L. Wu, J. Chen, Y. Chen, J. Koo, and C.-H. Hahm, "A unified pyramid recurrent network for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1578–1587.

[11] H. Zhang, Y. Zhao, and R. Wang, "A flexible recurrent residual pyramid network for video frame interpolation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 474–491.

[12] M. Xu, J. Chen, H. Wang, S. Liu, G. Li, and Z. Bai, "C3DVQA: Full-reference video quality assessment with 3D convolutional neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 4447–4451.

[13] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[14] D. Danier, F. Zhang, and D. Bull, "ST-MFNet: A spatio-temporal multi-flow network for frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3521–3531.

[15] J. Yu, H. Oh, and J. Yang, "Adversarial denoising diffusion model for unsupervised anomaly detection," in *Proc. Deep Generative Models Health Workshop NeurIPS*, 2023.

[16] J. Li, A. Mądry, J. Peebles, and L. Schmidt, "On the limitations of first-order approximation in GAN dynamics," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3005–3013.

[17] H. Sim, J. Oh, and M. Kim, "XVFI: Extreme video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14489–14498.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[18] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, Mar. 2011.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[20] L. P. Cinelli, M. A. Marins, E. A. B. da Silva, and S. L. Netto,*Variational autoencoder*, in *Variational Methods for Machine Learning With Applications To Deep Networks*. Cham, Switzerland: Springer, 2021, pp. 111–149.

[21] L. Yang et al., "Diffusion models: A comprehensive survey of methods and applications," 2022, *arXiv:2209.00796*.

[22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[23] W. Chan, D. J. Fleet, A. A. Gritsenko, J. Ho, M. Norouzi, and D. Fleet, "Video diffusion models," in *Proc. Adv. Neural Inform. Process. Syst.*, 2022, pp. 8633–8646.

[24] S. Jain, D. Watson, E. Tabellion, A. Hołyński, B. Poole, and J. Kontkanen, "Video interpolation with diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 7341–7351.

[25] Z. Huang et al., "Motion-aware latent diffusion models for video frame interpolation," 2024, *arXiv:2404.13534*.

[26] Z. Lyu and C. Chen, "TLB-VFI: Temporal-aware latent Brownian bridge diffusion for video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2025, pp. 16260–16269.

[27] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12873–12883.

[28] A. Van Den Oord et al., "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2270–2279.

[31] A.-M. Huang and T. Q. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 694–708, May 2008.

[32] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-compensated frame rate up-conversion—Part II: New algorithms for frame interpolation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 142–149, Jun. 2010.

[33] B. Zhao and X. Li, "Edge-aware network for flow-based video frame interpolation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1401–1408, Jan. 2024.

[34] G. Zhang, C. Liu, Y. Cui, X. Zhao, K. Ma, and L. Wang, "VFIMamba: Video frame interpolation with state space models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 107225–107248.

[35] M. W. Jeong and C. E. Rhee, "LC-mamba: Local and continuous mamba with shifted windows for frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 17671–17681.

[36] W. Seo, J. Oh, and M. Kim, "BiM-VFI: Bidirectional motion field-guided frame interpolation for video with non-uniform motions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 7244–7253.

[37] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2020, *arXiv:2010.02502*.

[38] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *J. Mach. Learn. Res.*, vol. 23, no. 47, pp. 1–33, 2021.

[39] Z. Zhang et al., "EDEN: Enhanced diffusion for high-quality large-motion video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 2105–2115.

[40] Y. Hai, G. Wang, T. Su, W. Jiang, and Y. Hu, "Hierarchical flow diffusion for efficient frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 22943–22952.

[41] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.

[42] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2020, *arXiv:2011.13456*.

[43] T. Liu, G. Greenberg, and I. Shomorony, "CVQVAE: A representation learning based method for multi-omics single cell data integration," in *Proc. Mach. Learn. Comput. Biol.*, 2022, pp. 1–15.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[46] X. Cheng and Z. Chen, "Video frame interpolation via deformable separable convolution," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10607–10614.

[47] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9000–9008.

[48] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015, *arXiv:1508.06576*.

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1125–1134.

[50] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[51] D. M. Argaw and I. S. Kweon, "Long-term video frame interpolation via feature propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3533–3542.

[52] D. Ma, F. Zhang, and D. R. Bull, "BVI-DVC: A training database for deep video compression," *IEEE Trans. Multimedia*, vol. 24, pp. 3847–3858, 2022.

[53] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[54] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[55] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, "Channel attention is all you need for video frame interpolation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10663–10671.

[56] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, "The 2019 DAVIS challenge on VOS: Unsupervised multi-object segmentation," 2019, *arXiv:1905.00737*.

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[58] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[59] D. Danier, F. Zhang, and D. Bull, "FloLPIPS: A bespoke video quality metric for frame interpolation," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2022, pp. 283–287.

[60] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 30, 2017, pp. 6626–6637.

[61] T. Ding, L. Liang, Z. Zhu, and I. Zharkov, "CDFI: Compression-driven network design for frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7997–8007.

[62] J. Park, C. Lee, and C.-S. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14519–14528.

[63] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, "FLAVR: Flow-agnostic video representations for fast frame interpolation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 2070–2081.

[64] X. Jin et al., "UPR-Net: A unified pyramid recurrent network for video frame interpolation," *Int. J. Comput. Vis.*, vol. 133, no. 1, pp. 16–30, Jan. 2025.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on October 29,2026 at 20:20:04 UTC from IEEE Xplore. Restrictions apply.