REALTIME VIDEO FRAME INTERPOLATION USING ONE-STEP VIDEO DIFFUSION SAMPLING

Anonymous authors
Paper under double-blind review

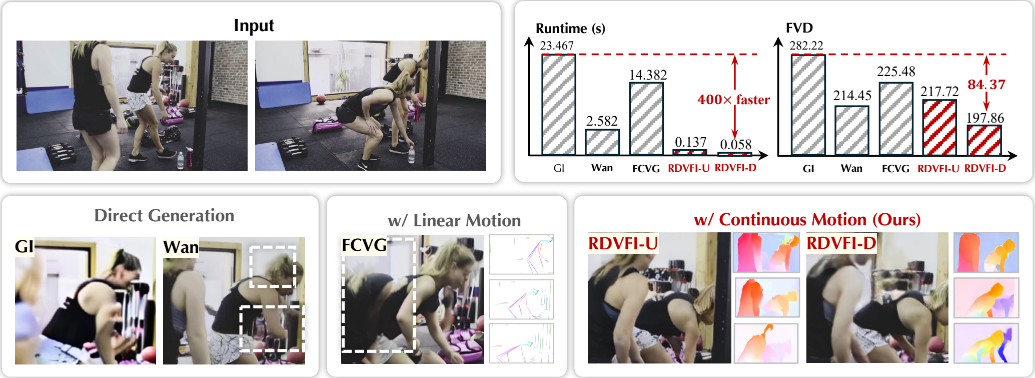

Figure 1: Compared with existing video diffusion-based interpolation methods, our method interpolates at extreme high efficiency, achieving 400× acceleration than the current multi-step solution with also better results.

ABSTRACT

Recent research on video Frame Interpolation (VFI) shows that a pretrained Video Diffusion Model (VDM) can solve many challenging scenarios, including large or complex motion. However, VDMs require tedious diffusion sampling, making the inference slow. One possible way to accelerate is to distill a multi-step model into a one-step model, but additional modules are often introduced during distillation, which significantly increase training overhead. Instead, we propose a Real-time Diffusion-based Video Frame Interpolation pipeline, RDVFI. RDVFI achieves efficient interpolation by disentangling this task into two subproblems: motion and appearance generation. Specifically, RDVFI first calculates pixel movements across frames with the continuous motion fields, only utilizing a few sparse key frames. As a result, RDVFI only forwards the diffusion model for these sparse key frames rather than for each intermediate frame, effectively reducing one-step training cost. In the second appearance estimation step, RDVFI then only needs to create intermediate frames by warping input frames with sampled optical flows from the estimated continuous motion field in the first step. Because our diffusion model creates motions only, it can work at a fixed and relatively small resolution, leading to superior training and inference efficiency. Extensive experiments show that our RDVFI generates comparable or superior interpolation quality compared with existing multi-step solutions. It also offers outstanding inference efficiency, interpolating 17FPS at 1024×576 resolution, achieving **50× acceleration** than the fastest diffusion-based generation by Wan et al. (2025).

1 INTRODUCTION

Video frame interpolation (VFI) aims to generate intermediate frames between low-frame-rate inputs, which has wide applications in super slow-motion video generation (Niklaus et al., 2017; Jiang et al., 2018; Liu et al., 2019; Hu et al., 2022b), virtual reality (Anderson et al., 2016; Yang et al., 2023), and video compression (Wu et al., 2018; Xu et al., 2024). The core of VFI is determining the pixel movements across frames, which is ambiguous and challenging because pixels can move from the first image to the second along different paths. Traditional non-generative methods usually solve this challenge with pre-determined motion models, like the linear model by Jiang et al. (2018); Bao et al. (2019), polynomial by Xu et al. (2019); Liu et al. (2020)), or a learned motion model by Huang et al.; Reda et al.; Li et al. (2023). Still, these methods can only handle simple motion and may fail on complex and large real-world motion. Danier et al. and Lew et al. (2025) try to improve the robustness of motion estimation using a pretrained image generation model, but they still mainly focus on small and simple motion.

Recently, with the rapid development of video generation models, researchers (Wang et al., 2024; Zhu et al., 2024; Wan et al., 2025) remodeled video interpolation as a conditional generation task through recursive diffusion sampling with Video Diffusion Models (VDM). While these solutions show great potential in handling complex motion, they still face several critical challenges: (1) Multi-step sampling in the VDM restricts its efficiency. Current VDM-based interpolation inherits the sampling strategy of a pretrained VDM, which is often slow, taking more than 20 steps. (2) Computational overhead and performance degradation due to the additional modules. The majority of VDM-based interpolation networks are trained by finetuning and fusing pretrained image-to-video diffusion models (VDMs) (Wang et al., 2024; Zhu et al., 2024), optionally with auxiliary modules, *e.g.*, ControlNet, further slowing down inference (Zhu et al., 2024). (3) Interpolation instability. Fusing two image-to-video generation models (Wang et al., 2024; Zhang et al., 2024) conditioned on only one input image usually introduces mismatched results and unstable interpolation.

Although one-step distillation can significantly accelerate diffusion and has been widely used in image (Liu et al., 2022; Yin et al., 2024) or video (Zhang et al., 2024) generation, there are still challenges when applying it to interpolation. First, existing one-step training may introduce additional training overhead, making high-resolution training challenging. Normally, previous methods either jointly optimize two diffusion models (Yin et al., 2024) or introduce adversarial networks (Zhang et al., 2024; Mao et al., 2025). These additional modules improve visual quality with more resource consumption. As a result, the majority of models either only support LoRA training (Mao et al., 2025), or can only be fully finetuned on low-resolution frames (Zhang et al., 2024) (trained on 768x448, 14 frames). Second, some one-step training using adversarial loss may hurt the fidelity of video interpolation, introducing non-existing high-frequency details or unnatural motion of objects.

To address these issues, we propose RDVFI, a real-time one-step diffusion-based interpolation model that can deal with large, complex motions. Unlike previous diffusion-based interpolation that directly generates output frames, our RDVFI decouples the problem into motion prediction and appearance estimation. Specifically, RDVFI first constructs a continuous motion field, recording the movement of all pixels between all interpolated frames. As previous work shows (Tulyakov et al., 2022), even a low-resolution and sparse motion can represent complex motion, so we train a one-step diffusion model to estimate a sparse motion field on a few key frames (e.g. 7 key frames). To generate dense and high-resolution interpolation results (e.g., interpolate 24 frames between every 2 frames), we can sample their motion from the sparse motion field and upsample it to the full resolution. This design greatly reduces the computational cost of the diffusion-based motion estimator. In the next step, with the estimated motion, RDVFI further generates each frame by warping pixels from two input frames and fusing them using another small synthesis network. With this design, the computationally expensive diffusion model only runs on fixed and low-resolution sparse input frames. Thus, our RDVFI can flexibly interpolate at dynamic spatial and temporal resolutions with superior computation efficiency.

Furthermore, to effectively train RDVFI, we also introduce the latent-pixel training strategy. Instead of end-to-end training, we break the training into two stages. The first stage only trains a motion-guided decoding, which takes the VAE latents of all intermediate frames and outputs the continuous motion field. Note that in actual inference, the network only takes the first and the last frame as input to the motion estimator. Therefore, in the second stage, we train a one-step diffusion model that estimates the latent features of all intermediate frames, which will be used as input to the motion decoder in the first stage. This training strategy effectively decouples the appearance model training from the motion module and reduces the training overhead at each stage. As a result, our model can efficiently train on high-resolution videos (up to 1280x720 25-frame), resulting in high-quality interpolation.

With all these designs, our RDVFI outperforms state-of-the-art multi-step VDM-based VFI methods, as shown in Fig. 1. Benefit from less diffusion samplings at lower temporal and spatial resolution, our RDVFI interpolates at real-time (17FPS at 1024×576 resolution) [1], accelerating by 50× compared with the latest baseline (Wan et al., 2025). In summary, our contributions include:

- We propose a novel one-step diffusion model for real-world VFI, RDVFI. To our knowledge, it is the first diffusion-based VFI method with one-step inference.
- We propose an efficient and effective VFI pipeline by disentangling motion and appearance generations, which flexibly interpolate at dynamic spatial and temporal resolutions.
- We design the latent-pixel training strategy for efficient one-step VFI diffusion training.
- Extensive experimental results demonstrate that the proposed RDVFI outperforms state-of-the-art methods across various benchmarks with remarkable efficiency gain.

2 Related Work

2.1 Video Frame Interpolation

Conventional video frame interpolation methods can be distinguished by how they determine intermediate motions. Creating motions across consecutive timesteps across input frames is an ill-posed issue. Early methods solve this challenge by assuming pixels move along a trajectory defined by a specific mathematical model, *e.g.,* linear (Jiang et al., 2018; Bao et al., 2019), quadratic (Xu et al., 2019; Liu et al., 2020); however, these methods cannot deal with diverse real-world motions, especially complex ones. Some methods (Huang et al.; Reda et al.; Li et al., 2023) adopt a data-driven manner, forcing the network to predict flows between specific intermediate and input frames. These methods reduce warping artifacts, but do not help solve large, complex motions, as they cannot model nonlinear and non-rigid movements.

2.2 Diffusion Models in VFI

Recently, researchers have found that diffusion models, particularly Latent Diffusion Models (LDM), advance VFI quality. These diffusion-based methods have two main categories: image diffusion-based (Danier et al.; Lew et al., 2025) and video diffusion-based (Feng et al., 2024; Yang et al., 2024; Wang et al., 2024; Zhu et al., 2024). Image diffusion-based methods improve optical flow accuracy between one specific intermediate and input frames with diffusion models. However, it is challenging to estimate optical flows for complex motions due to their nonlinear, nonrigid nature and ambiguities. As a result, these methods still focus on small, simple motions. Video diffusion-based methods simultaneously generate all intermediate frames through recursive sampling, introducing substantial cost in time and computational resources. These methods fuse two fine-tuned image-to-video SVD (Blattmann et al., 2023) to produce forward and backward frames, conditioned on only the first or second input frame. However, their bi-directional predictions are usually mismatched when input frames contain complex motions, as they only sample the diffusion with one input frame; Zhu et al. (2024) tries to solve this problem by involving an additional matching module to provide linear motion control signals, but it is hard to fit diverse real-world motions.

2.3 Diffusion Acceleration

Reducing diffusion sampling steps is a straightforward and effective way to boost diffusion inference, involving methods like rectified flow (Liu et al., 2022; 2023), adversarial training (Lin et al., 2025; Zhang et al., 2024), and score distillation (Wang et al., 2023; Yin et al., 2024). One-step acceleration is an extreme setting that receives significant attention and has been integrated into image and video generation algorithms. For instance, Liu et al. (2023) accelerates text-to-image generation with rectified flow; Yin et al. (2024) mitigates target and generated distribution gaps for superior image generation quality; Lin et al. (2025) and Zhang et al. (2024) introduce adversarial loss for better video generation quality. However, these techniques cannot work for VFI tasks because of the unaffordable training overhead. In addition, intermediate frames that are away from

[1] All runtimes in this work are calculated on a single A100 at 1024x576 resolution for $\times 24$ interpolation.

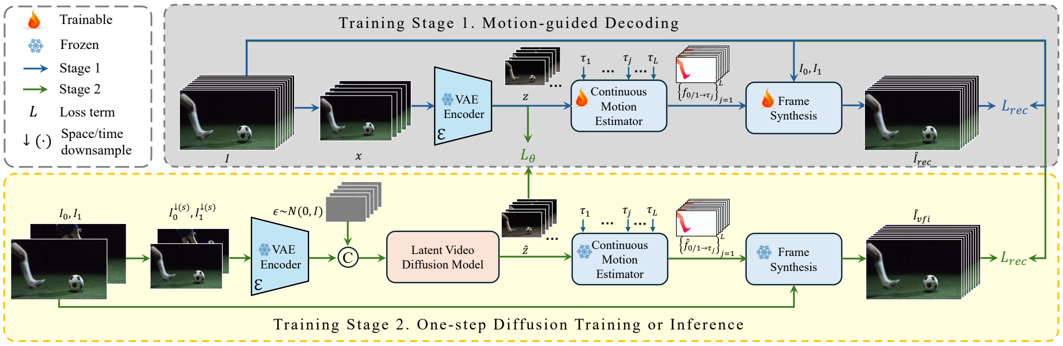
Trainable
Frozen
Stage 1
Stage 2
L Loss term
$\downarrow(\cdot)$ Space/time downsample
Training Stage 1. Motion-guided Decoding
I
x
VAE Encoder
$\mathcal{E}$
z
L_θ
τ_1 ... τ_j ... τ_L
Continuous Motion Estimator
$\{f_{0/1\to\tau_j}\}_{j=1}^{L}$
I_0, I_1
Frame Synthesis
$\hat{I}_{rec}$
L_{rec}
I_0, I_1
$I_0^{\downarrow(s)}, I_1^{\downarrow(s)}$
$\epsilon \sim N(0, I)$
VAE Encoder
$\mathcal{E}$
C
Latent Video Diffusion Model
$\hat{z}$
τ_1 ... τ_j ... τ_L
Continuous Motion Estimator
$\{\hat{f}_{0/1\to\tau_j}\}_{j=1}^{L}$
Frame Synthesis
$\hat{I}_{vfi}$
L_{rec}
Training Stage 2. One-step Diffusion Training or Inference

Figure 2: Overall framework of the proposed RDVFI. Our RDVFI interpolates intermediate frames $\hat{I}_{vfi}$ using one-step diffusion sampling with input two ending frames I_0, I_1. For efficient and effective training, we propose a two-stage training strategy. **Stage 1** trains a motion-guided decoding that decodes VAE latent into continuous motion field, which is used to synthesize high-resolution frames. **Stage 2** trains a one-step diffusion model to generate latent features of intermediate frames from two input frames. Its output will be used as input to the motion estimator trained in stage 1.

the input usually have higher ambiguities. Simply adapting existing one-step text-to-image or text-to-video acceleration techniques may hinder interpolation consistency. Thus, this work proposes a simple and effective one-step diffusion sampling method that accelerates VFI diffusion inference.

3 Method

In this section, we first introduce the overall framework (Sec. 3.1), and then discuss each modules, including the continuous motion field (Sec. 3.2), one-step Video Diffusion Model (Sec. 3.3), and frame synthesis network (Sec. 3.4). Finally, we describe the training process (Sec. 3.5).

3.1 Overall Framework

Fig. 2 shows our overall structure. Given two input frames $\{I_0, I_1\} \in R^{H\times W}$, we first downsample them into a fixed low resolutions $\{I_0^{\downarrow(s)}, I_0^{\downarrow(s)}\}$ and encode them with a pretrained diffusion VAE encoder $\mathcal{E}(\cdot)$ and obtain low-resolution latent features $\{z_0, z_1\} \in R^{h\times w}$, where H, W, h, w are image height, image width, latent height, latent width, respectively. Instead of directly estimating the intermediate frames conditioned on encoded input latent features with the diffusion model, our RDVFI adopts a two-step estimation, including motion and appearance estimation.

In the first step of motion estimation, we first estimate the latent features z_k of key frames at a fixed low resolution $(l \times h \times w)$ by a diffusion model, even if the input frames are very high-resolution. Then, we estimate the continuous motion field with estimated latent features. This is much more efficient than previous diffusion-based interpolations that directly run diffusion model on high resolution.

In the second step for appearance estimation, for each frame to be interpolated, RDVFI samples a movement of between this frame and two input frames $\{f_{0\to\tau_j}, f_{1\to\tau_j}\}_{j=1}^{L} \in R^{L\times h\times w}$ from the continuous motion field, and synthesizes intermediate frames $\{I_{\tau_j}\}_{j=1}^{L}$ based on the warped two input frames, as shown in Fig. 3. Details of each step are introduced below.

3.2 Continuous Motion Field

Motivation Current VFI methods usually model pixel movements across input frames in non-parametric or parametric fashions. Non-parametric methods (Huang et al.; Reda et al.; Li et al., 2023; Danier et al.; Lew et al., 2025) directly create optical flows in a data-driven manner for each interpolation. They thus cannot deal with complex motions because estimating optical flows for complex motions is challenging due to their nonlinear, non-rigid nature, even with known ground truth frames. Parametric ones (Jiang et al., 2018; Bao et al., 2019; Xu et al., 2019; Liu et al., 2020;

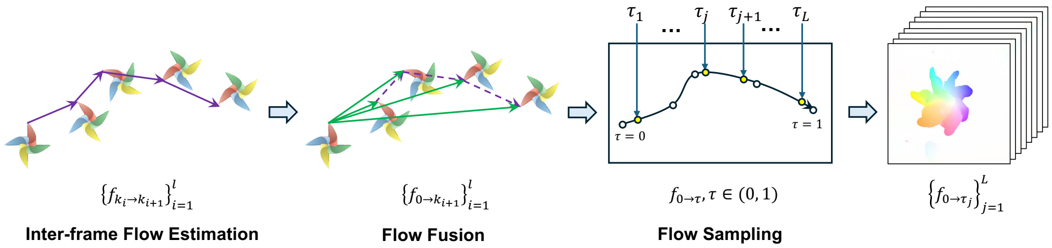
τ_1
τ_j
τ_{j+1}
τ_L
...
$\tau = 0$
$\tau = 1$
$\{f_{k_i \to k_{i+1}}\}_{i=1}^{l}$
$\{f_{0 \to k_{i+1}}\}_{i=1}^{l}$
$f_{0 \to \tau}, \tau \in (0, 1)$
$\{f_{0 \to \tau_j}\}_{j=1}^{L}$
Inter-frame Flow Estimation
Flow Fusion
Flow Sampling

Figure 3: Continuous motion field estimation pipeline. Given key frames' latent features $\{z_{k_i}\}_{i=1}^{l}$, RDVFI first estimates inter-frame flows (purple arrows) with a neural network ϕ_1^f following Eq. 1 and then iteratively fuses these estimated flows to get pixel offsets from input to each key frame (green arrows) following Eq. 2. Because we update a slight offset to the previous estimation $f_{0\to k_i}$ based on $f_{k_i \to k_{i+1}}$, the network ϕ_2^f can estimate $f_{0\to k_{i+1}}$ efficiently and effectively. We further fit pixel movement splines $f_{0\to\tau}$ with the estimated pixel offsets between key and input frames (white points), which enables us to sample optical flows for interpolation at any intermediate time step $\tau \in (0, 1)$ (yellow points). In a reversed order, we estimate the backward interpolation optical flows in the same way and networks.

Tulyakov et al., 2022) approximate pixel movements with a specific mathematical function, such as linear and quadratic. These functions are simple due to limited inputs and cannot fit real-world, fast-changing pixel movements. Our continuous motion field advances these parametric methods by involving key frames generated by the one-step diffusion sampling. These key frames break down the time interval between input frames, enabling our continuous motion field to model pixel movements across frames precisely. Please find Sec. A for more explanations.

Pipeline Our continuous motion field aims to solve the above limitations of existing pixel movements modeling techniques to advance VFI in more practical applications. Instead of using a predefined mathematical function to approximate the full pixel movements across input frames, we use the motion of several key frames to define a smooth spline trajectory between two input frames. The motion of key frames are estimated from the movement of latent VAE features. Based on that, RDVFI then estimates forward and backward pixel moving trajectories:

$$f_{k_i \to k_{i+1}}, f_{k_{i+1} \to k_i} = \phi_1^f(z_{k_i}, z_{k_{i+1}}), \tag{1}$$

where ϕ_1^f is a neural network, $0 < k_1 < ... < k_l < 1$ are time steps of key frames and $f_{k_i \to k_{i+1}}$ is optical flow from time k_i to k_{i+1}.

Further, we adopt a novel iterative motion estimation to improve accuracy. unlike existing methods that directly estimate flows between input and each intermediate frames as (Lew et al., 2025; Danier et al.) from scratch, our RDVFI iteratively fuses these pixels moving trajectories for optical flows between input and each intermediate frames with a neural network ϕ_2^f:

$$f_{0 \to k_{i+1}} = \phi_2^f(f_{0 \to k_i}, f_{k_i \to k_{i+1}}, z_0, z_{k_{i+1}}). \tag{2}$$

Eq. 2 shows an example of how we iterate optical flows from time 0 to 1, and we simultaneously estimate these flows from time 1 to 0 in a reversed order. We utilize the same network ϕ_2 and weights for all estimation. Because $\phi_2^f(\cdot)$ updates a small pixel offset to the previous estimation $f_{0\to k_i}$ and inter-frame motion $f_{k_i \to k_{i+1}}$, decomposing large complex motions into small and simple components, our RDVFI can solve more challenging sequence motions compared with existing flow-based diffusion methods (Lew et al., 2025; Danier et al.). The $f_{0\to k_{i+1}}$ is a temporally discrete and spatially dense pixel offset function. Inspired by (Tulyakov et al., 2022), we further fit pixel movements splines with estimated $f_{0\to k_{i+1}}$ for densification, which can generate flows $f_{0\to\tau}$, where $\tau \in (0, 1)$ can be any intermediate time steps. We define this as the continuous motion field.

3.3 One-step Video Diffusion Sampling

The continuous motion estimation introduced above still relies on the low-resolution latent features of intermediate key frames. However, during the actual inference, there are only two input frames and no intermediate latent features. Therefore, we use latent video diffusion model (LVDM) to generate them from two input frames.

More specifically, for efficiency, RDVFI downsamples input video sequences $I = \{I_0, I_{\tau_1}, ..., I_{\tau_L}, I_1\}$ along spatial and temporal dimensions, producing samples at a fixed resolution and length $x = \{I_{k_1}^{\downarrow(s)}, I_{k_2}^{\downarrow(s)}, ..., I_{k_l}^{\downarrow(s)}\}$. This downsampling step greatly reduces the computational cost on heavy diffusion model. In the forward process, RDVFI first encodes these samples to latent features $z = \mathcal{E}(x)$. Then, the corrupted latent z^t is obtained by adding Gaussian noise $z^t = \alpha^t z + \sigma^t \epsilon$, where $t \in [0, T]$ is a corruption step and z^T matches pure noise, α^t and σ^t define a fixed noise schedule, ϵ is Gaussian Noise. We reverse such corruption with a denoising network f_θ, conditioning on encoded input frames z_0, z_1, where the training objective is:

$$\mathcal{L}_\theta(t) = ||f_\theta(z^t; t, z_0, z_1) - v^t||, \tag{3}$$

$v^t = \alpha^t \epsilon - \sigma^t z$ is referred as v-prediction (Salimans & Ho). After convergence, we iteratively reverse the noising process and obtain the denoised latent features $\hat{z}$. Unlike existing methods (Wang et al., 2024; Zhu et al., 2024), decoding the denoised latent features $\hat{z}$ with the VAE decoder $\mathcal{D}$ that pairs with the encoder $\mathcal{E}$ for videos, our RDVFI decoding latent features with the estimated continuous motion field in Sec. 3.2 using frame synthesis network in Sec. 3.4.

3.4 Frame Synthesis Network

Given the motion field estimated from intermediate latent features $\hat{z}$ by the LVDM, the last step is to synthesize full-resolution dense output frames. We use a frame synthesis network to calculate intermediate frames at any time $\{\tau_j\}_{j=1}^L \in (0, 1)$, based on forward and backward flows $(\{f_{0\rightarrow\tau} and f_{1\rightarrow\tau}\})$. As shown in Algo. 1 and Fig. 2, we first sample optical flows at each interpolation timestep $\tau = \tau_j$ from the estimated continuous motion field to generate flows $\{f_{0\rightarrow\tau_j}, f_{1\rightarrow\tau_j}\}$ to warp frames. To further improve the interpolation by involving frame variations that flows cannot model, we warp the denoised latent features by sampling inter-frame motion from the continuous motion field. To avoid diffusion model biases towards appearance rather than motion generation, we detach the gradient of denoised latent here. We then utilize a multi-scale neural network ϕ^s to correct and fuse the final interpolation results $\{\hat{I}_{\tau_j}\}_{j=1}^L$. Please find Sec. B.3 for more details.

Algorithm 1: RDVFI Frame synthesis.

```
for j=1, 2, ..., L do
    Sample optical flows $\{f_{0\rightarrow\tau_j}, f_{1\rightarrow\tau_j}\}$ from $\{f_{0\rightarrow\tau}, f_{1\rightarrow\tau}\}$
    $I_{0\rightarrow\tau_j} = Warp(I_0, f_{0\rightarrow\tau_j}), I_{1\rightarrow\tau_j} = Warp(I_1, f_{1\rightarrow\tau_j})$
    Find the nearest key frame timestep, satisfying $k_i < \tau_j < k_{i+1}, k_0 = 0, k_{l+1} = 1$
    Sample optical flows $\{f_{\tau_i\rightarrow\tau_j}, f_{\tau_{i+1}\rightarrow\tau_j}\}$ from $\{f_{0\rightarrow\tau}, f_{1\rightarrow\tau}\}$
    $z_{k_i\rightarrow\tau_j} = Warp(z_{k_i}, f_{k_i\rightarrow\tau_j}), z_{k_{i+1}\rightarrow\tau_j} = Warp(z_{k_{i+1}}, f_{k_{i+1}\rightarrow\tau_j})$
    $\hat{I}_j = \phi^s(I_0, I_1, I_{0\rightarrow\tau_j}, I_{1\rightarrow\tau_j}, f_{0\rightarrow\tau_j}, f_{1\rightarrow\tau_j}, z_{k_i\rightarrow\tau_j}, z_{k_{i+1}\rightarrow\tau_j})$
end for
```

3.5 Training Strategies

A simple end-to-end training of our network may result in poor training convergence. Instead, we proposed a two-stage, motion-guided decoding stage and one-step diffusion stage, as shown in Fig. 2. In the motion-guided decoding stage, we aim to train a robust motion-guided decoding pipeline to replace the original diffusion VAE decoder $\mathcal{D}$, which disentangles motion and appearance generation in VFI. More specifically, we enforce the motion-guided decoding pipeline to reconstruct high-frame-rate input video clip I by interpolating with ground truth latent features z and input frames I_0, I_1. The training objective is defined by combining LPIPS (Zhang et al., 2018) L_{lpips} and L2-norm:

$$L_{rec} = w_1 L_{lpips}(I, \hat{I}_{rec}) + w_2 ||I - \hat{I}_{rec}||_2, \tag{4}$$

where the $\hat{I}_{rec}$ is the reconstructed frames, w_1, w_2 are loss coefficients and are set to be 0.5 and 1, respectively. To this target, we remove the latent warping for $\{z_{k_i\rightarrow\tau_j}, z_{k_{i+1}\rightarrow\tau_j}\}$ in the first $3/4$ training iterations to force the network to interpolate only by flows and warping. We then freeze all parameters except those related to latent warping for superior interpolation quality in the last $1/4$

training iterations. We do not generate ground truth optical flows to train our continuous motion field. Estimating optical flows for distant frames is challenging due to their non-linear, non-rigid nature and ambiguities. These inaccurate "ground truths" may hinder our network training and lead to performance degradation.

In the one-step diffusion training stage, we freeze all parameters of the motion-guided decoding pipeline after it converges and only update the denoiser network. The denoiser network aims to fully remove added noise within one-step denoising. The training objective is as follows:

$$L = \lambda_1 L_\theta(T) + \lambda_2 L_{rec}(I, \hat{I}_{vfi}). \tag{5}$$

Here, the L_θ is defined in Eq. 3, which works for latent domain adaptation; λ_1 and λ_2 are loss weights; $\hat{I}_{vfi}$ are interpolation results. The λ_1 is a piecewise coefficient, which returns one for the first $2/5$ iterations and annealing with a factor of 0.996 for each 100 iterations. λ_2 is a step coefficient, zero for the first $2/5$ iterations and one afterward. By doing so, we skip pixel-domain decoding in the first $2/5$ iterations, quickly adapting models to the VFI task.

4 Experiments

4.1 Implementation Details

Our RDVFI has two versions: RDVFI-U, which builds upon 1.5B SVD (Blattmann et al., 2023) and RDVFI-D based on 1.3B (Wan et al., 2025). We have seven and eight key frames in RDVFI-U and method-D, respectively. During training, we randomly resize images to one resolution of $(448 \times 256, 576 \times 320, 1024 \times 576, 1280 \times 720)$. Diffusion models and the continuous motion estimator then work at the 448×256 resolution by resizing input images, and the motion-guided decoding module generates intermediate frames at input resolutions with up-scaled optical flows. We utilize a fixed 2e-5 learning rate for all experiments. We adopt eight A100-80G GPUs and a total batch size of 8. We train 200K iterations for the first training stage and 500K for the second. Please refer to Sec. B for network implementation details.

4.2 Datasets and Evaluation Metrics

Following (Zhu et al., 2024), we form our training dataset by filtering video clips from the DAVIS dataset (Pont-Tuset et al., 2017), and the RealEstate10K dataset (Zhou et al., 2018), supplemented by high-frame-rate videos from Pixels and YouTube. We filtered out video clips that contain abnormal motions like static or scene changes according to optical flows across consecutive frames. Each training sample contains 25 frames and has spatial resolution 1280×720 similar to (Zhu et al., 2024). We evaluate our methods on two benchmark datasets, the DAVIS-7 dataset by Jain et al. for $\times 8$ interpolation, the evaluation dataset by Zhu et al. (2024) for $\times 24$ interpolation, as they contain diverse motion patterns and objects. Additionally, to ensure data diversity, we created an evaluation set comprising 52 human-selected, high-quality videos from Pixels, producing 100 video clips for $\times 24$ interpolation at a resolution of 1024×576. We report numeric comparison results on both reconstruction metrics, SSIM, and perceptual metrics, including LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017), and FVD (Ge et al., 2024). We introduce FID and FVD for evaluation as they evaluate distribution distances between prediction results and ground truth, which convincingly evaluate VFI quality in large complex motions with various motion patterns.

4.3 Benchmarking

Selected Methods and Setting We compare the proposed RDVFI with several state-of-the-art methods, including conventional non-generative ones, such as RIFE (Huang et al.), FILM (Reda et al.), and AMT (Li et al., 2023), as well as the diffusion-based ones, including LDMVFI (Danier et al.), MoMo (Lew et al., 2025), TRF (Feng et al., 2024), ViBiDSampler (Yang et al., 2024), GI (Wang et al., 2024), FCVG (Zhu et al., 2024), and Wan 1.3B InP (Wan et al., 2025). Because methods may require different training strategies for the best performance, we utilize the released weights without further tuning, as the authors have best tuned them. We evaluate Wan (Wan et al., 2025) with blank text input.

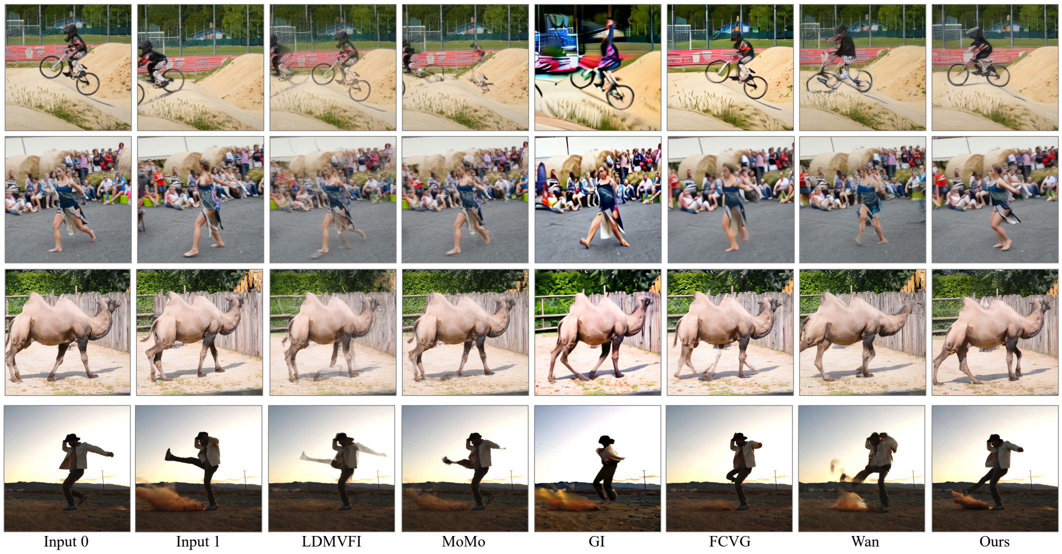

Figure 4: Visual comparison results on DAVIS-7 (Jain et al.) (top two rows) and FCVG (Zhu et al., 2024) (bottom two rows) datasets. Our RDVFI-D method consistently outperforms existing diffusion-based VFI methods with less blurring, ghosting, color shifts, and fractions.

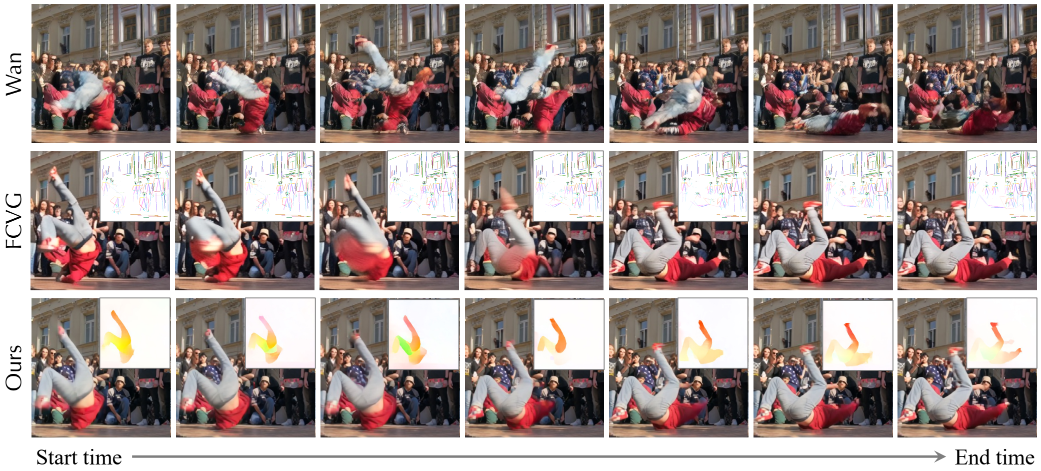

Figure 5: Sequence interpolation comparison. Our RDVFI-D can correctly estimate the continuous motion field between input frames from diffusion outputs and thus interpolate with correct motion and superior visual quality, compared with the direct generation method Wan (Wan et al., 2025) and the linearly-controlled FCVG (Zhu et al., 2024).

Results We report numeric comparison results between the proposed RDVFI and baseline methods in Tab. 1. The results show that our RDVFI outperforms existing baseline methods across the reported two benchmark datasets. Our DiT-based method RDVFI-D outperforms the SVD-based version, RDVFI-U, benefiting from a more powerful backbone network. Image diffusion-based interpolation methods cannot create complex motions between intermediate and input frames, thus producing severe ghosting effects. Directly generating intermediate frames with Video Diffusion Models (VDM) may introduce severe degradations (Feng et al., 2024; Yang et al., 2024; Wang et al., 2024), due to the mismatched interpolation results from separate forward and backward interpolation from two image-to-video SVD (Blattmann et al., 2023) models; FCVG (Zhu et al., 2024) attempts to mitigate such misalignment by involving linear motion controls, however, sacrificing motion generation ability of VDMs and utilizing them as shaders. As shown in Fig. 5, FCVG (Zhu et al., 2024) produces severe degradations when the matching-based linear motion controller does not distinguish the falling boy. Sharing the same backbone, however, our RDVFI-D significantly outperforms Wan (Wan et al., 2025). Our motion-guided decoding pipeline creates intermediate

Methods	DAVIS-7 (Jain et al.)			(Zhu et al., 2024)			Pixels		
	LPIPS↓	FID↓	FVD↓	LPIPS↓	FID↓	FVD↓	LPIPS↓	FID↓	FVD↓
Non-Generative Methods									
AMT (Li et al., 2023)	0.254	34.65	234.50	0.224	44.74	375.00	0.361	41.36	378.86
RIFE (Huang et al.)	0.258	23.98	240.04	0.247	39.01	366.14	0.278	31.10	207.82
FILM (Reda et al.)	0.271	30.16	214.80	0.241	39.82	279.08	0.251	27.06	158.68
Image Diffusion-based Methods									
LDMVFI (Danier et al.)	0.276	22.10	245.02	0.228	37.74	371.49	0.287	31.36	211.70
MoMo (Lew et al., 2025)	0.268	23.67	240.09	0.207	33.59	261.37	0.269	27.31	230.19
Video Diffusion-based Methods									
TRF (Feng et al., 2024)	0.270	29.12	230.12	0.331	45.37	305.88	0.301	35.31	279.65
ViBiDSampler (Yang et al., 2024)	0.261	27.33	208.53	0.292	39.83	257.15	0.263	29.28	184.57
GI (Wang et al., 2024)	0.267	27.71	211.47	0.334	43.08	282.22	0.273	33.27	251.38
FCVG (Zhu et al., 2024)	0.266	25.96	207.17	0.266	31.24	225.48	0.257	24.51	137.57
Wan (Wan et al., 2025)	0.323	26.97	248.14	0.223	28.52	214.45	0.261	22.48	131.22
Our RDVFI-U	0.260	23.65	201.49	0.220	27.03	217.72	0.261	23.71	129.33
Our RDVFI-D	**0.251**	**21.17**	**189.37**	**0.201**	**19.98**	**197.86**	**0.253**	**19.48**	**119.21**

Table 1: Numeric comparison results on three benchmark datasets from (Jain et al.), (Zhu et al., 2024) and Pixels. The best and the second-best results are highlighted by **bold** and underline. The proposed RDVFI outperforms existing baseline methods on most metrics.

Method	GPU Mem. (GB)	Runtime (sec.)
Non-Generative Methods		
AMT	13.5	0.210
RIFE	1.4	0.025
FILM	8.0	0.830
Image Diffusion-based Methods		
LDMVFI	21.7	1.563
MoMo	3.9	0.157

Method	Category	GPU Mem. (GB)	Runtime (sec.)
Video Diffusion-based Methods			
TRF	Zero-shot	13.3	7.382
ViBiDSampler	Zero-shot	26.24	3.708
GI	Fine-tune	23.5	29.613
FCVG	Fine-tune	27.6	14.381
Wan	Fully Trained	18.0	2.579
Our RDVFI-U	Fine-tune	14.2	0.137
Our RDVFI-D	Fine-tune	13.1	0.057

Table 2: We compare the inference efficiency between our RDVFI and existing interpolation baseline methods, including AMT (Li et al., 2023), RIFE (Huang et al.), FILM (Reda et al.), LDMVFI (Danier et al.), MoMo (Lew et al., 2025), TRF (Feng et al., 2024), ViBiDSampler (Yang et al., 2024), GI (Wang et al., 2024), FCVG (Zhu et al., 2024), and Wan (Wan et al., 2025). Our RDVFI is the fastest and most memory-efficient diffusion-based interpolation method. Although RIFE (Huang et al.) is slightly faster than our RDVFI, our RDVFI outperforms it with a clear margin for large complex motions.

frames by warping, forcing the diffusion model to create latents that can correctly restore motions across frames. Thus, our interpolation is smoother and more stable, producing the fewest artifacts compared to other VDM-based solutions.

4.4 Efficiency Comparisons

We report detailed efficiency metrics in Tab. 2. As we can observe, our RDVFI is the fastest diffusion-based interpolation method, that can interpolate at real time (17 FPS) at the resolution of 1024 × 576. Although a non-generative baseline method, RIFE (Huang et al.), is slightly faster than our RDVFI, it cannot deal with large complex motions and introduce severe visual degradations. In contrast, our RDVFI can efficiently and effectively interpolate large complex motions, outperforming RIFE with a clear margin, as shown in Tab. 1, Fig. 4, and Fig. 5.

4.5 Ablation Study

We investigate the effectiveness of the proposed pipeline and training strategy. We utilize the same configuration in Sec. 4.1 for all ablation studies.

Motion-guided Decoding Pipeline We degrade our RDVFI-D by replacing the motion-guided decoding pipeline with the original VAE decoder, which is defined as “VAE Dec”. In addition, we also claim that the iterative motion fusion for continuous motion field estimation is one of our key contributions for accurate motion estimation. Thus, we degrade our motion-guided decoding pipeline by removing the iterative flow fusion, estimating pixel offsets between each key frame and input frames from scratch by regression. We define this setting as “Direct Warping”.

Training Strategy To validate our latent-pixel training strategy, we degrade our training objective by removing the pixel-space perceptual loss only, formatting“L+P-L2 loss”, and all pixel-wise loss, defined as “L loss”.

Resolution	Runtime (ms)	
	VAE-Dec	RDVFI-D
576 × 320	103.3	55.3
1024 × 576	267.7	58.01
1280 × 720	314.3	63.49

(a) Per-frame runtime comparison

Setting	LPIPS↓	FID↓	FVD↓
Direct Warping	0.287	42.33	327.11
L loss	0.269	33.29	281.37
L+P-L2 loss	0.251	29.37	267.42
RDVFI-D	0.224	23.71	209.38

(b) Comparisons on different model settings

Table 3: Ablation study on inference efficiency, network design, and training strategy. Our method outperforms all ablation experiments, showing the best interpolation efficiency and accuracy.

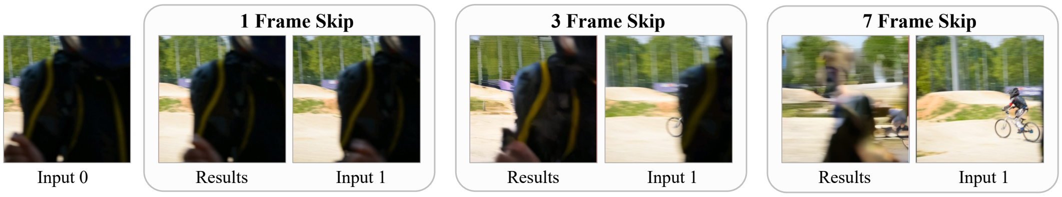

Figure 6: Our RDVFI can robustly interpolate when input frames have consistent objects and enough overlapping regions, such as the 1 frame and 3 frame skip cases. However, when the overlapped regions become minimal, like the ending input frame produced by 7 frames skipped, the algorithm cannot generate coherent and accurate intermediate optical flows to warp input pixels for interpolation, resulting in severe degradations.

As shown in Tab. 3, our RDVFI-D consistently outperforms all degraded versions on efficiency and accuracy. Benefiting from our motion-guided decoding, RDVFI-D can interpolate at different resolutions at the same diffusion sampling cost, resulting in a significant efficiency gain compared with traditional VAE decoders, as shown in Tab. 3a. Tab. 3b also shows the effectiveness of our flow fusion strategy and training objective design. For fair comparison, we fine-tune each ablation experiment and report our results at the same iterations.

4.6 Failure Case

Our method warps input frames with generated intermediate optical flows for interpolation, thus requiring the input frames to contain consistent objects and enough overlapping regions. As shown in Fig. 6, with the same starting frame, our method can robustly interpolate with the ending input frame produced by 1 frame and 3 frames skipped. However, when the overlapped regions become minimal, even with a changed main object in the 7-frame-skipped situation, the algorithm struggles to generate coherent and accurate optical flows to move pixels from the inputs to each intermediate frame, resulting in severe degradations through interpolation.

5 Conclusion and Limitation Discussions

In this paper, we propose the first real-time video diffusion-based Video Frame Interpolation (VFI) pipeline that can run 17FPS at 1024×576 resolution with even superior interpolation quality than current multi-step solutions. Our work is advancing diffusion-based VFI to more practical and challenging scenarios, such as super slow motion and video compression. This work also hopes to suggest that solving ambiguous components with diffusion models, rather than end-to-end generation, may lead to superior accuracy and efficiency.

Limitations Despite the superior interpolation results for complex motions, our method may lag behind conventional non-generative ones for extremely small and simple motions because the continuous motion field degrades to simple functions that existing methods can solve well. Our diffusion sampling may introduce fluctuations, and cannot obtain a significant performance gain compared with existing methods.

Ethics Statement This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation were involved. All datasets used, including DAVIS-7 (Jain et al.) and FCVG (Zhu et al., 2024), were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

Reproducibility Statement We have made every effort to ensure that the results presented in this paper are reproducible. All code, weights, and dataset will be released upon publication. The experimental setup, including training steps, model configurations, and hardware details, is described in detail in the paper. We have also provided a full description of network designs in Sec. 3, network structures in Sec. B, and implementation details in Sec. 4.1, to assist others in reproducing our experiments.

Additionally, the DAVIS-7 (Jain et al.) and FCVG (Zhu et al., 2024) we adopt for evaluation are publicly available, ensuring consistent and reproducible evaluation results.

We believe these measures will enable other researchers to reproduce our work and further advance the field.

References

Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. Jump: virtual reality video. *ACM TOG*, 35(6):1–13, 2016.

Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, pp. 3703–3712, 2019.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. In *AAAI*.

Haiwen Feng, Zheng Ding, Zhihao Xia, Simon Niklaus, Victoria Abrevaya, Michael J Black, and Xuaner Zhang. Explorative inbetweening of time and space. In *ECCV*, pp. 378–395. Springer, 2024.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022a.

Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *CVPR*, pp. 3553–3562, 2022b.

Zhewei Huang, Tianyuan Zhang, et al. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*.

Siddhant Jain, Daniel Watson, Eric Tabellion, Ben Poole, et al. Video interpolation with diffusion models. In *CVPR*.

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 *CVPR*, pp. 9000–9008, 2018.

Jaihyun Lew, Jooyoung Choi, Chaehun Shin, Dahuin Jung, and Sungroh Yoon. Disentangled motion modeling for video frame interpolation. In *AAAI*, volume 39, pp. 4607–4615, 2025.

Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, pp. 9801–9810, 2023.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. In *ICLR*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *ICLR*, 2023.

Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. pp. 41–56. Springer, 2020.

Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *AAAI*, volume 33, pp. 8794–8802, 2019.

Xiaofeng Mao, Zhengkai Jiang, Fu-Yun Wang, Jiangning Zhang, Hao Chen, Mingmin Chi, Yabiao Wang, and Wenhan Luo. Osv: One step is enough for high-quality image to video generation. In *CVPR*, pp. 12585–12594, 2025.

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, pp. 261–270, 2017.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, et al. Film: Frame interpolation for large motion. In *ECCV*.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*.

Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *CVPR*, pp. 17755–17764, 2022.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steven M Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. *ICLR*, 2024.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *NeurIPS*, 36:8406–8441, 2023.

Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *ECCV*, pp. 416–431, 2018.

Chenming Xu, Meiqin Liu, Chao Yao, Weisi Lin, and Yao Zhao. Ibvc: Interpolation-driven b-frame video compression. *Pattern Recognition*, 153:110465, 2024.

Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *NeurIPS*, 32, 2019.

Serin Yang, Taesung Kwon, and Jong Chul Ye. Vibidsampler: Enhancing video interpolation using bidirectional diffusion sampler. *arXiv preprint arXiv:2410.05651*, 2024.

Sushu Yang, Peng Yang, Jiayin Chen, Qiang Ye, Ning Zhang, and Xuemin Shen. Delay-optimized multi-user vr streaming via end-edge collaborative neural frame interpolation. 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pp. 6613–6623, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhixing Zhang, Yanyu Li, Yushu Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Dimitris Metaxas, Sergey Tulyakov, et al. Sf-v: Single forward video generation model. *NeurIPS*, 37:103599–103618, 2024.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Tianyi Zhu, Dongwei Ren, Qilong Wang, Xiaohe Wu, and Wangmeng Zuo. Generative inbetweening through frame-wise conditions-driven video generation. *arXiv preprint arXiv:2412.11755*, 2024.

In addition to our paper, we also provide a video demo to compare the proposed RDVFI against existing diffusion-based interpolation methods. Because many artifacts, such as unreasonable generated motions and detail inconsistency, are hard to observe in the image domain, we strongly suggest the reviewers to watch our demo for performance comparisons.

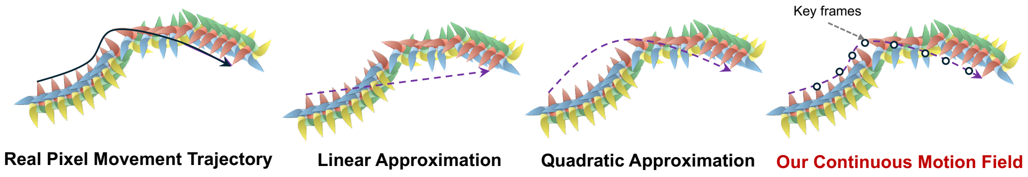

Figure 7: Existing simple functions (linear and quadratic) cannot fit real pixel movements. Our continuous motion field utilizes more complex movement splines, fitting these complex movements better and accordingly producing reliable interpolation results.

A More Explanations to the Continuous Motion Field

Existing data-driven non-parametric methods generate optical flows between the middle and input frames. They require iterative interpolation for frame sequences, resulting in inconsistent interpolation and blurring effects due to multiple warping operations. Parametric methods can simultaneously generate optical flows between all intermediate and input frames from the constructed object movement function, leading to more consistent motion generation and exquisite details.

However, approximating real pixel movement trajectories parametrically is challenging because we do not have enough known frames to fit complex mathematical functions. Existing solutions adopt simple functions, such as linear and quadratic, to approximate pixel movements. However, as shown in Fig. 7, they cannot fit fast-changing real-world motions, hindering interpolation quality. Our continuous motion field solves this challenge by involving the one-step diffusion model to generate sparse key frames to enable more complex functions for approximating real pixel movements. The continuous motion field obviously shows superior motion approximation ability and leads to superior interpolation accuracy for large, complex motions.

B Network Details

B.1 Video Diffusion Model

We build our VFI Video Diffusion Models (VDM) upon SVD (Blattmann et al., 2023) and Wan 2.1 1.3B T2V (Wan et al., 2025) model. We do not apply LoRA (Hu et al., 2022a) fine-tuning technique for both models because supervised fine-tuning leads to significant performance gain. To avoid overfitting and backbone model degradation through fine-tuning, we collect and train our diffusion on a large-scale dataset that consists of more than 500K valid video clips with random data augmentation, like horizontal flipping, random resizing, and cropping. We do not utilize the pretrained image-to-video SVD because it only conditions on the first input frame, introducing different frame information biases than VFI, as frame information decreases with their distance to known frames. We also do not utilize the Wan InP model, which supports start-end-frame generation because it injects input frame conditions by concatenating input and intermediate noisy latent features along the channel dimension, potentially introducing inconsistent interpolation. Thus, we initialize our VDM with the T2V diffusion weights, concatenating input and intermediate latent features along the temporal dimension, and modify the timestep embedding by assigning input latent features the least noise level in the diffusion scheduler.

B.2 Continuous Motion Field Estimator

Our continuous intermediate motion field estimator consists of two networks: $\phi_1^f(\cdot)$ and $\phi_2^f(\cdot)$. The detailed network working flow and structures are shown in Algo. 2 and Tab. 4, respectively. We use

```
Algorithm 2: Continuous Intermediate Motion Estimator
Input: Denoised latent features $z_0, z_{\tau_0}, ..., z_{\tau_{N-1}}, z_1$; Two neural networks $\phi_1^f$, $\phi_2^f$
Output: Flows for each interpolation $\{f_{0\to\tau}, f_{1\to\tau}\}_{\tau=\tau_0}^{\tau_{N-1}}$
Calculate $f_{\tau_i\to\tau_{i\pm1}} = \phi_1(z_i, z_{i\pm1})$;
Initialize $f_{0\to\tau_{-1}} = 0, f_{1\to\tau_N} = 0$;
foreach i=0, 1,..., N-1 do
    Warp $z_0$ with previous flow estimation results $z_{0\to\tau_{i-1}} = Warp(f_{0\to\tau_{i-1}}, z_0)$;
    Flow fusion for time $\tau = \tau_i$: $f_{0\to\tau_i} = \phi_2(z_0, z_{\tau_i}, z_{0\to\tau_{i-1}}, f_{0\to\tau_{i-1}}, f_{\tau_{i-1}\to\tau_i}) + f_{0\to\tau_{i-1}}$
foreach i=N-1, N-2, ..., 0 do
    Warp $z_0$ with previous flow estimation results $z_{1\to\tau_{i+1}} = Warp(f_{1\to\tau_{i+1}}, z_1)$;
    Flow fusion for time $\tau = \tau_i$: $f_{1\to\tau_i} = \phi_2(z_1, z_{\tau_i}, z_{1\to\tau_{i+1}}, f_{1\to\tau_{i+1}}, f_{\tau_{i+1}\to\tau_i}) + f_{1\to\tau_{i+1}}$
```

the same $\phi_1^f(\cdot), \phi_2^f(\cdot)$ for estimation (same network, same weights) for all intermediate frames and both flow fusion directions.

B.3 Frame Synthesis

We synthesize intermediate frames with estimated optical flows by borrowing IFBlocks from RIFE (Huang et al.), shown in Tab. 5. The frame synthesis network $\phi^s(\cdot)$ two IFBlocks at $\frac{1}{4}$ and $\frac{1}{2}$ scales, respectively. We warp frame using refined optical flows, resulting in $I_{0\to\tau_j}, I_{1\to\tau_j}$, respectively, and fuse with the mask m by:

$$I_{\tau_j} = sigmoid(m) \cdot I_{0\to\tau_j} + (1 - sigmoid(m)) \cdot I_{1\to\tau_j}. \tag{6}$$

where I_{τ_j} is the interpolated frame.

#	Operation	Input
	ϕ_1	
1	Concatenate	$z_{k_i}, z_{k_{i+1}}$
2	Conv 3 × 3 × 256	#1
3	Conv 3 × 3 × 256	#2
4	LeakyReLU(0.2)	#3
5	Conv 3 × 3 × 256	#4
6	LeakyReLU(0.2)	#5
7	Conv 3 × 3 × 256	#6
8	LeakyReLU(0.2)	#7
9	Add	#2, #8
10	Conv 3 × 3 × 2	#9
	ϕ_2	
11	Warp	#10, z_0
12	Concatenate	#11, z_0, $z_{k_{i+1}}$, $f_{0\to k_i}$, #10
13	Conv 3 × 3 × 256	#12
14	LeakyReLU(0.2)	#13
15	Conv 3 × 3 × 256	#14
16	LeakyReLU(0.2)	#15
17	Conv 3 × 3 × 256	#16
18	LeakyReLU(0.2)	#17
19	Add	#13, #18
20	Conv 3 × 3 × 2	19
21	Add	#20, $f_{0\to k_{i+1}}$

Table 4: Network implementation details of the continuous motion field estimator networks. Convolution specifications are given in order kernel height × kernel width × output channel; negative slope of LeakyReLU is in the parentheses. We take the forward flow fusion as an example and can simply estimate the backward flow fusion results using the same network but in reverse order.

#	Operation	Input
1	Warp	I_0, $f_{0\to\tau_j}$
2	Warp	I_1, $f_{1\to\tau_j}$
3	Concatenate	m, #1, #2, I_0, I_1, $f_{0\to\tau_j}$, $f_{1\to\tau_j}$
4	Resize	#3
5	Conv $3 \times 3 \times C$	#4
6	LeakyReLU(0.2)	#5
7	Conv $3 \times 3 \times C$	#6
8	LeakyReLU(0.2)	#7
9	Conv $3 \times 3 \times C$	#8
10	LeakyReLU(0.2)	#9
11	Add	#10, #5
12	Conv $3 \times 3 \times 5$	#11
13	Add	First two channels of #12, $f_{0\to\tau_i}$
14	Add	Following two channels of #12, $f_{1\to\tau_i}$
15	Add	Last channel of #12, m

Table 5: Network implementation details of the IFBlock networks. Convolution specifications are given in order kernel height × kernel width × output channel; negative slope of LeakyReLU is in the parentheses. Convolution channel C varies with resize scales, which are 256 and 128 for 0.25 and 0.5 scales, respectively. We take the forward flow fusion as an example and can simply estimate the backward flow fusion results using the same network but in reverse order.

#	Operation	Input
1	IFBlock at $1/4$ scale	$I_0, I_1, f_{0\to\tau_j}, f_{1\to\tau_j}$
2	IFBlock at $1/2$ scale	#1, I_0, I_1
3	Gradient Detach	z_{k_i}
4	Gradient Detach	$z_{k_{i+1}}$
5	Warp	#3, $f_{k_i\to\tau_j}$
6	Warp	#4, $f_{k_{i+1}\to\tau_j}$
7	Concatenate and Upscale	#3, #4
8	Conv $3 \times 3 \times 64$	#7
9	Conv $3 \times 3 \times 64$	#8
10	LeakyReLU(0.2)	#9
11	Conv $3 \times 3 \times 64$	#10
12	LeakyReLU(0.2)	#11
13	Conv $3 \times 3 \times 64$	#12
14	LeakyReLU(0.2)	#13
15	Add	#8, #14
16	Conv $3 \times 3 \times 3$	#15

Table 6: Full network implementation details of the frame synthesis network.

As shown in Tab. 6, we then refine the fused intermediate frame $\hat{I}_{\tau_j}$ with denoised latent features for variations that cannot be modeled by flows. We first detach the gradient of denoised key frame latent features $\{z_{k_i}, z_{k_{i+1}}\}$, where $z_{k_i} < \tau_j < z_{k_{i+1}}$ to force the frame synthesis network interpolate with correct motion rather than directly decoding latent features during training. Then, we warp these latent features with sampled optical flow $\{f_{k_i\to\tau_j}, f_{k_{i+1}\to\tau_j}\}$. We upscale the warped latent to the same resolution as the input frames, following a resblock at 64 channels. Finally, we add them to the fused intermediate frame to generate the final interpolation results $\hat{I}_{vfi}$ at time $\tau = \tau_j$.

C More Visualization

We provide additional visualization in Fig.8 and Fig.9. As we can observe, existing video interpolation methods cannot deal with nonlinear, non-rigid motions, interpolating by pixel-wise approximation and resulting in severe visual degradations. In contrast, our RDVFI can accurately decompose

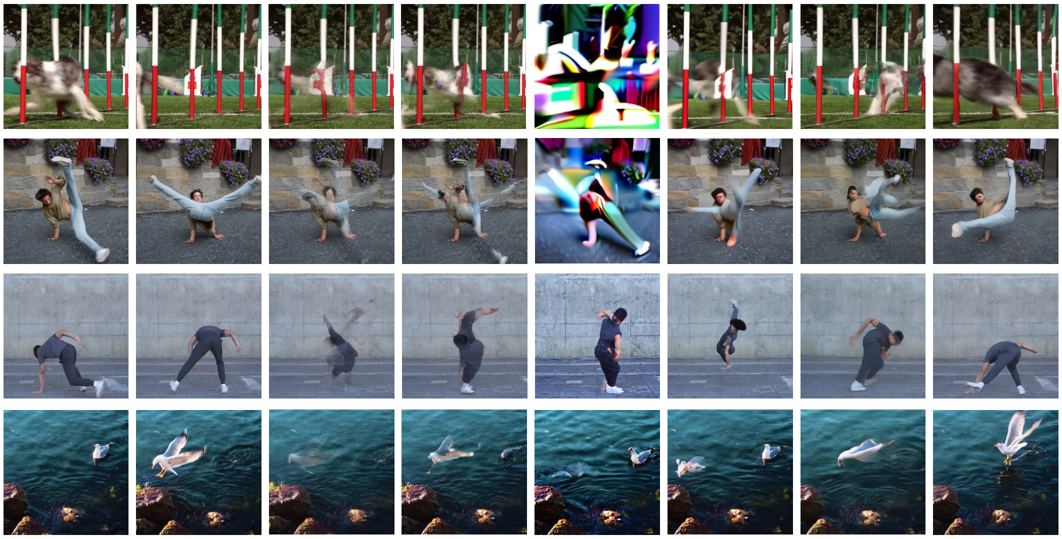

Figure 8: More visualization on challenging cases. The first two rows from DAVIS-7 dataset (Jain et al.), and the last two from Pixels. Our RDVFI consistently outperforms existing video frame interpolation methods on large complex motions.

large complex motions between input frames into small and easy-to-estimate components with diffusion models and iteratively fuse them for interpolation. Accordingly, our RDVFI can interpolate these challenging motions with superior visual quality.

D Discussion about Limitations

As shown in Fig.10, both non-generative methods (Huang et al.; Reda et al.) and diffusion-based methods (our RDVFI and Wan (Wan et al., 2025)) can interpolate small and simple motions well. However, non-generative methods only focus on small and simple motions during training, while both Wan (Wan et al., 2025) and our RDVFI deal with both small simple motions and large complex ones. As a result, these non-genrative methods usually achieve superior numeric metrics.

E LLM Usage

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis.

The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

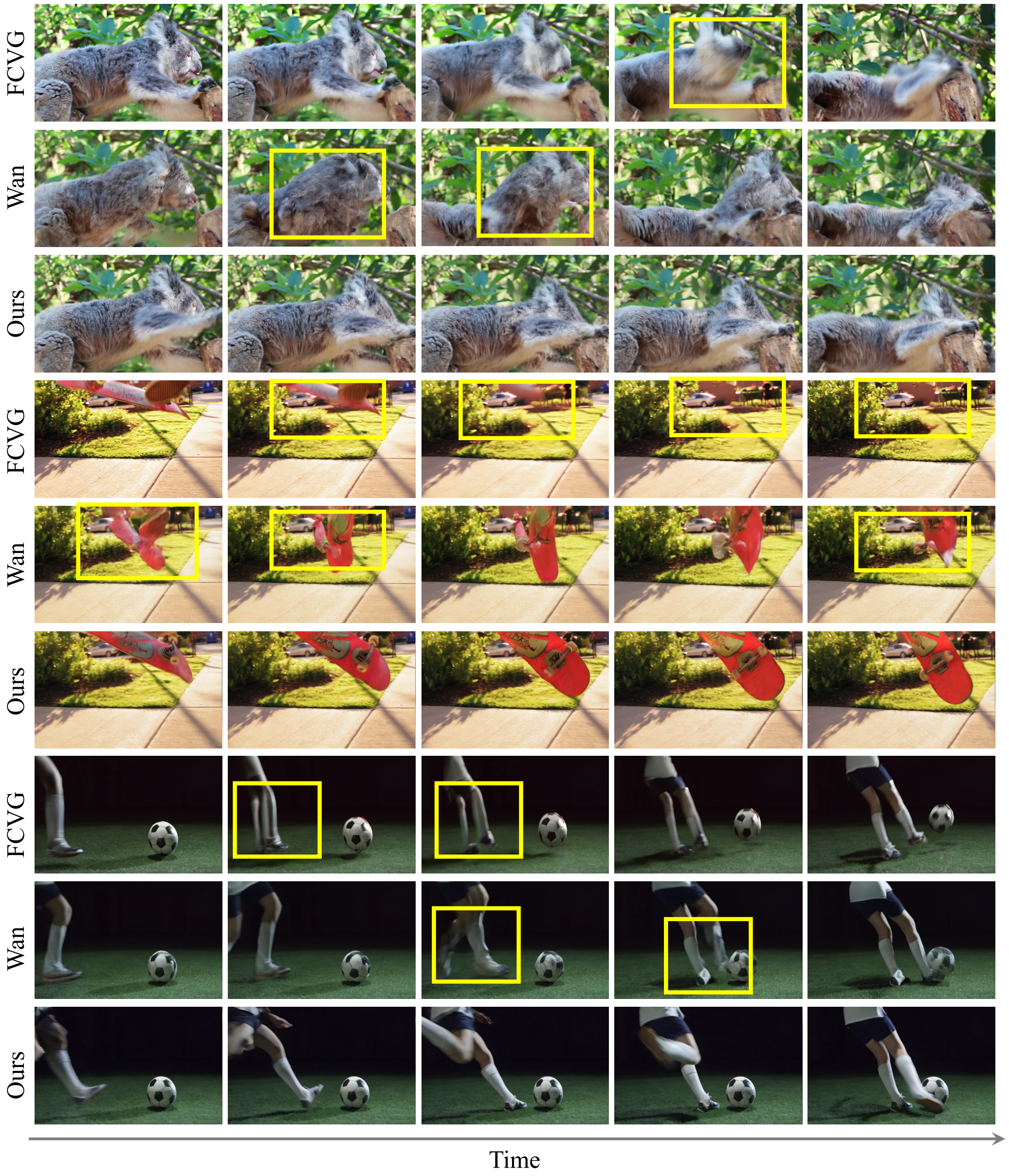

Figure 9: More sequence interpolation results. The first row comes from the FCVG dataset (Zhu et al., 2024) and the last two from Pixels dataset. Current video diffusion-based methods (Zhu et al., 2024; Wan et al., 2025) cannot generate accurate motions when they are nonlinear and non-rigid, performing pixel-wise approximation for interpolation that results in severe degradations (see yellow boxes). In contrast, our RDVFI can better solve these challenging motions, resulting in superior interpolation quality.

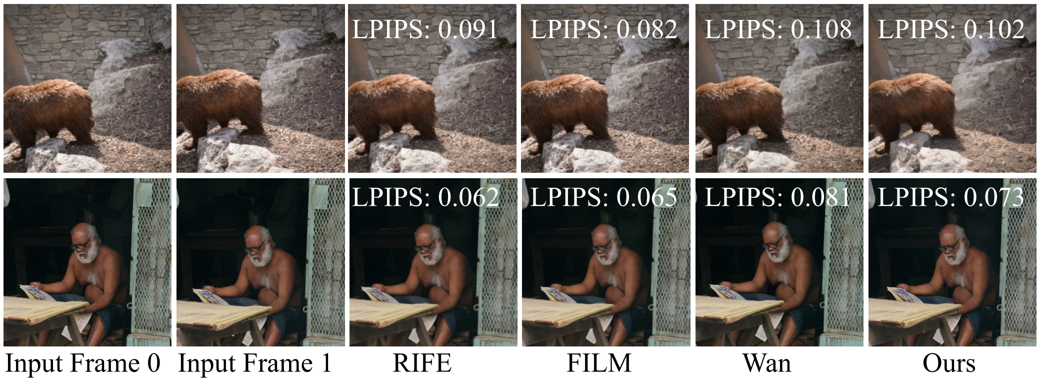

Figure 10: Non-generative methods can interpolate small and simple motions well, as they can approximate intermediate motions with the pre-determined functions. Although our RDVFI and diffusion-based interpolation methods, such as Wan (Wan et al., 2025) can interpolate well to these motions, non-generative models usually have superior numeric metrics as they only focus on small and simple motions.