LDMVFI: Video Frame Interpolation with Latent Diffusion Models

Duolikun Danier Fan Zhang David Bull
University of Bristol
{duolikun.danier, fan.zhang, dave.bull}@bristol.ac.uk

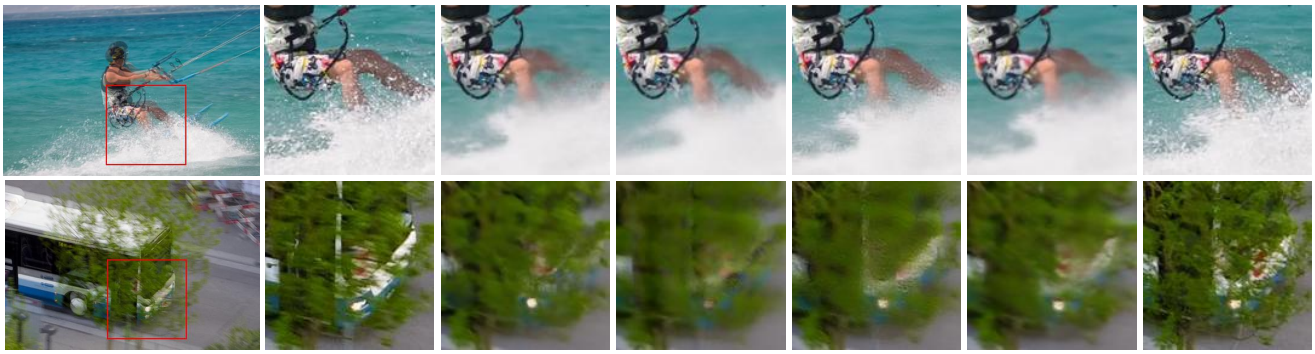


Figure 1. Visual examples of frames interpolated by the state-of-the-art methods and the proposed LDMVFI. Under large and complex motions, our method preserves the most high-frequency details, delivering superior perceptual quality.

Abstract

Existing works on video frame interpolation (VFI) mostly employ deep neural networks trained to minimize the L1 or L2 distance between their outputs and ground-truth frames. Despite recent advances, existing VFI methods tend to produce perceptually inferior results, particularly for challenging scenarios including large motions and dynamic textures. Towards developing perceptually-oriented VFI methods, we propose latent diffusion model-based VFI, LDMVFI. This approaches the VFI problem from a generative perspective by formulating it as a conditional generation problem. As the first effort to address VFI using latent diffusion models, we rigorously benchmark our method following the common evaluation protocol adopted in the existing VFI literature. Our quantitative experiments and user study indicate that LDMVFI is able to interpolate video content with superior perceptual quality compared to the state of the art, even in the high-resolution regime. Our source code will be made available here.

1. Introduction

The purpose of video frame interpolation (VFI) is to generate intermediate frames between two existing consecutive frames in a video sequence. It is commonly used to synthetically increase frame rate, for example to generate jitter-free slow-motion content [29]. VFI has also been used in video compression [72], novel view synthesis [18], medical imaging [31] and animation production [62].

Existing VFI methods [29, 74, 2, 50, 37, 5, 15, 47, 30, 51, 35, 41, 59] are mostly based on deep neural networks. These deep models differ in architectural designs and motion modeling approaches, but are mainly PSNR-oriented in the sense that they are trained to minimize the L1 or L2 distance between their outputs and the ground-truth intermediate frames. Although some previous works [49, 46, 11] have used feature distortion-based [61] or GAN-based [19] losses to fine-tune the PSNR-oriented models, the major contributor to the optimization objective (and hence the final performance of the model) is still the L1/L2-based distortion loss. As a result, it has been reported [12] that existing methods, while achieving high PSNR values, tend to under-perform perceptually, especially under challenging scenarios including large motion and dynamic textures.

In contrast, diffusion models [22, 65, 24, 55] have recently shown remarkable performance in generating realistic, perceptually-optimized images and videos. Diffusion models have reportedly outperformed other generative models including GANs and VAEs [14, 23]. However, despite their strong ability to synthesize high-fidelity visual content, the application of diffusion models for VFI and

their potential advantages over current PSNR-oriented VFI approaches have not been fully investigated.

In this work, we develop a latent diffusion model for video frame interpolation (LDMVFI), where we formulate VFI as a conditional image generation problem. Specifically, we adopt the recently proposed *latent diffusion models* [55] (LDMs, a.k.a. Stable Diffusion) within a framework comprising an autoencoding model that projects images into a latent space, and a denoising U-Net which performs reverse diffusion process in that latent space. To better adapt LDMs to VFI, we devise VFI-specific components, notably a novel vector quantization-based VFI-autoencoding model, VQ-FIGAN, with which our method shows superior performance over vanilla LDMs.

Despite the paradigmatic shift from mainstream PSNR-oriented VFI methods, we adhere to the commonly adopted VFI benchmarking protocol and evaluate the proposed method on various VFI test sets, covering both low and high resolution contents (up to 1920×1080). Our results indicate that LDMVFI performs favorably against the state of the art in terms of three perceptual metrics [76, 25, 10]. We also conducted a user study to collect subjective judgments on the quality of full HD videos interpolated by our method and several competitive counterparts to further confirm the superior perceptual performance of LDMVFI.

To the best of our knowledge, this work is the first attempt to address VFI as a conditional generation problem using latent diffusion models, as well as the first to demonstrate the perceptual superiority of diffusion-based VFI over the state-of-the-art PSNR-based approaches. Our contributions are summarized below.

- We present LDMVFI, a novel latent diffusion model for VFI that leverages the high-fidelity image synthesis ability of diffusion models to perform conditional generation.
- We design novel VFI-specific components in LDMs, including a vector-quantized autoencoding model, VQ-FIGAN, which further enhances VFI performance.
- We demonstrate through quantitative and subjective experiments that the proposed method produces perceptually superior interpolation results over the state of the art.

2. Relate Work

Video Frame Interpolation. Existing VFI approaches are mostly based on deep learning, and can be generally categorized as flow-based or kernel-based. Flow-based methods rely on optical flow estimation to generate interpolated frames. To obtain the optical flow from input frames to the non-existent middle frame (or the other way around), some methods [29, 73, 46, 47, 60] assume certain motion types to infer the intermediate optical flows using the flows between two input frames, while others [39, 74, 50, 51, 41, 35, 53, 26] directly estimate the intermediate flows. On the other hand, kernel-based methods argue that optical flows can be unreliable in dynamic texture scenes, so they predict locally adaptive convolution kernels to synthesize output pixels, allowing more flexible many-to-one mapping. While earlier methods [48, 49] in this class predict fixed-size kernels, more recent ones [37, 58, 15, 5, 6, 20, 9, 59] tend to adopt deformable convolution [8] kernels. Other than these two classes, there are also attempts to combine flows and kernels [2, 3, 11], and to perform end-to-end frame synthesis [7, 30].

It is noted that the above methods are trained to optimize PSNR-oriented loss functions, i.e., the L1/L2 distance between the model outputs and the ground-truth frames. Although this results in reasonably good PSNR performance, it has been previously reported [12] that PSNR does not fully reflect the perceptual quality of interpolated videos, exhibiting poor correlation performance with subjective ground truth. To improve perceptual performance, some existing methods [46, 47] use the VGG [61] feature-based loss in combination with the L1 loss. However, we observe two limitations of this approach. Firstly, it was shown in previous works [12] that such deep feature-based distances also fail to predict the perceived quality of VFI-generated videos. Secondly, in these models, the L1 loss still plays a more important role (compared to the VGG loss) during the optimization process, constraining the perceptual performance. An alternative approach uses GANs [37, 11] to enhance perceptual quality of interpolated videos. However, due to the instability of GAN training [19], these models are typically pre-trained using L1 loss, then fine-tuned with an adversarial loss. There is thus only limited impact of the additional adversarial fine-tuning on the final perceptual quality.

Diffusion Models. Recently, diffusion models (DMs) [22, 65, 55, 38, 24] have demonstrated remarkable performance in synthesizing high-fidelity images and videos. In their original form [22], DMs generate new images by progressively denoising a Gaussian noise image; the process corresponds to the reverse of a Markov chain that gradually adds noise to a clean image. DMs have been reported to offer superior performance [14, 23] compared to GANs [19] and VAEs [34] in image generation tasks. The only previous application of DMs on VFI is [70], but this work focused on low-resolution images and the model lacked any VFI-specific innovations, showing limited interpolation performance. The recently proposed latent diffusion models (LDMs) have demonstrated strong ability to synthesize high-resolution images by performing diffusion processes in latent space. However, we observe that LDMs have not previously been exploited for VFI.

3. Proposed Method: LDMVFI

Given two consecutive frames I^0, I^1 from a video, VFI aims to generate the non-existent intermediate frame I^n

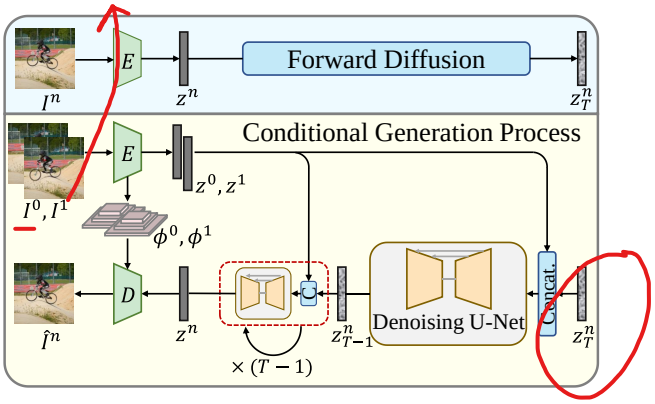


Figure 2. Overview of the diffusion processes in LDMVFI. The encoder and decoder enable projection between image and latent spaces, and the diffusion processes take place in the latent space.

where $n = 0.5$ for $\times 2$ upsampling. Approaching VFI from a generative perspective, our goal is to learn a parametric approximation of the conditional distribution $p(I^n|I^0, I^1)$ using a dataset $\mathcal{D} = \{I_s^0, I_s^n, I_s^1\}_{s=1}^S$. To achieve this, we adopt the latent diffusion models [55] to perform conditional generation for VFI. The proposed LDMVFI contains two main components: (i) a VFI-specific **autoencoding model**, VQ-FIGAN, that projects frames into a latent space, and reconstructs the target frame from the latent encoding; (ii) a **denoising U-Net** that performs reverse diffusion process in the latent space for conditional image generation. Figure 2 shows the overview of the LDMVFI.

3.1. Latent Diffusion Models

Before presenting LDMVFI, we briefly describe the latent diffusion model (LDMs) [55]. LDMs are built upon the denoising diffusion probabilistic models [22] (referred to as diffusion models in this paper), which are a class of generative models that approximates a data distribution $p(x)$ by learning the reverse conditional probability distributions of a pre-defined Markov chain. DMs comprise two processes: **forward diffusion** and **reverse diffusion**.

Formally, starting from a "clean" image x_0, we can define a Markov chain q of length T which gradually adds Gaussian noise to x_0 according to a noise schedule $\{\beta_t\}_{t=1}^T$:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \quad (1)$$

Here the noise schedule $\{\beta_t\}_{t=1}^T$ is designed such that $q(x_T|x_0) \approx \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$, i.e. as the forward diffusion process finishes, the last state of the image becomes close to a pure Gaussian noise.

Since the reverse process $q(x_{t-1}|x_t)$ of the Markov chain above is intractable, to sample (generate) images, we can use a θ-parameterized Gaussian distribution $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2\mathbf{I})$ to approximate these reverse conditional probabilities (provided that β_t is sufficiently small in each forward step [63]), where σ_t^2 can be set to a value based on β_t [22, 38, 55], and μ_θ is realized through a neural network. Then, to perform unconditional generation, we can start from a Gaussian noise $x_T \sim \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$ and sample the less noisy versions of the image from $p_\theta(x_{t-1}|x_t)$ until x_0. Using the parameterization in [22], instead of learning to predict the mean $\mu_\theta(x_t, t)$, we can train a neural network to predict the noise $\epsilon_\theta(x_t, t)$ using the objective

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_0, \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}), t\sim\mathcal{U}(1,T)}\left[\, \|\epsilon - \epsilon_\theta(x_t, t)\|^2 \,\right], \quad (2)$$

where x_t is sampled from the forward diffusion process and $\mathcal{U}(1, T)$ denotes uniform distribution over $\{1, 2, \ldots, T\}$. This corresponds to a reweighted variant of the variational lower bound on the log-likelihood $\log p_\theta(x_0)$. We provide the derivation of the loss function in Appendix A.

Under the conditional generation setting, we can approximate the conditional reverse diffusion process $p_\theta(x_{t-1}|x_t, y)$, where the condition y can denote the two input frames in the context of VFI. Accordingly, the noise-prediction network, $\epsilon_\theta(x_t, t, y)$, can be trained for sampling from the reverse diffusion process.

Latent diffusion models contain an image encoder $E : x \mapsto z$ that encodes an image x into a lower-dimensional latent representation z, and a decoder $D : z \mapsto x$ that reconstructs the image x. The forward and reverse diffusion processes then happen in the latent space, and the training objective for learning the reverse diffusion process becomes

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{E(x_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}), t\sim\mathcal{U}(1,T)}\left[\, \|\epsilon - \epsilon_\theta(z_t, t)\|^2 \,\right]. \quad (3)$$

By projecting images into a compact latent space, LDMs allow the diffusion process to concentrate on the semantically important portions of the data and enable more computationally efficient sampling process.

3.2. Autoencoding with VQ-FIGAN

In the original form of LDMs, the autoencoding model $\{E, D\}$ is considered as a perceptual image codec. Its design purpose is to project images into efficient latent representations where high-frequency details are removed during encoding and recovered in the decoding. However, in the context of VFI, such information is likely to affect the perceived quality of the interpolated videos, so the limited reconstruction ability of the decoder can negatively impact the VFI performance. To enhance high-frequency detail recovering, we propose a VFI-specific autoencoding model: VQ-FIGAN, which is illustrated in Figure 3. While the backbone model is similar to the original VQGAN [17] used in [55], there are three major differences.

Firstly, we take advantage of a property of VFI problems - that the neighboring frames are available during inference, in order to design a frame-aided decoder. Specifically, given the ground-truth target frame $I^n \in \mathbb{R}^{H\times W\times 3}$,

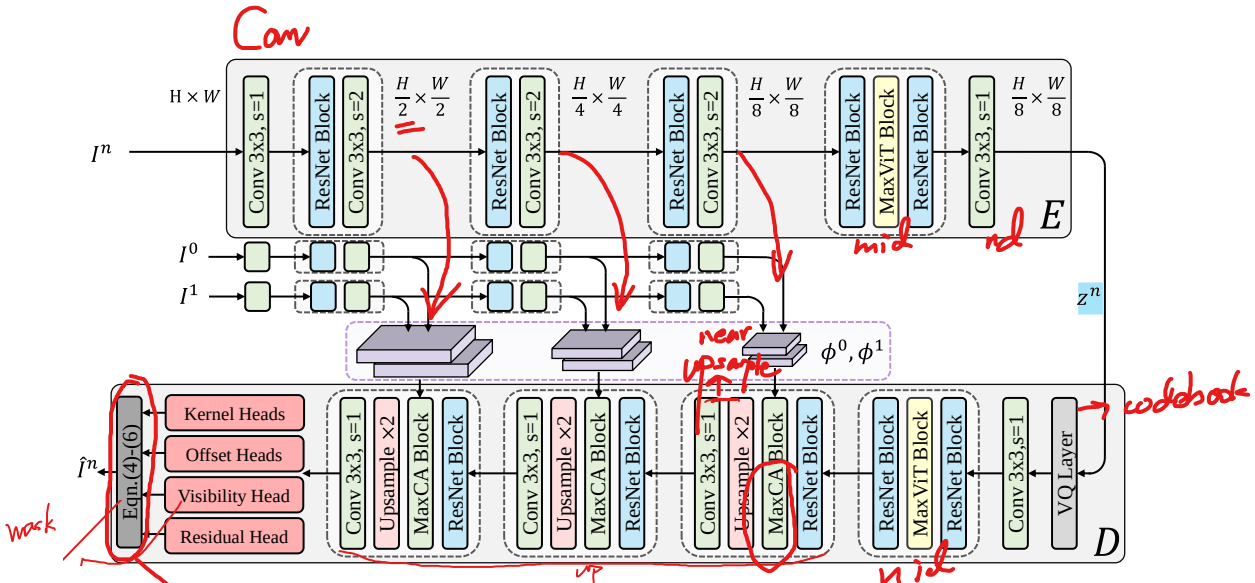


Figure 3. The architecture of the VFI autoencoding model, VQ-FIGAN. It differs from the original VQGAN [17] in three aspects: (i) the use of features extracted by the encoder from neighboring frames during the decoding via MaxViT-based cross attention; (ii) use of more efficient MaxViT block instead of the vanilla self-attention; (iii) frame reconstruction via locally adaptive deformable convolution. The kernel, offset, visibility and residual heads contain $3\times\{\texttt{conv3x3, ReLU}\}$.

the encoder E produces the latent encoding $z^n = E(I^n)$, where $z^n \in \mathbb{R}^{\frac{H}{f}\times\frac{W}{f}\times 3}$, and f is a hyper-parameter (Figure 3 shows the case for $f = 8$). Then, the decoder D outputs a reconstructed frame, $\hat{I}^n$, taking input z^n, as well as the feature pyramids, ϕ^0, ϕ^1, extracted by E from two neighboring frames, I^0, I^1. During decoding, these feature pyramids are fused with the decoded features from z^n at multiple layers using the MaxCA blocks, which are newly designed MaxViT [67]-based Cross Attention blocks, where the query embeddings for the attention mechanism are generated using decoded features from z^n, and the key and value embeddings are obtained from ϕ^0, ϕ^1.

Secondly, we notice that the original VQGAN makes use of the full self-attention [69] as in ViT[16], which can be computationally heavy (quadratic complexity) and memory-intensive, especially when the image resolution is high. For more efficient inference on high-resolution (e.g. full HD) videos, we employ the recently proposed MaxViT block [67] to perform self-attention. The multi-axis self-attention layer in the MaxViT block combines windowed attention and dilated grid attention to perform both local and global operations, while achieving linear complexity with respect to the input size.

Thirdly, instead of having the decoder directly output the reconstructed image $\hat{I}^n$, the proposed VQ-FIGAN outputs the deformable convolution-based interpolation kernels [8, 37, 5] to enhance VFI performance. Specifically, given that H, W are the frame height and width, the output of the decoder network contains parameters of the locally adaptive deformable convolution kernels of size $K \times K$: $\{\Omega^\tau, \alpha^\tau, \beta^\tau\}$ where $\tau = 0, 1$ indexes the input frames. Here $\Omega \in [0,1]^{H\times W\times K\times K}$ contains the weights of the kernels, and $\alpha, \beta \in \mathbb{R}^{H\times W\times K\times K}$ are their spatial offsets (horizontal and vertical respectively). The decoder also outputs a visibility map $v \in [0,1]^{H\times W}$ to account for occlusion [29], and a residual map $\delta \in \mathbb{R}^{H\times W}$ to further enhance VFI performance [6]. To generate the interpolated frame, firstly locally adaptive deformable convolutions are performed for each input frame $\{I^\tau\}_{\tau=0,1}$:

$$I^{n\tau}(h,w) = \sum_{i=1}^{K}\sum_{j=1}^{K} \Omega^\tau_{h,w}(i,j) \cdot P^\tau_{h,w}(i,j), \quad (4)$$

$$P^\tau_{h,w}(i,j) = I^\tau(h + \alpha^\tau_{h,w}(i,j), w + \beta^\tau_{h,w}(i,j)), \quad (5)$$

in which $I^{n\tau}$ denotes the result obtained from I^τ, and $P^\tau_{h,w}$ is the patch sampled from I^τ for output location (h,w). These intermediate results are then combined using the visibility and residual maps:

$$\hat{I}^n = v \cdot I^{n0} + (1 - v) \cdot I^{n1} + \delta. \quad (6)$$

We adopt the separable deformable convolution implementation in [5] which exploits separability properties of kernels [54] to reduce memory requirements while maintaining VFI performance.

Training VQ-FIGAN. We follow the original training settings of VQGAN in [17, 55], where the loss function consists of an LPIPS-based [76] perceptual loss, a patch-based adversarial loss [27] and a latent regularization term based on a vector quantization (VQ) layer [68]. We refer the readers to [17] for details on training VQGANs.

3.3. Conditional Generation with LDM

The trained VQ-FIGAN allows us to access a compact latent space in which we perform forward diffusion by grad-

Algorithm 1 Training

1: **Input**: dataset $\mathcal{D} = \{I_s^0, I_s^n, I_s^1\}_{s=1}^S$ of consecutive frame triplets, maximum diffusion step T, noise schedule $\{\beta_t\}_{t=1}^T$
2: **Load**: pre-trained VQ-FIGAN encoder E
3: **Initialize**: denoising U-Net ϵ_θ
4: **repeat**
5: Sample $(I^0, I^n, I^1) \sim \mathcal{D}$
6: Encode $z^0 = E(I^0), z^n = E(I^n), z^1 = E(I^1)$
7: Sample $t \sim \mathcal{U}(1, T)$
8: Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
9: Sample z_t^n from forward diffusion step t using ϵ, t, β_t
10: Take a gradient descent step on
$$\nabla_\theta \left\| \epsilon - \epsilon_\theta(z_t^n, t, z^0, z^1) \right\|^2$$
11: **until** converged

Algorithm 2 Inference

1: **Input**: Original frames I^0, I^1, noise schedule $\{\beta_t\}_{t=1}^T$, maximum diffusion step T
2: **Load**: pre-trained denoising U-Net ϵ_θ, VQ-FIGAN encoder E and decoder D
3: Sample $z_T^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: Encode $z^0 = E(I^0), z^1 = E(I^1)$ and store features ϕ^0, ϕ^1
5: **for** $t = T, \ldots, 1$ **do**
6: Predict noise $\hat{\epsilon} = \epsilon_\theta(z_t^n, t, z^0, z^1)$
7: Compute the mean μ_θ from $\hat{\epsilon}$
8: Compute the standard deviation σ_t from β_t
9: Sample z_{t-1}^n from $p_\theta(z_{t-1}^n | z_t^n) = \mathcal{N}(\mu_\theta, \sigma_t^2 \mathbf{I})$
10: **end for**
11: **return** $\hat{I}^n = D(z_0^n, \phi^0, \phi^1)$ as the interpolated frame

ually adding Gaussian noise to the latent z^n of the target frame I^n according to a pre-defined noise schedule, and learn the reverse (denoising) process to perform conditional generation. To this end, we adopt the noise-prediction parameterization [22] of DMs and train a denoising U-Net by minimizing the re-weighted variational lower bound on the conditional log-likelihood $\log p_\theta(z^n | z^0, z^1)$ where z^0, z^1 are the latent encodings of the two input frames.

Training. Specifically, the denoising U-Net ϵ_θ takes as input the "noisy" latent encoding z_t^n for the target frame I^n (sampled from the t-th step in the forward diffusion process), the diffusion step t, as well as the conditioning latents z^0, z^1 for the input frames I^0, I^1. It is trained to predict the noise added to z^n at each time step t by minimizing

$$\mathcal{L} = \mathbb{E}_{z^n, z^0, z^1, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} \left[\left\| \epsilon - \epsilon_\theta(z_t^n, t, z^0, z^1) \right\|^2 \right], \quad (7)$$

where $t \sim \mathcal{U}(1, T)$. The training procedure of ϵ_θ is summarized in Algorithm 1, and we provide the derivation and full details of this process in Appendix B. Intuitively, the training is performed by alternately adding a random noise to z^n using the forward diffusion Markov chain (lines 7-9), and having the network ϵ_θ predict the noise added given the step t, conditioning on z^0, z^1 (line 10).

Inference. To interpolate $\hat{I}^n$ from I^0, I^1, as shown in Algorithm 2, we start by sampling a Gaussian noise z_T^n in the latent space (line 3), and perform T steps of denoising until we obtain z_0^n (lines 5-10). Within each step, firstly the network ϵ_θ predicts the noise $\hat{\epsilon}$ (line 6) which is used to calculate the mean μ_θ of the approximated reverse conditional $p_\theta(z_{t-1}^n | z_t^n)$ (line 7), and the variance σ_t^2 is obtained based on β_t (line 8). Then z_{t-1}^n is sampled from this distribution (line 9). Finally, the decoder D produces the interpolated frame (line 11) from the denoised latent z_0^n, with the help of feature pyramids ϕ^0, ϕ^1 extracted by the encoder E from I^0, I^1 (line 4). See full details in Appendix B.

Network Architecture. We employ the time-conditioned U-Net [56] as in [55] for ϵ_θ, but with one modification: all the vanilla self-attention blocks [69] are replaced with the aforementioned MaxViT blocks [67] for computational efficiency. The conditioning mechanism for in U-Net is a simple concatenation of z_t^n and z^0, z^1 at the input. See the full architecture in Appendix D.

4. Experimental setup

Implementation Details. We set the downsampling factor of VQ-FIGAN to $f = 32$, by repeating the `ResNetBlock+Conv3x3` layer in Figure 3 twice. The size of the kernels output by the decoder is $K = 5$. Regarding the diffusion processes, following [55], we adopt a linear noise schedule and a codebook size of 8192 for vector quantization in VQ-FIGAN. We sample from all diffusion models with the DDIM [64] sampler for 200 steps (details provided in Appendix C. We also follow [55] to train the VQ-FIGAN using the ADAM [33] optimizer and the denoising U-Net using the Adam-W optimizer [40], with the initial learning rates set to 10^{-5} and 10^{-6} respectively. All models were trained until convergence, which corresponds to around 70 epochs for VQ-FIGAN, and around 60 epochs for the U-Net. NVIDIA RTX 3090 GPUs were used for all training and evaluation.

Training Dataset. Considering the limited motion diversity and magnitudes in the commonly adopted Vimeo90k dataset [74], we follow [11] to additionally incorporate the BVI-DVC [42] quintuplets. The final training set consists of 64612 frame septuplets from Vimeo90k and 17600 frame quintuplets from BVI-DVC provided by [11]. Since we need only a frame triplet (I^0, I^n, I^1) to train the model, for each septuplet or quintuplet, we only use the three frames in the center. For data augmentation, we randomly crop 256×256 patches from the triplets, and perform random flipping and temporal order reversing.

Test Datasets. We evaluate models on the most commonly used VFI benchmarks, including Middlebury [1], UCF-101 [66], DAVIS [52], and SNU-FILM [7] datasets.

	Middlebury			UCF-101			DAVIS			RT (sec)	#P (M)
	PSNR↑	LPIPS↓	FloLPIPS↓	PSNR↑	LPIPS↓	FloLPIPS↓	PSNR↑	LPIPS↓	FloLPIPS↓		
BMBC	36.368	0.023	0.046	32.576	0.034	0.045	26.835	0.125	0.185	0.51	11.0
AdaCoF	35.256	0.031	0.052	32.488	0.034	0.046	26.234	0.148	0.198	0.01	21.8
CDFI	36.205	0.022	0.048	32.541	0.036	0.049	26.471	0.157	0.211	0.02	5.0
XVFI	34.724	0.036	0.070	32.224	0.038	0.050	26.475	0.129	0.185	0.08	5.6
ABME	37.639	0.027	0.040	32.055	0.058	0.069	26.861	0.151	0.209	0.27	18.1
IFRNet	36.368	0.020	0.045	32.716	0.032	0.044	27.313	0.114	0.170	0.02	5.0
VFIformer	35.566	0.031	0.065	32.745	0.039	0.051	26.241	0.191	0.242	1.74	5.0
ST-MFNet	N/A	N/A	N/A	33.384	0.036	0.049	28.287	0.125	0.181	0.14	21.0
FLAVR	N/A	N/A	N/A	33.224	0.035	0.046	27.104	0.209	0.248	0.02	42.1
MCVD	20.539	0.123	0.138	18.775	0.155	0.169	18.946	0.247	0.293	52.55	27.3
LDMVFI	34.033	0.019	0.044	32.186	0.026	0.035	25.541	0.107	0.153	8.48	439.0

Table 1. Quantitative comparison results of LDMVFI ($f = 32$) and 10 tested methods on Middlebury, UCF-101, DAVIS. Note ST-MFNet and FLAVR require four input frames cannot be evaluated on Middlebury dataset which contains frame triplets. For each column, we highlight the best result red and the second best in blue. The last two columns show the average run time (RT) needed to to interpolate one 480p frame, and the number of parameters (#P) in each model.

	SNU-FILM-Easy			SNU-FILM-Medium			SNU-FILM-Hard			SNU-FILM-Extreme		
	PSNR↑	LPIPS↓	FloLPIPS↓	PSNR↑	LPIPS↓	FloLPIPS↓	PSNR↑	LPIPS↓	FloLPIPS↓	PSNR↑	LPIPS↓	FloLPIPS
BMBC	39.809	0.020	0.031	35.437	0.034	0.059	29.942	0.068	0.118	24.715	0.145	0.237
AdaCoF	39.632	0.021	0.033	34.919	0.039	0.066	29.477	0.080	0.131	24.650	0.152	0.234
CDFI	39.881	0.020	0.031	35.224	0.036	0.066	29.660	0.081	0.141	24.645	0.163	0.255
XVFI	38.903	0.022	0.037	34.552	0.039	0.072	29.364	0.075	0.138	24.545	0.142	0.233
ABME	39.697	0.022	0.034	35.280	0.042	0.076	29.643	0.092	0.168	24.541	0.182	0.300
IFRNet	39.881	0.019	0.030	35.668	0.033	0.058	30.143	0.065	0.122	24.954	0.136	0.229
ST-MFNet	40.775	0.019	0.031	37.111	0.036	0.061	31.698	0.073	0.123	25.810	0.148	0.238
FLAVR	40.161	0.022	0.034	36.020	0.049	0.077	30.577	0.112	0.169	25.206	0.217	0.303
MCVD	22.201	0.199	0.230	21.488	0.213	0.243	20.314	0.250	0.292	18.464	0.320	0.385
LDMVFI	38.674	0.014	0.024	33.996	0.028	0.053	28.547	0.060	0.114	23.934	0.123	0.204

Table 2. Quantitative comparison results on SNU-FILM (note VFIformer is not included because the GPU goes out of memory).

These test sets cover spatial resolutions from 225 × 225 up to 1280 × 720, and various levels of VFI difficulties. To further assess the perceptual performance on full HD (1920 × 1080) content, we make use of the BVI-HFR [43] dataset which covers various texture and motion types. This dataset is chosen because it contains relatively long videos which facilitate a rigorous user study, while other full HD video datasets [60, 44] used for VFI evaluation contain much shorter clips.

Evaluation Methods. As the main focus of this work is on improving the perceptual quality of interpolated content, we adopt a perceptual image quality metric LPIPS [76], and a bespoke VFI metric, FloLPIPS [10] for performance evaluation. These metrics have shown superior correlation with human judgments of VFI quality compared to commonly used quality measurements, PSNR and SSIM [71]. For completeness, we still provide benchmark results based on PSNR and SSIM, but they are limited in reflecting the perceptual quality of interpolated content [12] and are therefore not the focus of the paper. To measure the true perceptual performance of the VFI methods, we also conducted a psychophysical subjective experiment, in which the proposed method was compared against the state of the art (see Sec. 5.2).

5. Experiments

5.1. Quantitative Evaluation

The proposed LDMVFI was compared against 10 recent state-of-the-art VFI methods, including BMBC [50], AdaCoF [37], CDFI [15], XVFI [60], ABME [51], IFRNet [35], VFIformer [41], ST-MFNet [11], FLAVR [30], and MCVD [70]. It it noted that MCVD is the only existing diffusion-based VFI method. All these models were re-trained on our training dataset for fair comparison.

Performance. Table 1 shows the performance of the evaluated methods on the Middlebury, UCF-101 and DAVIS test sets in terms of three perceptual metrics and PSNR (SSIM results are included in Appendix E). It is observed from the table that although the PSNR performance of LDMVFI is not comparable to other methods, LDMVFI outperforms all the other VFI methods on the perceptual metrics. The model performance on the four splits of the SNU-FILM dataset measured by the perceptual metrics are summarized in Table 2, which again demonstrates the superior perceptual quality of videos interpolated by LDMVFI. It is also noticed that the other diffusion-based VFI method, MCVD, does not offer satisfactory overall performance,

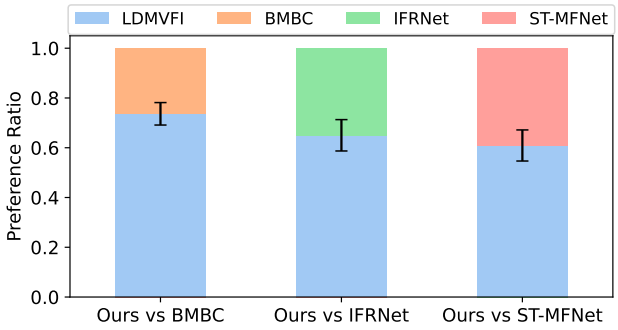


Figure 4. Results of the user study in terms of preference ratio. Error bar reflects the standard error over test sequences.

which implies that directly applying the original formulation of diffusion models to VFI is not sufficient to get enhanced performance. This further demonstrates the effectiveness of the LDMVFI design. The full evaluation results in terms of all five metrics for all test sets are presented in Appendix E.

Complexity. The average time taken to interpolate a 480p frame on an RTX 3090 GPU and the number of parameters of each model are presented in Table 1. It is observed that the inference speed of LDMVFI is much lower compared to other methods, and this is mainly due to the iterative denoising operation performed during sampling. This is a common drawback of existing diffusion models. Various methods have been proposed to speed up sampling process of diffusion models [57, 32], which can also be applied to LDMVFI. The number of parameters in LDMVFI is also large, and this is because we adopted (with some modifications, see Sec. 3.2) the existing denoising U-Net [55] designed for generic image generation. We leave the design of a more efficient denoising U-Net and the improvement of LDMVFI sampling speed as future work. More discussion on the limitations is included in Appendix H.

5.2. Subjective Experiment

To further confirm the superior perceptual quality of the videos interpolated by LDMVFI compared to the state of the art, and also to measure the temporal consistency of LDMVFI, we conducted a subjective experiment where human participants were hired to rate the quality of videos interpolated by ours and competing methods.

Test Videos. We use the 22 high-quality full HD 30fps videos from the BVI-HFR [43] dataset as source content. There are two reasons for selecting this dataset over the commonly used high resolution datasets in VFI [60, 44]. Firstly, the length of the BVI-HFR videos facilitates the user study where the participants should ideally watch longer clips rather than short segments [45]. Secondly, these videos cover an excellent range of video features related to motion and texture as shown by the analysis in the original paper [43], allowing for a more thorough benchmarking of VFI methods. To generate the test content, the 22 videos were first truncated to 5 seconds (150 frames) following [45]. Then we used four different VFI methods to interpolate all videos to 60fps. Other than LDMVFI, the tested methods include ST-MFNet, IFRNet and BMBC, which showed the most competitive performance on the more challenging test sets (e.g. DAVIS, SNU-FILM-extreme) in the quantitative experiment. As a result, we obtain 88 test videos generated by four VFI methods.

Test Methodology. Following previous works [39, 46, 30], the 2AFC approach is adopted for the subjective experiment, where the participant is asked to choose the one with better perceptual quality from a pair of two videos. Specifically, in each test session, the participant was displayed 66 pairs of videos where one video in each pair is interpolated by LDMVFI and the other is interpolated by ST-MFNet, IFRNet or BMBC. The display order of the 66 pairs, and the order of test videos within each pair are both randomized. The user is unaware of the methods used for generating the videos. Each pair is presented twice to the participant, who is then asked to provide an answer to the question "which of the two videos is of higher quality?". Twenty participants were hired in total. See Appendix G for more details of the user study.

Results. After collecting all the user data, for each of the 22 source sequences, the ratio of users that preferred LDMVFI is calculated. Figure 4 reports the average preference ratios for LDMVFI and the standard error over the sequences. It can be seen that in all comparisons, LDMVFI achieved higher preference ratios. T-test analysis on the sequence-wise preference ratios shows that the advantage of LDMVFI over the other three tested methods is statistically significant at 95% confidence level. These results further confirm the superior perceptual performance of LDMVFI.

Visual Examples. Figure 1 shows example blocks interpolated by LDMVFI and the competing methods, clearing demonstrating that the LDMVFI results have the best visual quality. More sample frames and videos are provided in Appendix J.

5.3. Ablation Study

In this section we experimentally validate and study different components and hyper-parameters in LDMVFI. The ablation study results are summarized in Table 3, which shows evaluation results of 8 variants of the proposed model on three test sets (see the full ablation study results in Appendix F). These variants differ in four aspects: downsampling factor f, denoising U-Net's base channel size c, the autoencoding model's architecture and the conditioning mechanism for denoising.

Effectiveness of VQ-FIGAN. To validate the effectiveness of the proposed VQ-FIGAN design, we tested two variants of the model: V1 and V2. V2 outputs the frame $\hat{I}^n$ di-

	f	c	AE Model	Cond. Mode	Middlebury		UCF-101		DAVIS	
					LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓
V1	32	256	frame	concat	0.077	0.085	0.063	0.067	0.168	0.200
V2	32	256	MaxCA+frame	concat	0.028	0.045	0.032	0.041	0.135	0.176
V3	8	256	MaxCA+kernel	concat	0.018	0.046	0.028	0.036	0.125	0.168
V4	16	256	MaxCA+kernel	concat	0.017	0.037	0.026	0.035	0.107	0.154
V5	64	256	MaxCA+kernel	concat	0.026	0.049	0.033	0.039	0.131	0.172
V6	32	64	MaxCA+kernel	concat	0.020	0.048	0.029	0.037	0.133	0.179
V7	32	128	MaxCA+kernel	concat	0.019	0.048	0.028	0.036	0.131	0.175
V8	32	256	MaxCA+kernel	MaxCA	0.019	0.036	0.027	0.036	0.108	0.158
Ours	32	256	MaxCA+kernel	concat	0.019	0.044	0.026	0.035	0.107	0.153

Table 3. Ablation experiment results, showing performance of variants of the proposed LDMVFI.

rectly instead of predicting the deformable kernels, i.e. the convolutional heads after the last `Conv3x3` layer are removed (see Figure 3). For V1, a further change is made by removing the feature-aided reconstruction process that involves ϕ^0, ϕ^1 and replacing `MaxCABlocks` in the decoder with `ResNetBlocks`. As such, there is no information from the neighbor frames during decoding. Note that V1 is similar to the original VQGAN [17]. Table 3 shows that without the deformable convolution-based synthesis, the performance of V2 sees an evident decrease. Furthermore, V1 shows a more severe drop in performance indicating the effectiveness of using features of neighbor frames during reconstruction.

Downsampling Factor f. Here we study how the dimension of the latent space affects the VFI performance. Specifically, to obtain V4 ($f = 16$), we remove one `ResNetBlock+Conv3x3` layer and one `ResNetBlock+MaxCABlock+Upsample+Conv3x3` layer from the encoder and decoder of VQ-FIGAN respectively (see Figure 3). We repeat this process once to obtain V3 ($f = 8$). To create V5 ($f = 64$) we add these layers instead of removing them. Table 3 shows that as f increases from 8 to 32, there is generally an increasing trend in model performance (except that V4 outperformed LDMVFI on Middlebury). Such improvement is more obvious on DAVIS which mainly contains challenging large motion content. However, looking at $f = 64$, the general performance deteriorates significantly. The reason for this can be that to a reasonable extent, increasing f allows the VQ-FIGAN decoder to make use of more information from the neighboring frames which can benefit frame interpolation, while preserving sufficient information for conditional generation in the latent space. However, if the downsampling is too aggressive (e.g. $f = 64$), the information needed to perform reverse latent diffusion can be insufficient, resulting in degraded quality of latent generation. A similar trade-off between downsampling factor and model performance was also observed in [55].

Effect of Model Size. We also study how the model size of the denoising U-Net affects the performance. The denoising U-Net contains multiple layers of ResNet and MaxViT blocks, where the number of feature channels is a multiple of a base channel number c (detailed in Appendix D). In the default LDMVFI, $c = 256$. This corresponds to a total of 439.0M parameters. We experiment with $c = 128, 64$ (V7,V6), which reduce the model parameters to 126.8M and 48.7M respectively. Table 3 reflects a decreasing trend in model performance as c is decreased, implying that the choice of $c = 256$ is effective. However, we did not experiment with larger c for memory issues.

Conditioning Mechanism. In LDMVFI, the mechanism for conditioning the denoising U-Net on the latents z^0, z^1 is concatenation, i.e., we concatenate z_t^n, z^0, z^1 to form the U-Net input. In [55], an alternative cross attention-based conditioning mechanism is introduced. Based on this, We create a variant of LDMVFI (V8) where the conditioning is done using MaxCA blocks at different layers of the U-Net. As shown in Table 3, V8 occasionally outperformed LDMVFI, e.g. in FloLPIPS on Middlebury. Given the limited improvement, we adopted the simpler concatenation-based conditioning for LDMVFI.

6. Conclusion

In this work we propose LDMVFI, the first generative modeling approach that addresses video frame interpolation as a conditional generation problem using latent diffusion models. It contains two major components: an autoencoding model that provides access to a low-dimensional latent space, and a denoising U-Net that performs reverse diffusion on latent representations. To leverage latent diffusion models for VFI, we present several innovative designs including a VFI-specific autoencoding network VQ-FIGAN that employs efficient self-attention modules and deformable kernel-based frame synthesis techniques to deliver enhanced VFI performance. LDMVFI was comprehensively evaluated on a wide range of test sets (including full HD content) using both quantitative metrics and subjective experiments. The results demonstrate its superior perceptual performance over the state of the art.

References

[1] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011. 5, 14, 17

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun 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. 1, 2

[3] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[4] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, 2006. 17

[5] Xianhang Cheng and Zhenzhong Chen. Video frame interpolation via deformable separable convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10607–10614, 2020. 1, 2, 4

[6] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 4

[7] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. 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. 2, 5, 14, 17

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017. 2, 4

[9] Duolikun Danier, Fan Zhang, and David Bull. Enhancing deformable convolution based video frame interpolation with coarse-to-fine 3d cnn. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1396–1400. IEEE, 2022. 2

[10] 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. IEEE, 2022. 2, 6, 14

[11] 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. 1, 2, 5, 6, 18

[12] Duolikun Danier, Fan Zhang, and David Bull. A subjective quality study for video frame interpolation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1361–1365. IEEE, 2022. 1, 2, 6

[13] Emily Denton. Ethical considerations of generative ai. In *AI for Content Creation Workshop, CVPR*, volume 27, 2021. 17

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2, 13, 17

[15] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. CDFI: Compression-driven network design for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8001–8011, 2021. 1, 2, 6, 18

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 4

[17] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 3, 4, 8

[18] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016. 1

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1, 2

[20] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14004–14013, 2020. 2

[21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 17

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3, 5, 12, 13, 17

[23] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 1, 2

[24] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2

[25] Qiqi Hou, Abhijay Ghildyal, and Feng Liu. A perceptual quality metric for video frame interpolation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, pages 234–253, 2022. 2

[26] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Computer Vision–ECCV 2022:*

17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV, pages 624–642. Springer, 2022. 2

[27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 4

[28] Recommendation ITU-R BT. 500-11, methodology for the subjective assessment of the quality of television pictures,". *International Telecommunication Union, Tech. Rep*, 2002. 14

[29] 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. 1, 2, 4

[30] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. FLAVR: Flow-agnostic video representations for fast frame interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2071–2082, 2023. 1, 2, 6, 7, 18

[31] Alexandros Karargyris and Nikolaos Bourbakis. Three-dimensional reconstruction of the digestive wall in capsule endoscopy videos using elastic video interpolation. *IEEE Transactions on Medical Imaging*, 30(4):957–971, 2010. 1

[32] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 7, 17

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[34] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. 2

[35] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. IFRNet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2022. 1, 2, 6, 18

[36] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019. 17

[37] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. AdaCoF: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020. 1, 2, 4, 6, 18

[38] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 2, 3, 12

[39] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. 2, 7

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[41] 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. 1, 2, 6, 18

[42] Di Ma, Fan Zhang, and David Bull. BVI-DVC: A training database for deep video compression. *IEEE Transactions on Multimedia*, pages 1–1, 2021. 5, 17

[43] Alex Mackin, Fan Zhang, and David R Bull. A study of high frame rate video formats. *IEEE Transactions on Multimedia*, 21(6):1499–1512, 2018. 6, 7, 17

[44] Christopher Montgomery and H Lars. Xiph.org video test media (derf's collection), the xiph open source community. *Online, https://media.xiph.org/video/derf*, 1994. 6, 7

[45] Felix Mercer Moss, Ke Wang, Fan Zhang, Roland Baddeley, and David R Bull. On the optimal presentation duration for subjective video quality assessment. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(11):1977–1987, 2015. 7

[46] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 1, 2, 7

[47] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 1, 2

[48] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 2

[49] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 1, 2

[50] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 109–125. Springer, 2020. 1, 2, 6, 18

[51] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *International Conference on Computer Vision*, 2021. 1, 2, 6, 18

[52] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 724–732, 2016. 5, 14, 17

[53] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 250–266. Springer, 2022. 2

[54] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2754–2761, 2013. 4

[55] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn 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. 1, 2, 3, 4, 5, 7, 8, 12, 13, 17

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 5

[57] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 7, 17

[58] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video interpolation via generalized deformable convolution. *arXiv e-prints*, pages arXiv–2008, 2020. 2

[59] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17482–17491, 2022. 1, 2

[60] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: extreme video frame interpolation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 6, 7, 18

[61] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 2

[62] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6587–6595, 2021. 1

[63] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 3, 12

[64] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 5, 13

[65] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 2

[66] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5, 14, 17

[67] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 459–479. Springer, 2022. 4, 5, 13, 17

[68] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017. 4

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 4, 5

[70] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD - masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems*, 2022. 2, 6, 17, 18

[71] 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. 6, 14

[72] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 416–431, 2018. 1

[73] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, 2019. 2

[74] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 1, 2, 5, 17

[75] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022. 12

[76] 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. 2, 4, 6, 14

A. Details of LDMVFI Loss Function (Diffusion Part), Training, and Inference

In this section we derive the training objective of the denoising U-Net in LDMVFI which is responsible for performing conditional generation. As mentioned in the main paper, diffusion models consist of a forward diffusion and a reverse denoising process. The forward diffusion process is defined by a Markov chain that gradually adds noise to a "clean" image x_0 using a pre-defined noise schedule $\{\beta_t\}_{t=1}^T$ in T steps, with conditional probabilities

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{8}$$

$$\Rightarrow \quad q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{9}$$

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, according to the conditional independence property of Markov chain, one can sample from the forward diffusion process at an arbitrary time step t with

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{10}$$

In order to sample x_t from this distribution in practice, we can use a reparameterization

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{11}$$

Here the noise schedule $\{\beta_t\}_{t=1}^T$ is designed such that $\bar{\alpha}_T \approx 0$ and $q(x_T|x_0) \approx \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$. That is, as the forward diffusion process comes to an end, the last state of the image becomes close to a pure Gaussian noise.

Given the forward diffusion process, one could generate new samples by starting from pure Gaussian noise and sampling from the reverse conditionals $q(x_{t-1}|x_t)$. However, $q(x_{t-1}|x_t)$ is intractable, so one can use a θ-parameterized Gaussian distribution

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2\mathbf{I}), \tag{12}$$

$$\Rightarrow \quad p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{13}$$

to approximate the reverse Markov chain (provided that β_t is sufficiently small in each forward step [63]). Here μ_θ corresponds to a neural network, and σ_t^2 can be set to a value based on β_t [22, 38, 55]. One can then derive ([75, 22]) the variational lower bound on the data log-likelihood:

$$\mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(x_0)q(x_{1:T}|x_0)}\Big[-\log\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\Big] =: L, \tag{14}$$

and training can be done by minimizing L. It can be shown [63, 22] that L can be decomposed as

$$L = \mathbb{E}_q\Big[D_{\mathrm{KL}}\big(q(x_T|x_0)||p_\theta(x_T)\big) + \sum_{t=2}^{T} D_{\mathrm{KL}}\big(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)\big) - \log p_\theta(x_0|x_1)\Big]. \tag{15}$$

The first term in (15) can be ignored because the prior $p_\theta(x_T)$ can be set to a standard normal distribution, and the last term was handled by a separate decoder in [22], leaving the second term as the main focus for learning the reverse diffusion process. The $q(x_{t-1}|x_t, x_0)$ in the second term is tractable and it can be derived that

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t\mathbf{I}), \tag{16}$$

where

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \tag{17}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{18}$$

Since both $q(x_{t-1}|x_t, x_0)$ and $p_\theta(x_{t-1}|x_t)$ are Gaussian distributions, the KL-divergence in the second term in (15) takes the form

$$L_{t-1} =: D_{\mathrm{KL}}\big(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)\big) = \mathbb{E}_q\Big[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\right\|^2\Big]. \tag{19}$$

In [22], it is noted that plugging (11) into $\tilde{\mu}(x_t, x_0)$, the latter can be written as

$$\tilde{\mu}(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\big(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\big). \tag{20}$$

Therefore, it is proposed that instead of predicting $\mu_\theta(x_t, t)$ directly, one can predict the noise $\epsilon_\theta(x_t, t)$ then infer $\mu_\theta(x_t, t)$ with

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\big(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\big). \tag{21}$$

Plugging in (20) and (21) into (19), the loss then reads

$$L_{t-1} = \mathbb{E}_{x_0\sim q(x_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\Big[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta(x_t, t)\right\|^2\Big] + C \tag{22}$$

$$= \mathbb{E}_{x_0\sim q(x_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\Big[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\Big] + C, \tag{23}$$

where C absorbs terms independent of θ. It was observed in [22] that setting the multiplicative term before the norm in (23) to 1 provides improved performance, i.e.

$$L_{\mathrm{DM}} = \mathbb{E}_{x_0,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\big[\left\|\epsilon - \epsilon_\theta(x_t, t)\right\|^2\big], \tag{24}$$

which corresponds to a re-weighted version of the variational lower bound. Latent diffusion models (LDMs) adopt the similar overall framework derived above, but performs the diffusion processes in a lower-dimensional latent space provided by an autoencoding model that contains an encoder $E : x \mapsto z$ and a decoder $D : z \mapsto x$. Accordingly, the noise predictor ϵ_θ in LDMs are trained to optimize

$$L_{\mathrm{LDM}} = \mathbb{E}_{x_0,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\big[\left\|\epsilon - \epsilon_\theta(z_t, t)\right\|^2\big]. \tag{25}$$

In LDMVFI, the noise prediction also conditions on the latent encodings z^0, z^1 of the two input frames I^0, I^1, and the loss we use becomes

$$\mathcal{L} = \mathbb{E}_{z^n,z^0,z^1,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\big[\left\|\epsilon - \epsilon_\theta(z_t^n, t, z^0, z^1)\right\|^2\big] \tag{26}$$

$$= \mathbb{E}_{z^n,z^0,z^1,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\big[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}z_0^n + \sqrt{1-\bar{\alpha}_t}\epsilon, t, z^0, z^1)\right\|^2\big]. \tag{27}$$

B. Details of LDMVFI Training and Inference

The full training and inference algorithms are summarized in Algorithm 3 and 4. In addition to the algorithms presented in the main paper, here we provide detailed equations, which refer to those derived in Section A above.

C. Details of DDIM Sampling Process

As stated in the main paper, in order to sample from LDMVFI and other diffusion models, we use the DDIM [64] sampler, which has been shown to achieve sampling quality on par with the full original sampling method (Algorithm 4), but with fewer steps. We refer the reader to the original paper for details on the derivation and design of DDIM, and present the DDIM sampling procedure for LDMVFI in Algorithm 5.

D. Denoising U-Net Architecture

The high-level architecture of the denoising U-Net used in LDMVFI is shown in Figure 5. As stated in the main paper, one modification of its original version in [55] is to replace all vanilla self-attention layers with the MaxViT blocks [67]. Figure 5 demonstrates how the hyper-parameter c (a base channel size mentioned in the paper) affects the overall model size. We refer the reader to the original papers [22, 14, 55] and to our code for full details of the U-Net.

Algorithm 3 Training

Input: dataset $\mathcal{D} = \{I_s^0, I_s^n, I_s^1\}_{s=1}^S$ of consecutive frame triplets, maximum diffusion step T, noise schedule $\{\beta_t\}_{t=1}^T$
Load: pre-trained VQ-FIGAN encoder E
Initialize: denoising U-Net ϵ_θ
repeat
  Sample $(I^0, I^n, I^1) \sim \mathcal{D}$
  Encode $z^0 = E(I^0), z^n = E(I^n), z^1 = E(I^1)$
  Sample $t \sim \mathcal{U}(1, T)$
  Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $z_t^n = \sqrt{\bar{\alpha}_t} z^n + \sqrt{1-\bar{\alpha}_t}\epsilon$
  Take a gradient descent step on
$$\nabla_\theta \left\| \epsilon - \epsilon_\theta(z_t^n, t, z^0, z^1) \right\|^2$$
until converged

Algorithm 4 Inference

Input: original frames I^0, I^1, noise schedule $\{\beta_t\}_{t=1}^T$, maximum diffusion step T
Load: pre-trained denoising U-Net ϵ_θ, VQ-FIGAN encoder E and decoder D
Sample $z_T^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Encode $z^0 = E(I^0), z^1 = E(I^1)$ and store features ϕ^0, ϕ^1
for $t = T, \ldots, 1$ **do**
  Predict noise $\hat{\epsilon} = \epsilon_\theta(z_t^n, t, z^0, z^1)$
  $\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(z_t^n - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}\right), \quad \sigma_t^2 = \tilde{\beta}_t$
  Sample $\zeta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $z_{t-1}^n = \mu_\theta + \sigma_t \zeta$
end for
return $\hat{I}^n = D(z_0^n, \phi^0, \phi^1)$ as the interpolated frame

Algorithm 5 DDIM Sampling for LDMVFI

Input: original frames I^0, I^1, noise schedule $\{\beta_t\}_{t=1}^T$, maximum DDIM step $\mathcal{T}$
Load: pre-trained denoising U-Net ϵ_θ, VQ-FIGAN encoder E and decoder D
Sample $z_\mathcal{T}^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Encode $z^0 = E(I^0), z^1 = E(I^1)$ and store features ϕ^0, ϕ^1
for $t = \mathcal{T}, \ldots, 1$ **do**
  Predict noise $\hat{\epsilon} = \epsilon_\theta(z_t^n, t, z^0, z^1)$
  $\hat{z}_0^n = \frac{1}{\sqrt{\bar{\alpha}_t}}(z_t^n - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon})$
  $z_{t-1}^n = \sqrt{\bar{\alpha}_{t-1}}\hat{z}_0^n + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return $\hat{I}^n = D(z_0^n, \phi^0, \phi^1)$ as the interpolated frame

E. Full Quantitative Evaluation Results

The full evaluation results of LDMVFI and the compared VFI methods on all test sets (Middlebury [1], UCF-101 [66], DAVIS [52] and SNU-FILM [7]) in terms of all metrics (PSNR, SSIM [71], LPIPS [76], and FloLPIPS [10]) are summarized in Tables 5-6.

F. Full Ablation Study Results

In the main paper, due to space limit we presented the ablation experiment on three test sets. Here we also include the evaluation results of all 8 variants (mentioned in the main paper) on the SNU-FILM dataset. These are summarized in Tables 7.

G. Details of Subjective Experiment

The user study and the use of human data have undergone an internal ethics review and have been approved by the Institutional Review Board.

The subjective experiment was conducted in a lab-based environment. The monitor used to display videos was a BENQ XL2720Z (59.8×33.6cm screen size). The spatial resolution of the display was set to 1920 × 1080 and the frame rate was set to 60Hz. The viewing distance of the participants was 1m (approximately three times screen height [28]).

In the main paper, we state that t-tests are performed between the sequence-wise preference ratios of the proposed method and the three tested baselines. Here we report the p values for the t-tests in Table 4.

Comparison	p value
Ours vs BMBC	3.8×10^{-9}
Ours vs IFRNet	1.6×10^{-3}
Ours vs ST-MFNet	1.7×10^{-2}

Table 4. The results of t-test analysis on the user study results.

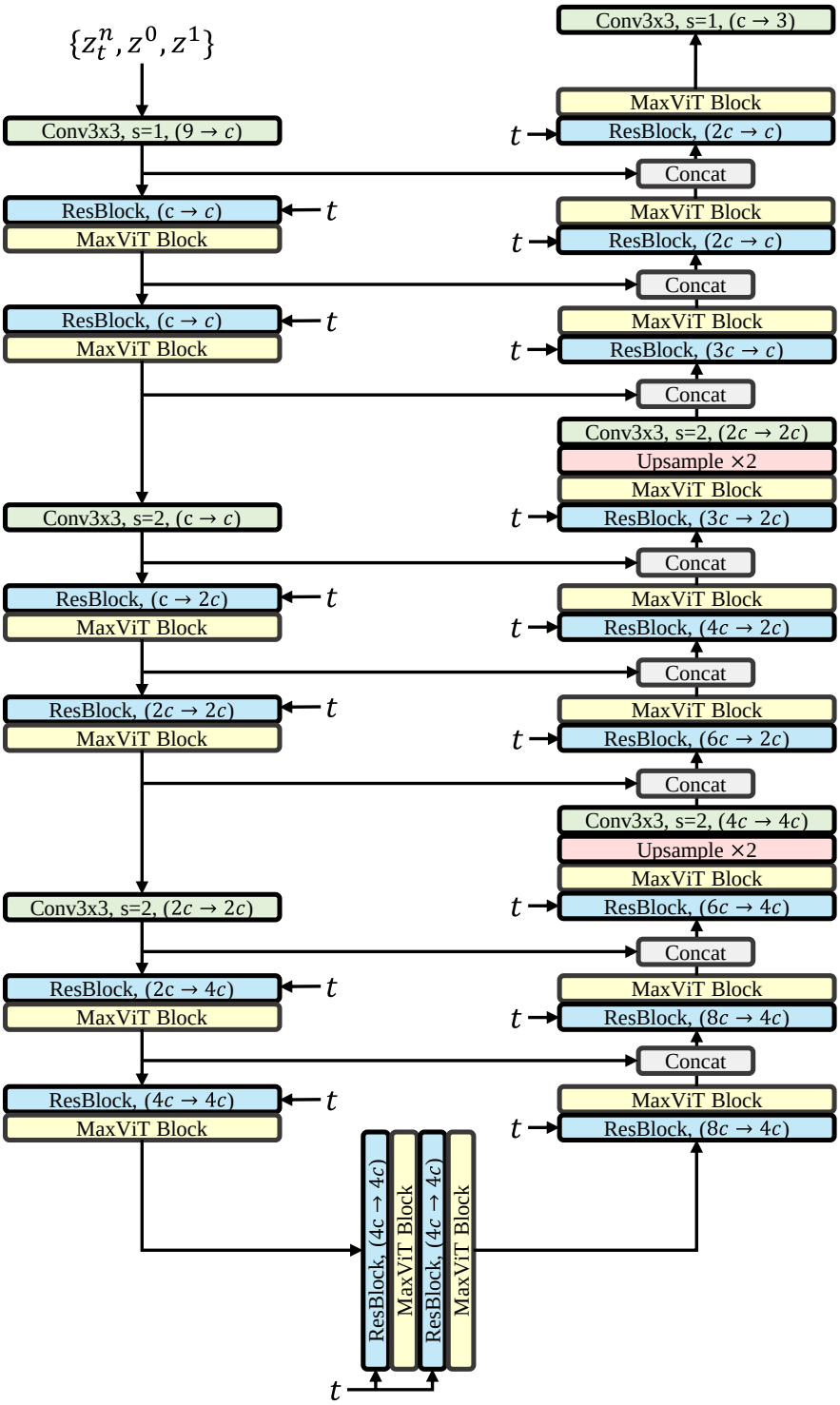


Figure 5. The architecture of the denoising U-Net. The hyper-parameter c is a base channel size. In each block, the $(\cdot \rightarrow \cdot)$ indicates the input and output channels of the block.

	Middlebury				UCF-101				DAVIS			
	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓
BMBC	36.368	0.982	0.023	0.046	32.576	0.968	0.034	0.045	26.835	0.869	0.125	0.185
AdaCoF	35.256	0.975	0.031	0.052	32.488	0.968	0.034	0.046	26.234	0.850	0.148	0.198
CDFI	36.205	0.981	0.022	0.048	32.541	0.968	0.036	0.049	26.471	0.857	0.157	0.211
XVFI	34.724	0.975	0.036	0.070	32.224	0.966	0.038	0.050	26.475	0.861	0.129	0.185
ABME	37.639	0.986	0.027	0.040	32.055	0.967	0.058	0.069	26.861	0.865	0.151	0.209
IFRNet	36.368	0.983	0.020	0.045	32.716	0.969	0.032	0.044	27.313	0.877	0.114	0.170
VFIformer	35.566	0.977	0.031	0.065	32.745	0.968	0.039	0.051	26.241	0.850	0.191	0.242
ST-MFNet	N/A	N/A	N/A	N/A	33.384	0.970	0.036	0.049	28.287	0.895	0.125	0.181
FLAVR	N/A	N/A	N/A	N/A	33.224	0.969	0.035	0.046	27.104	0.862	0.209	0.248
MCVD	20.539	0.820	0.123	0.138	18.775	0.710	0.155	0.169	18.946	0.705	0.247	0.293
LDMVFI	34.033	0.971	0.019	0.044	32.186	0.963	0.026	0.035	25.541	0.833	0.107	0.153

Table 5. Quantitative comparison results of LDMVFI ($f = 32$) and 10 tested methods. Note ST-MFNet and FLAVR require four input frames so cannot be evaluated on Middlebury that contains frame triplets. In each column we color the best result and the second best.

	SNU-FILM-Easy				SNU-FILM-Medium			
	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓
BMBC	39.809	0.990	0.020	0.031	35.437	0.978	0.034	0.059
AdaCoF	39.632	0.990	0.021	0.033	34.919	0.975	0.039	0.066
CDFI	39.881	0.990	0.020	0.031	35.224	0.977	0.036	0.066
XVFI	38.903	0.989	0.022	0.037	34.552	0.975	0.039	0.072
ABME	39.697	0.990	0.022	0.034	35.280	0.977	0.042	0.076
IFRNet	39.881	0.990	0.019	0.030	35.668	0.979	0.033	0.058
ST-MFNet	40.775	0.992	0.019	0.031	37.111	0.985	0.036	0.061
FLAVR	40.161	0.990	0.022	0.034	36.020	0.979	0.049	0.077
MCVD	22.201	0.828	0.199	0.230	21.488	0.812	0.213	0.243
LDMVFI	38.674	0.987	0.014	0.024	33.996	0.970	0.028	0.053

	SNU-FILM-Hard				SNU-FILM-Extreme			
	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓
BMBC	29.942	0.933	0.068	0.118	24.715	0.856	0.145	0.237
AdaCoF	29.477	0.925	0.080	0.131	24.650	0.851	0.152	0.234
CDFI	29.660	0.929	0.081	0.141	24.645	0.854	0.163	0.255
XVFI	29.364	0.928	0.075	0.138	24.545	0.853	0.142	0.233
ABME	29.643	0.929	0.092	0.168	24.541	0.853	0.182	0.300
IFRNet	30.143	0.935	0.065	0.122	24.954	0.859	0.136	0.229
ST-MFNet	31.698	0.951	0.073	0.123	25.810	0.874	0.148	0.238
FLAVR	30.577	0.938	0.112	0.169	25.206	0.861	0.217	0.303
MCVD	20.314	0.766	0.250	0.292	18.464	0.694	0.320	0.385
LDMVFI	28.547	0.917	0.060	0.114	23.934	0.837	0.123	0.204

Table 6. Quantitative comparison results on SNU-FILM (note VFIformer is not included because the GPU goes OOM).

	SNU-FILM-Easy		SNU-FILM-Medium		SNU-FILM-Hard		SNU-FILM-Extreme	
	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓	LPIPS↓	FloLPIPS↓
V1	0.054	0.056	0.068	0.082	0.104	0.142	0.172	0.243
V2	0.020	0.028	0.035	0.058	0.076	0.126	0.150	0.233
V3	0.016	0.026	0.030	0.052	0.068	0.119	0.144	0.227
V4	0.014	0.024	0.027	0.052	0.059	0.112	0.122	0.204
V5	0.019	0.029	0.034	0.059	0.072	0.123	0.149	0.234
V6	0.016	0.026	0.032	0.060	0.076	0.132	0.157	0.243
V7	0.016	0.024	0.031	0.054	0.073	0.127	0.155	0.241
V8	0.015	0.024	0.030	0.052	0.063	0.115	0.125	0.216
Ours	0.014	0.024	0.028	0.053	0.060	0.114	0.123	0.204

Table 7. Ablation experiment results on SNU-FILM. Full details of the variants (V1-8) can be found in the main paper.

H. Limitations and Societal Impact

Limitations. Firstly, as discussed in the main paper, the proposed LDMVFI shows a much slower inference speed compared to the other state-of-the-art methods. This is a common drawback of diffusion models [22, 55] mainly due to the iterative reverse process during generation. Various techniques [57, 32] have been proposed to speed up the sampling process and these can also be applied to LDMVFI. Secondly, the number of model parameters in LDMVFI is also larger than other methods. The main component of LDMVFI that accounts for the large model size is the denoising U-Net, which was developed in previous work [22, 14, 55] and modified in here by replacing the self-attention layers with MaxViT blocks [67]. To reduce the model size, techniques such as knowledge distillation [21] and model compression [4] can be used. The exploration of these techniques, as well as the accelerated diffusion sampling methods in the context of VFI, remain for future work.

Potential Negative Societal Impact. Firstly, in general, generative models for images and videos can be used to generate inappropriately manipulated content or in unethical ways (e.g. "deep fake" generation) [13]. Secondly, the two-stage training strategy, large model size and slow inference speed of LDMVFI mean that large-scale training and evaluation processes can consume significant amounts of energy, leading to increased carbon footprint [36]. We refer the readers to [13] and [36] for more detailed discussions on these matters.

I. Attribution of Assets: Code and Data

In this section, we summarize the sources and licenses of all the datasets and code used in the work. The attribution of datasets and code are shown in Tables 8 and 9 respectively. For MCVD [70], we used the `smmnist_DDPM_big5.yml` configuration[1], setting both the numbers of previous and future frames to 1, and the number of interpolated frames also to 1.

J. Additional Qualitative Examples

We include more visual examples of frames interpolated by LDMVFI and other competing methods in Figures 6 and 7.

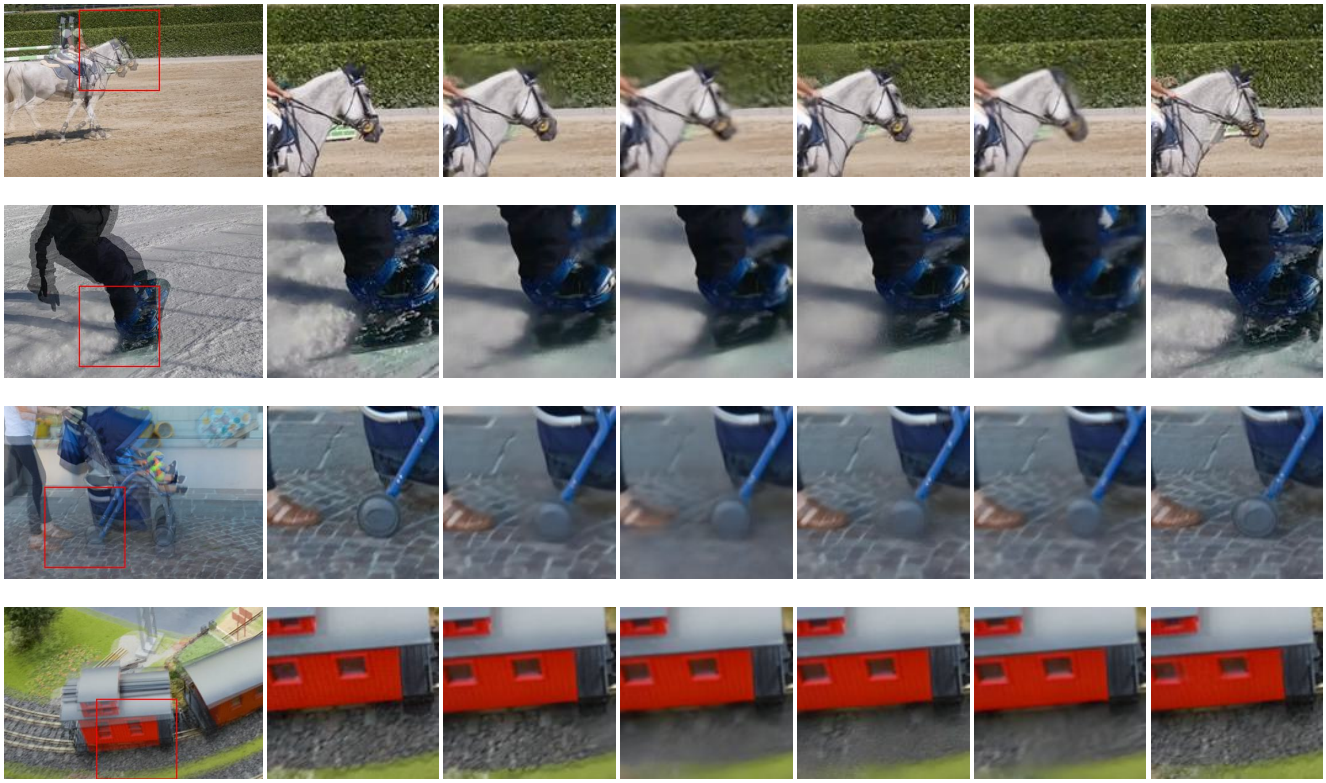
Dataset	Dataset URL	License / Terms of Use
Vimeo-90k [74]	http://toflow.csail.mit.edu	MIT license.
BVI-DVC [42]	https://fan-aaron-zhang.github.io/BVI-DVC/ (original videos); https://github.com/danielism97/ST-MFNet (quintuplets)	All sequences are allowed for academic research.
UCF101 [66]	https://www.crcv.ucf.edu/research/data-sets/ucf101/	No explicit license terms, but compiled and made available for research use by the University of Central Florida.
DAVIS [52]	https://davischallenge.org	BSD license.
SNU-FILM [7]	https://myungsub.github.io/CAIN/	MIT license .
Middlebury [1]	https://vision.middlebury.edu/flow/data/	All sequences are available for research use.
BVI-HFR [43]	https://fan-aaron-zhang.github.io/BVI-HFR/	Non-Commercial Government Licence for public sector information.

Table 8. License information for the datasets used in this work.

[1] https://github.com/voletiv/mcvd-pytorch/blob/master/configs/smmnist_DDPM_big5.yml

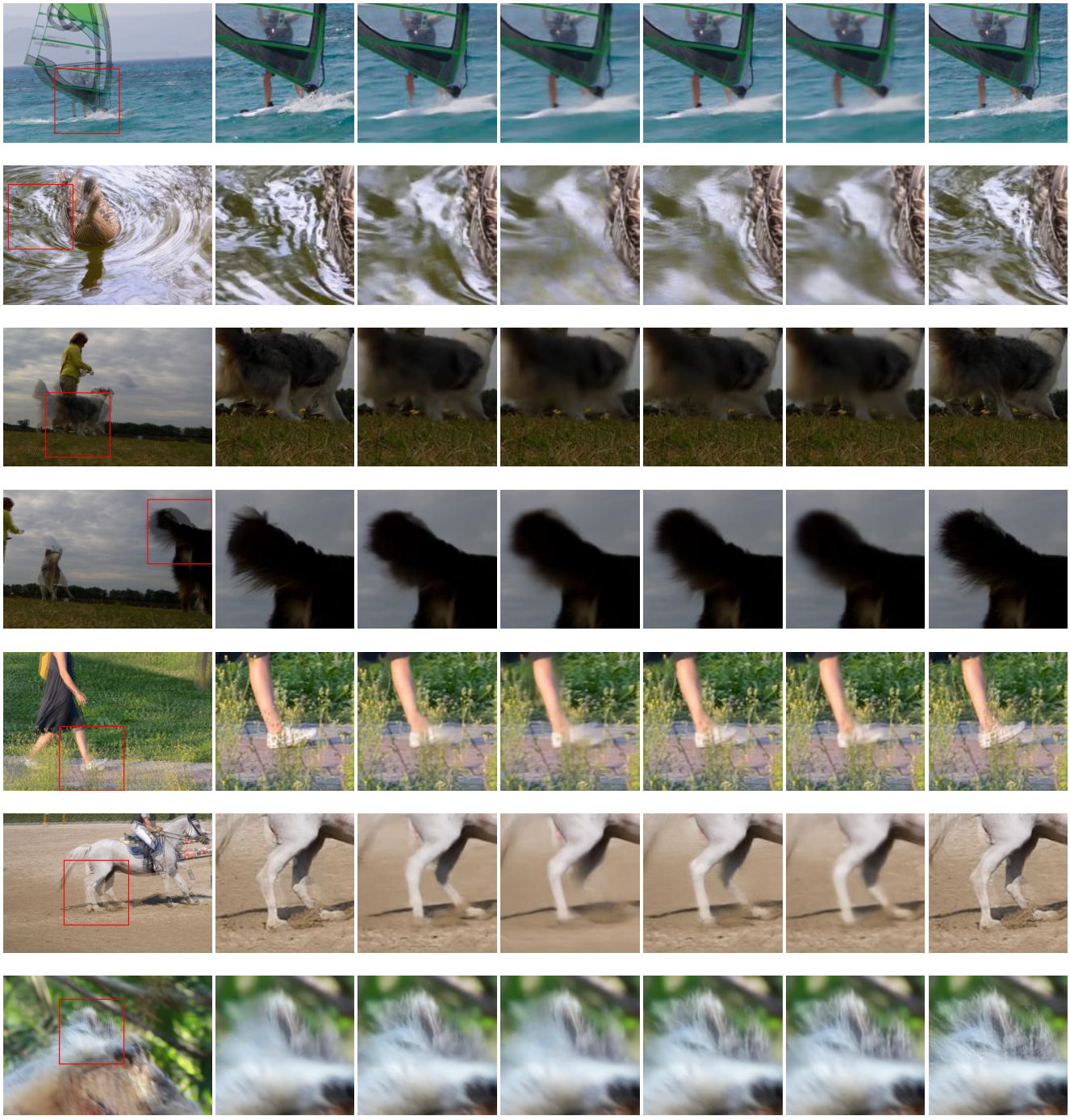
Method	Source code URL	License / Teams of Use
BMBC [50]	https://github.com/JunHeum/BMBC	MIT license.
AdaCoF [37]	https://github.com/HyeongminLEE/AdaCoF-pytorch	MIT license.
CDFI [15]	https://github.com/tding1/CDFI	Available for research use.
XVFI [60]	https://github.com/JihyongOh/XVFI	Research and education only.
ABME [51]	https://github.com/JunHeum/ABME	MIT license.
IFRNet [35]	https://github.com/ltkong218/IFRNet	MIT license.
VFIformer [41]	https://github.com/dvlab-research/VFIformer	Available for research use.
ST-MFNet [11]	https://github.com/danielism97/ST-MFNet	MIT license.
FLAVR [30]	https://github.com/tarun005/FLAVR	Apache-2.0 license.
MCVD [70]	https://github.com/voletiv/mcvd-pytorch	MIT license.

Table 9. License information for the code assets used in this work.



(a) Overlaid inputs (b) GT (c) BMBC (d) VFIformer (e) IFRNet (f) ST-MFNet (g) LDMVFI (ours)

Figure 6. More visual interpolation examples.



(a) Overlaid inputs (b) GT (c) BMBC (d) VFIformer (e) IFRNet (f) ST-MFNet (g) LDMVFI (ours)

Figure 7. Additional visual interpolation examples.