High-Resolution Frame Interpolation with Patch-based Cascaded Diffusion

Junhwa Hur*, Charles Herrmann*, Saurabh Saxena, Janne Kontkanen, Wei-Sheng Lai, Yichang Shih, Michael Rubinstein, David J. Fleet†, Deqing Sun

Google

Abstract

Despite the recent progress, existing frame interpolation methods still struggle with processing extremely high resolution input and handling challenging cases such as repetitive textures, thin objects, and large motion. To address these issues, we introduce a *patch-based* cascaded pixel diffusion model for high resolution frame interpolation, HiFI, that excels in these scenarios while achieving competitive performance on standard benchmarks. Cascades, which generate a series of images from low to high resolution, can help significantly with large or complex motion that require both global context for a coarse solution and detailed context for high resolution output. However, contrary to prior work on cascaded diffusion models which perform diffusion on increasingly large resolutions, we use a single model that always performs diffusion at the same resolution and upsamples by processing patches of the inputs and the prior solution. We show that this technique drastically reduces memory usage at inference time and also allows a single model at test time, solving both frame interpolation (base model's task) and spatial up-sampling, saving training cost. We show that HiFI helps significantly with high resolution and complex repeated textures that require global context. HiFI demonstrates comparable or beyond state-of-the-art performance on multiple benchmarks (Vimeo, Xiph, X-Test, and SEPE-8K). We further introduce a new dataset, LaMoR, that focuses on particularly challenging cases, and HiFI also significantly outperforms other baselines. Please visit our project page for video results: https://hifi-diffusion.github.io

Introduction

In a short amount of time, smartphone cameras have become both ubiquitous and significantly higher quality, capturing spatially higher resolution images and videos. However, the temporal resolution—*i.e*. video frame rate—of captured videos has lagged behind the spatial resolution, due to a combination of computational and memory costs and limited exposure time. The conflict between increased user interest in creative video content and technical limitations for capturing high frame-rate video has increased interest in techniques for high-resolution frame interpolation, which enables the synthesis of new frames between existing ones to enhance a video's frame rate. Despite the progress, the latest techniques struggle in the high resolution setting, where challenging cases such as repetitive textures, detailed or thin objects become more common place.

Existing methods often design models using strong domain knowledge, *e.g*., correspondence matching (Ranjan and Black 2017; Ilg et al. 2017; Sun et al. 2018; Teed and Deng 2020) and synthesis based on warping (Hu et al. 2022; Jiang et al. 2018; Niklaus and Liu 2020; Park, Lee, and Kim 2021; Xue et al. 2019). Domain knowledge enables small models to perform well when trained on a small amount of data but may restrict their capabilities. For example, when motion cues are incorporated into the model, the final quality are bounded by the accuracy of the motion. This is particularly evident on high resolution inputs with large motion, repetitive texture, and thin structures, where motion estimation often struggles (see Fig. 1).

To address these challenges, we advocate a domain-agnostic diffusion approach, relying on model capacity and training data at scale for performance gains and generalization. Some recent work have explored diffusion for frame interpolation but towards generative aspect, *e.g*. better perceptual quality (Danier, Zhang, and Bull 2024) or complex and non-linear motion (Jain et al. 2024) between two frames very further apart in time. Their performance, however, falls behind in the classical setting which predicts an intermediate frame and evaluates its fidelity to the ground truth using standard metrics, *e.g*., PSNR or SSIM.

We instead introduce a *patch-based* cascaded pixel diffusion approach for **Hi**gh resolution **F**rame **I**nterpolation, dubbed HiFI. HiFI generalizes across diverse resolutions up to 8K images, a wide range of scene motions, and a broad spectrum of challenging scenes. The diffusion framework allows us to scale both the model capacity and data size. We also show that our model can effectively utilize large-scale video datasets. While cascades offer significant benefits for processing diverse input resolutions with different levels of motion, standard cascades, which denoise the entire high-resolution image, often struggle with memory issues at very high resolutions such as 8K. To save memory during inference, we propose a new *patch-based* cascade for frame interpolation, which always denoises the same resolution but

*These authors contributed equally.

†DF is also affiliated with the University of Toronto and the Vector Institute.

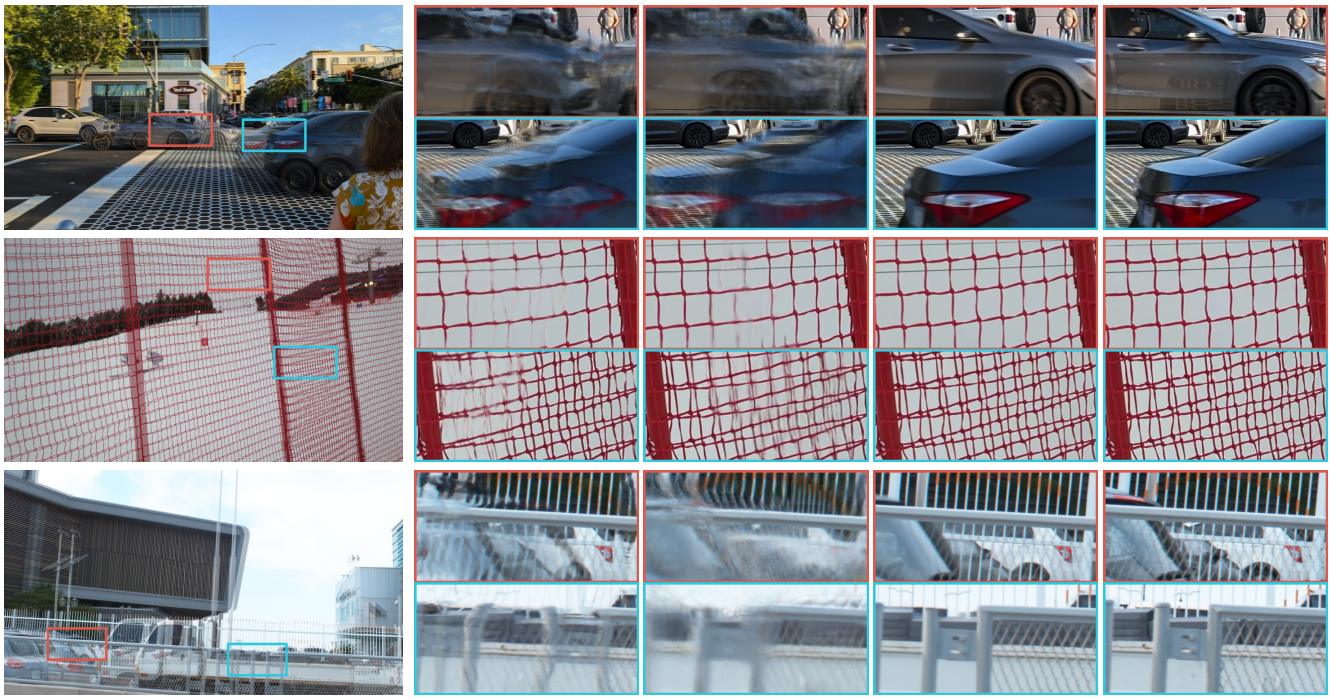


Figure 1: **Qualitative comparison on challenging cases** on our proposed LaMoR dataset (rows 1 and 2) and X-TEST (row 3). For challenging cases, such as large motion or repetitive textures, the proposed HiFI substantially outperforms other baselines.

is applied to patches of high resolution frames. This also allows us to use one model for both base and super-resolution tasks, saving time for training separate models for both tasks and disk space at inference time.

The proposed HiFI method achieves state-of-the-art accuracy on challenging high-resolution public benchmark datasets, Xiph (Niklaus and Liu 2020), X-TEST (Sim, Oh, and Kim 2021) and SEPE (Al Shoura et al. 2023), and demonstrates strong performance on challenging corner cases, *e.g.*, repetitive textures and large motion. We also introduce a new evaluation dataset, Large Motion and Repetitive texture (LaMoR), which specifically highlights these challenging cases and demonstrate that HiFI significantly outperforms existing baselines.

Related work

Domain-specific architecture for interpolation. Motion-based approaches synthesize intermediate frames using estimated bi-directional optical flow between two nearby frames. These methods employ forward splatting (Hu et al. 2022; Jin et al. 2023; Niklaus and Liu 2018, 2020) or backward warping (Huang et al. 2022; Jiang et al. 2018; Park, Lee, and Kim 2021; Park et al. 2020; Sim, Oh, and Kim 2021), followed by a refinement module that improves visual quality. Performance is often bounded by motion estimation accuracy, as inaccuracies in the motion cause artifacts during the splatting or warping process. As a result, they struggle on inputs for which optical flow estimation is problematic, *e.g.*, large motion, occlusion, and thin objects.

Phase-based approaches (Meyer et al. 2015, 2018) propose to estimate an intermediate frame in a phase-based representation instead of the conventional pixel domain. Kernel-based approaches (Cheng and Chen 2020; Lee et al. 2020; Niklaus, Mai, and Wang 2021; Niklaus, Mai, and Liu 2017a,b) present simple single-stage formulations that estimate per-pixel $n \times n$ kernels and synthesize the intermediate frame using convolution on input patches. Both approaches avoid reliance on motion estimator, but they do not usually perform well on high resolution input with large motion, even with deformable convolution (Cheng and Chen 2020).

Generic architecture for interpolation. Some methods explore using a generic architecture without domain knowledge, such as attention (Choi et al. 2020), transformer (Shi et al. 2022), 3D convolution under multi-frame input setup (Kalluri et al. 2023; Shi et al. 2022). However, both attention and 3D convolution are computationally expensive and thus prohibitive at 4K or 8K resolution.

Diffusion models for computer vision. Recently diffusion models have demonstrated their strength on generative computer vision applications such as image (Ho et al. 2022a; Peebles and Xie 2023; Rombach et al. 2022) and video generation (Blattmann et al. 2023; Ge et al. 2023; Ho et al. 2022b), image editing (Brooks, Holynski, and Efros 2023; Yang, Hwang, and Ye 2023), 3D generation (Qian et al. 2024; Shi et al. 2024b), *etc*. Beyond generation, diffusion has also shown to be effective for dense computer vision tasks and has been become the state-of-the-art technique for classical problems such as depth prediction (Ke et al. 2024; Saxena et al. 2023), optical flow prediction (Saxena et al.

2023), correspondence matching (Nam et al. 2024), semantic segmentation (Baranchuk et al. 2022; Xu et al. 2023), *etc*.

Diffusion models for interpolation. Two recent works explore diffusion for video frame interpolation from a generative perspective. LDMVFI (Danier, Zhang, and Bull 2024) proposes using a conditional latent diffusion model and optimizes it for perceptual frame interpolation quality, but the PSNR or SSIM metric of the predicted frames tends to be lower than that by state of the art. VIDIM (Jain et al. 2024) uses a cascaded pixel diffusion model but focuses on a task closer to the conditional video generation. Given two temporally-far-apart frames, the method generates a base video of 7 frames at 64×64 resolution and then upsamples them to 256×256. It is unclear whether a diffusion-based approach can achieve competitive results on the classical frame interpolation problem, where the input frames come from a video with high FPS and can be up to 8K resolution.

Cascaded diffusion models. Beginning with CDM (Ho et al. 2022a), cascades have become standard for scaling up the output resolution of pixel diffusion models. Diffusion cascades consist of a "base" model for an initial low-resolution solution and a number of separate "super-resolution" models to produce a higher-resolution output conditioned on the low resolution output. While effective for high-resolution output, memory cost increases proportionally with resolution since each super-resolution model performs diffusion at its output resolution. Even with specialized super-resolution architectures (Ho et al. 2022a; Saharia et al. 2022), the memory problem still persists as the target resolution increases significantly, *e.g*. from 1K to 8K.

High resolution diffusion. Recent works in high-resolution image generation have introduced training-free approaches through merging the score functions of nearby patches (Bar-Tal et al. 2023; Liu et al. 2024b) or expanding the network (Shi et al. 2024a; Kim, Hwang, and Park 2024). More recent work introduces models that are explicitly trained to denoise partitioned patches (or tiles) and then merge them into high-resolution output. Zheng et al. (2024) proposes to generate any-size high-resolution output by merging denoised non-overlapping tiles during sampling process. Ding et al. (2024) introduces to use score value and feature maps to encourage consistency between denoised patches. Skorokhodov et al. (2024) uses a hierarchical patch structure for efficient video generation, but requires specialized modules for global consistency.

Unlike these efforts, we focus on frame interpolation, an estimation task, and improve inference memory efficiency at extremely high resolutions (4K or 8K). In fact, patch-based techniques are well-suited for estimation tasks. Generation tasks require communication between patches for consistent and coherent generation at high resolution. Estimation tasks, however, benefit from strong conditioning signal (input frames), which provides this context and make the problem more localized to the patch level. This difference allows us to explore distinct architectural design choices.

For estimation tasks, the most similar to ours is DDVM (Saxena et al. 2023) which uses tiling for high-resolution inference. After the base model runs at a coarse solution, the output is upsampled by partially denoising tiles taken from the coarse solution and input frames. In the context of frame interpolation, we show that this tiling performs worse than our proposed patch-based cascade.

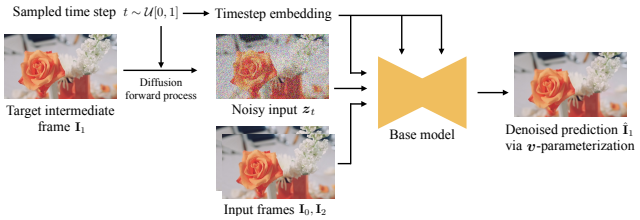


Figure 2: Our **base model** is conditioned on two input frames, $\mathbf{I}_0$ and $\mathbf{I}_2$, and predicts the intermediate frame $\mathbf{I}_1$. The model uses $\boldsymbol{v}$-parameterization (Salimans and Ho 2022; Saxena et al. 2024) for both model output and loss.

Diffusion for high-resolution frame interp.

We explore a pixel diffusion approach for classical frame interpolation and propose a patch-based cascade strategy for high-resolution inference by performing diffusion on patches of high resolution inputs. Our patch-based cascade enables high resolution output with low memory usage at inference time and allows us to use the same model for both base estimation and upsampling.

Architecture

Our method adopts a conditional image diffusion framework. Given a concatenation of temporally nearby frames $\mathbf{I}_0$ and $\mathbf{I}_2$ as a conditioning signal, our model aims to estimate an intermediate frame $\hat{\mathbf{I}}_1$ as a reverse diffusion process in the pixel space, as illustrated in Fig. 2. We take a generic efficient U-Net architecture from DDVM (Saxena et al. 2023) with $\boldsymbol{v}$-parameterization (Salimans and Ho 2022; Saxena et al. 2024) for both model output and loss. The U-Net includes self-attention layers at two bottom levels. Given a noisy image $\boldsymbol{z}_t = \alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\epsilon}$ as an input, the network predicts $\hat{\boldsymbol{v}}$, where $\boldsymbol{x}$ is the target image (*i.e*., $\mathbf{I}_1$), sampled random noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, sampled time step $t \sim \mathcal{U}[0, 1]$, and $\alpha_t^2 + \sigma_t^2 = 1$. We directly apply L1 loss on $\boldsymbol{v}$ parameter space, *i.e*., $||\hat{\boldsymbol{v}} - \boldsymbol{v}||_1$, where $\boldsymbol{v} = \alpha_t \boldsymbol{\epsilon} - \sigma_t \boldsymbol{x}$. The predicted image is recovered by $\hat{\boldsymbol{x}} = \alpha_t \boldsymbol{z}_t - \sigma_t \hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{x}} = \hat{\mathbf{I}}_1$.

Patch-based cascade model

Since our main focus is on extremely high resolutions (up to 8K), standard cascades, which denoise the entire high resolution image directly, would require either a considerable amount of memory at inference time or a careful architecture search to reduce the memory cost. We instead advocate for a patch-based cascade approach that performs diffusion at the same resolution on patches of the input frames. This avoids both of these issues, keeping the peak memory usage at inference time near constant and allowing us to use the same architecture for every upsample level. We also find that can re-use the same model in both the base and super-resolution settings, saving training time and disk space at inference.

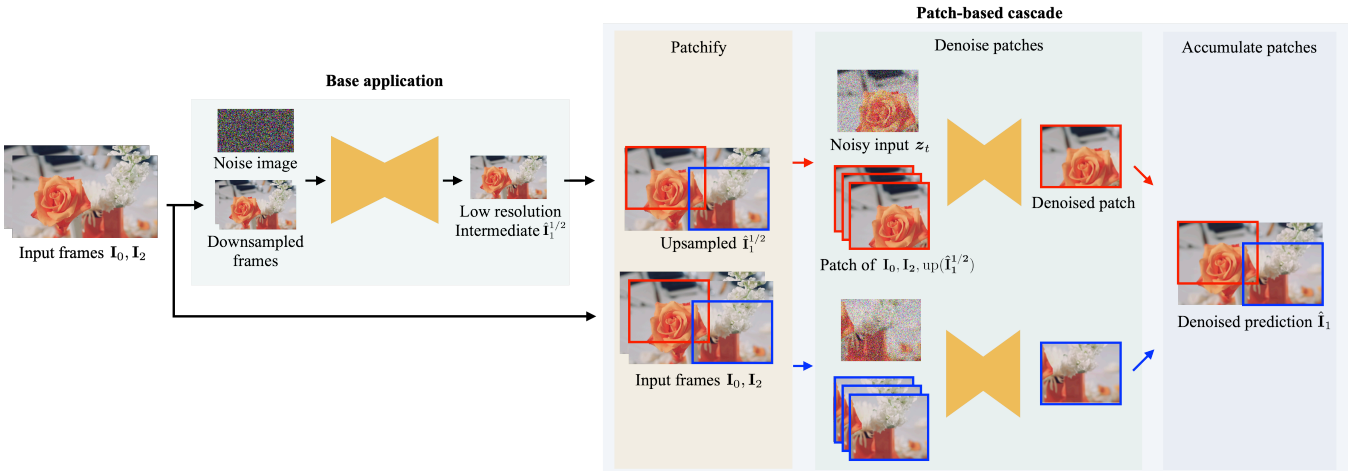


Figure 3: **Patch-based cascade model**. Given a low-resolution intermediate from the previous level, patch-based cascade creates patches from bi-linearly upsampled low-resolution intermediate and two input frames and uses these patches as conditioning for a diffusion process. It then combines denoised patches to form the whole image. At inference time, only a single weight-shared model is recursively used across different image scales as in Fig. 4. Two-stage cascade is shown for simplicity.

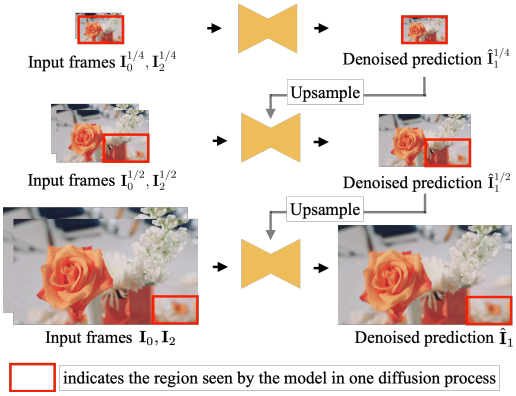


Figure 4: **Upsampling strategy**. Like a standard cascade, we process the image from coarse to fine, but we always denoise at the same resolution, as indicated by the red box. Details on each step of the cascade are in Fig. 3.

Approach. Fig. 4 shows our overall inference strategy: we adopt the well-known coarse-to-fine idea for cascades and build an N-level image pyramid. Starting from the lowest scale s_{N-1} (where $s_n \equiv 1/2^n$), we downsample the input conditioning images by a factor of s_{N-1} (*i.e.*, $\mathbf{I}_0^{s_{N-1}}$ and $\mathbf{I}_2^{s_{N-1}}$) and predict an intermediate image $\hat{\mathbf{I}}_1^{s_{N-1}}$ at the same scale s_{N-1}. We then apply $2\times$ bilinear upsampling to this intermediate image $\text{up}(\hat{\mathbf{I}}_1^{s_{N-1}})$ and use it as a conditioning signal for a denoising process at the scale s_{N-2}.

At each pyramid level, we upscale prediction via patch-based cascade, as shown in Fig. 3. For refinement at scale s_{N-2}, we take the prediction from the prior scale and then upsample it to match s_{N-2}. At each level, we perform three stages: *(i)* patchify, where we crop overlapping patches from the upsampled intermediate prediction and the input at that scale, *(ii)* denoise patches, where we run diffusion to obtain the prediction for each patch, and *(iii)* accumulate patches, where we use MultiDiffusion (Bar-Tal et al. 2023) to merge results from different patches for the prediction $\hat{\mathbf{I}}_2^{s_{N-2}}$. Here, we merge denoised patches at every denoising step. We then upsample $\hat{\mathbf{I}}_2^{s_{N-2}}$ to level s_{N-3} and repeat this process until $n=0$, the original input scale.

Training setup. For the patch-based cascade model, we want to train a diffusion model that is conditioned on a pair of input images and a half resolution representation of the target we aim to predict. We first predict the intermediate frame at a half resolution $\hat{\mathbf{I}}_1^{1/2}$ by feeding downsampled inputs to a pre-trained base model (Fig. 2) computed offline. This intermediate frame is then upsampled to the original scale and used as a conditioning image, along with the original inputs, for training the patch-based cascade model using standard diffusion. This inference step is performed offline to improve training efficiency.

Single model for all stages. By conditioning on the low resolution estimate but using dropout 50% during training time, we can use the same model for all cascade stage, including base and super-resolution; base generation is done by passing zeros as the low resolution condition. While similar to CFG (Ho and Salimans 2022), we do not combine unconditional and conditional estimations at inference. Empirically we find that a single shared model for all stages performs slightly better than having a dedicated super-resolution model. It also substantially reduces training time (training one model instead of multiple separate ones) and disk space at inference time (saving only one model). Interestingly, we observe that a dedicated super-resolution model trained without dropout on the coarse estimation does not work since it takes the shortcut of upsampling the low resolution instead of attending to the high resolution inputs.

Experiments

Implementation details. Similar to previous diffusion-based methods (Jain et al. 2024; Danier, Zhang, and Bull

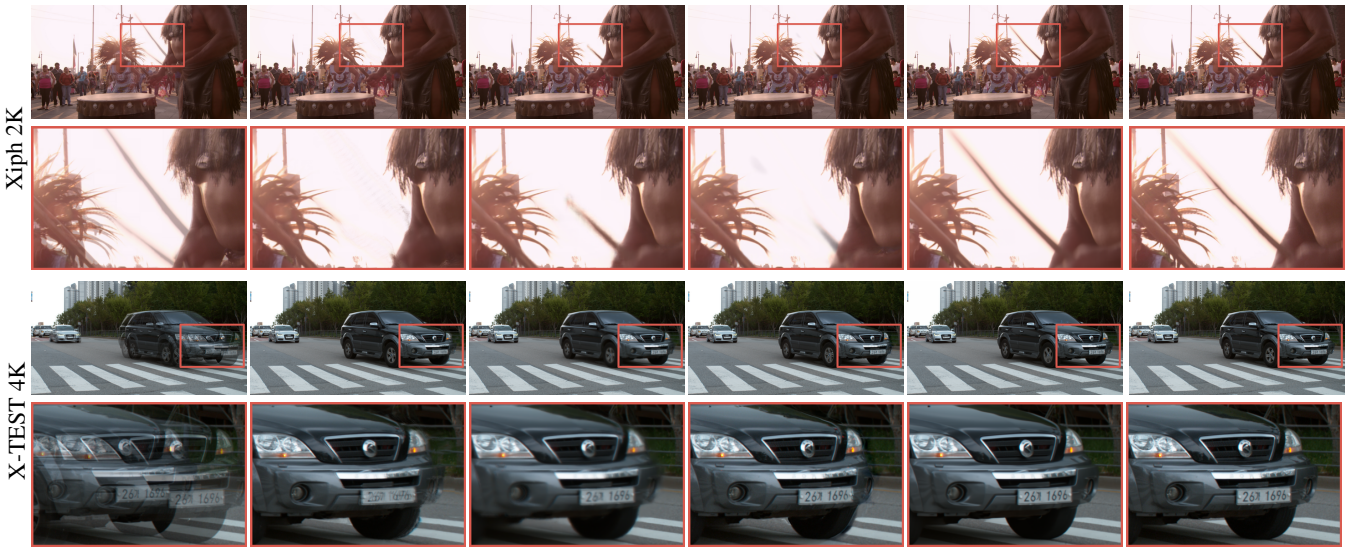


Figure 5: **Qualitative examples for public datasets.** Our method performs well even in cases of large motion and complex textures such as a thin object on the top and the plate number at the bottom.

2024), we utilize a large-scale video dataset for training, to test the scalability of the diffusion model better. The dataset contains up to 30 M videos with 40 frames, collected from the internet and other sources with licenses permitting research uses. We first train our base model on the dataset, and then we additionally include Vimeo-90K triplet (Xue et al. 2019) and X-TRAIN (Sim, Oh, and Kim 2021) to finetune the cascade model. For fair comparison, we also prepare a model trained on Vimeo-90K and X-TRAIN only from scratch. We use a mini-batch size of 256 and train the base model for 3 M iteration steps and the patch-based cascade model for 200 k iteration steps. We use the Adam optimizer (Kingma and Ba 2014) with a constant learning rate $1e^{-4}$ with initial warmup. For inference, we use 3-stage patch-based cascade setup with a patch size of 512×768, averaging 4 samples estimated via 4 sampling steps.

Our data augmentation includes random crop and horizontal, vertical, and temporal flip with a probability of 50%. We use a crop size of 352×480 for large-scale base model training and 224×288 for the cascade model training. We use a multi-resolution crop augmentation that crops an image patch with a random rectangular crop size between the original resolution and the final crop size and then resize it to the final crop size. While commonly used, we find random 90° rotation augmentation and photometric augmentation to be less effective, so we opt not to use them.

More details are in the supplementary material.

Public benchmark evaluation

We first evaluate HiFI on three popular benchmark datasets, Vimeo-90K triplet (Xue et al. 2019), Xiph (Niklaus and Liu 2020), and X-TEST (Sim, Oh, and Kim 2021), as shown in Table 1, as well as an 8K dataset, SEPE (Al Shoura et al. 2023).

Vimeo-90K. The low-resolution (256×448) Vimeo-90K is one of the most heavily studied benchmark, where numbers are highly saturated among different methods. HiFI achieves competitive accuracy with a generic training procedure. Please view the supplementary for further discussion.

Xiph and X-TEST. Both Xiph and X-TEST have high resolution (2K and 4K). The motion of X-TEST can be over 400 pixels at the 4K resolution, particularly challenging for existing methods. For X-TEST, we follow the evaluation protocol discussed in (Sim, Oh, and Kim 2021) that interpolates 7 intermediate frames. When trained on a combination of Vimeo and X-TRAIN, HiFI performs favorably against state of the art on Xiph and X-TEST datasets, both in 2K and 4K resolutions. Pre-training on a large video dataset significantly boosts the performance of HiFI on Xiph and X-TEST, setting a new state of the art. Visually, HiFI can better interpolate fine details with large motion at high resolution, as shown in Fig. 5. We will analyze key components that contribute to the performance in the ablation study below.

SEPE. We also test HiFI on SEPE that includes 8K resolution videos. Most methods we tested ran out of memory except M2M (PSNR 28.34 (dB) and SSIM 0.883) and SGM-VFI (Liu et al. 2024a) (PSNR 28.43 (dB) and SSIM 0.880), compared with PSNR 29.78 (dB) and SSIM 0.900 by HiFI. Please view the supplementary for visual comparison.

Large Motion and Repetitive texture dataset

Public benchmark datasets, while diverse, do not fully capture the failure modes of current methods, especially large motion or repetitive texture cases common in real-world videos. To better evaluate existing methods and further innovation, we introduce a Large Motion and Repetitive texture (**LaMoR**) dataset that includes such 19 challenging examples at 4K resolution in both portrait and landscape modes,

Table 1: **Results on public benchmark datasets**: HiFI performs favorably on the highly-saturated Vimeo-90K (Xue et al. 2019) and is substantially more accurate than existing two-frame methods on high-resolution Xiph (Niklaus and Liu 2020) and X-TEST (Sim, Oh, and Kim 2021) datasets. **Best** and second-best are highlighted in color.

Method	Training dataset	Vimeo-90K		Xiph 2K		Xiph 4K		X-TEST 2K		X-TEST 4K	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
M2M (Hu et al. 2022)	Vimeo	35.47	0.978	36.44	0.967	33.92	0.945	32.07	0.923	30.81	0.912
FILM (Reda et al. 2022)	Vimeo	36.06	0.970	36.66	0.951	33.78	0.906	31.61	0.916	26.98	0.839
AMT (Li et al. 2023)	Vimeo	36.53	**0.982**	36.38	0.941	34.63	0.904	-	-	-	-
UPR-Net (Jin et al. 2023)	Vimeo	36.42	**0.982**	-	-	-	-	-	-	30.68	0.909
FITUG (Plack et al. 2023)	Vimeo	36.34	0.981	-	-	-	-	-	-	-	-
TCL (Zhou et al. 2023)	Vimeo	**36.85**	**0.982**	-	-	-	-	-	-	-	-
IQ-VFI (Hu et al. 2024)	Vimeo	36.60	**0.982**	36.68	0.942	34.72	0.905	-	-	-	-
EMA-VFI (Zhang et al. 2023)	Vimeo (+septuplet for X-TEST)	36.64	**0.982**	36.90	0.945	34.67	0.907	32.85	0.930	31.46	0.916
XVFI (Sim, Oh, and Kim 2021)	Vimeo / X-TRAIN	35.07	0.976	-	-	-	-	30.85	0.913	30.12	0.870
BiFormer (Park, Kim, and Kim 2023)	Vimeo + X-TRAIN	-	-	-	-	34.48	0.927	-	-	31.32	0.921
HiFI (Ours)	Vimeo + X-TRAIN	35.70	0.979	36.64	0.967	34.45	0.948	33.03	0.927	32.03	0.918
	Vimeo + X-TRAIN + Raw videos	36.12	0.980	**37.36**	**0.969**	**35.40**	**0.953**	**33.94**	**0.941**	**32.92**	**0.931**

Figure 6: **A few examples from our LaMoR dataset** that includes challenging scenes, such as repetitive texture and large motion where typical methods fail.

Table 2: Results on **LaMoR.** HiFI is significantly more accurate than state-of-the-art methods.

Method	PSNR	SSIM
LDMVFI (Danier, Zhang, and Bull 2024)	21.952	0.828
EMA-VFI (Zhang et al. 2023)	22.327	0.845
M2M (Hu et al. 2022)	24.995	0.884
SGM-VFI (Liu et al. 2024a)	25.122	0.894
UPR-Net (Jin et al. 2023)	25.856	0.892
BiFormer (Park, Kim, and Kim 2023)	26.330	0.893
HiFI (Ours)	**28.141**	**0.912**

as shown in Fig. 6. As in Table 2 and Fig. 7, HiFI substantially outperform all state of the arts on challenging cases of repetitive textures and large motion.

Ablation study

Dedicated upsample model vs single model. In Table 3, we compare the accuracy of the base model, two distinct models for base and upsample, and our final setting of using the same model for base and upsample. Both cascade strategies are effective for handling large motion, substantially improving accuracy on X-TEST. Using the same model for both base and upsample performs on-par or even better than having a dedicated upsample model, especially on challenging X-TEST. This validates the strength of re-using the same model for both over the more expensive dedicate model setup. Increasing the number of upsample stages improves the accuracy but saturates over three.

Table 3: **Base *vs*. cascade models.** Our patch-based self-cascade formulation substantially outperforms the base with the same number of parameters. Through model sharing, our self-cascade generalizes better on the X-TEST dataset than the standard cascade but with half of the parameters.

Method	Model size	Vimeo-90K		X-TEST 2K		X-TEST 4K	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Base	647 M	35.44	0.978	30.32	0.879	28.57	0.876
Standard two-stage cascade	1294 M	**36.12**	**0.980**	33.86	0.939	32.48	0.926
Patch-based self-cascade × 2	647 M	**36.12**	**0.980**	33.93	**0.941**	32.77	0.930
Patch-based self-cascade × 3	647 M	**36.12**	**0.980**	**33.94**	**0.941**	**32.92**	**0.931**

Comparison with coarse-to-fine refinement. Coarse-to-fine tiling refinement from DDVM (Saxena et al. 2023) first predicts the target at low resolution, bilinearly upsamples it to the target resolution, and refines it from an intermediate sampling step in a patch-wise manner. Our patch-based cascade performs consistently better than the coarse-to-fine tiling refinement on the X-TEST benchmark; 32.92 (dB) vs 32.54 (dB) on 4K, and 33.94 (dB) *vs*. 33.03 (dB) on 2K.

Architecture. In Table 4, we analyze where the major gain originates from by ablating attention layers or diffusion process in the base model, given the same training assets (*e.g.*, datasets, computations, *etc.*). Using attention layers brings about moderate performance gains on both the small (Vimeo) and large (X-TEST) motion datasets. We find the attention layers help with handling large motion and repetitive textures, enabling the accurate interpolation of frames by capturing the global context of these textures. Remov-

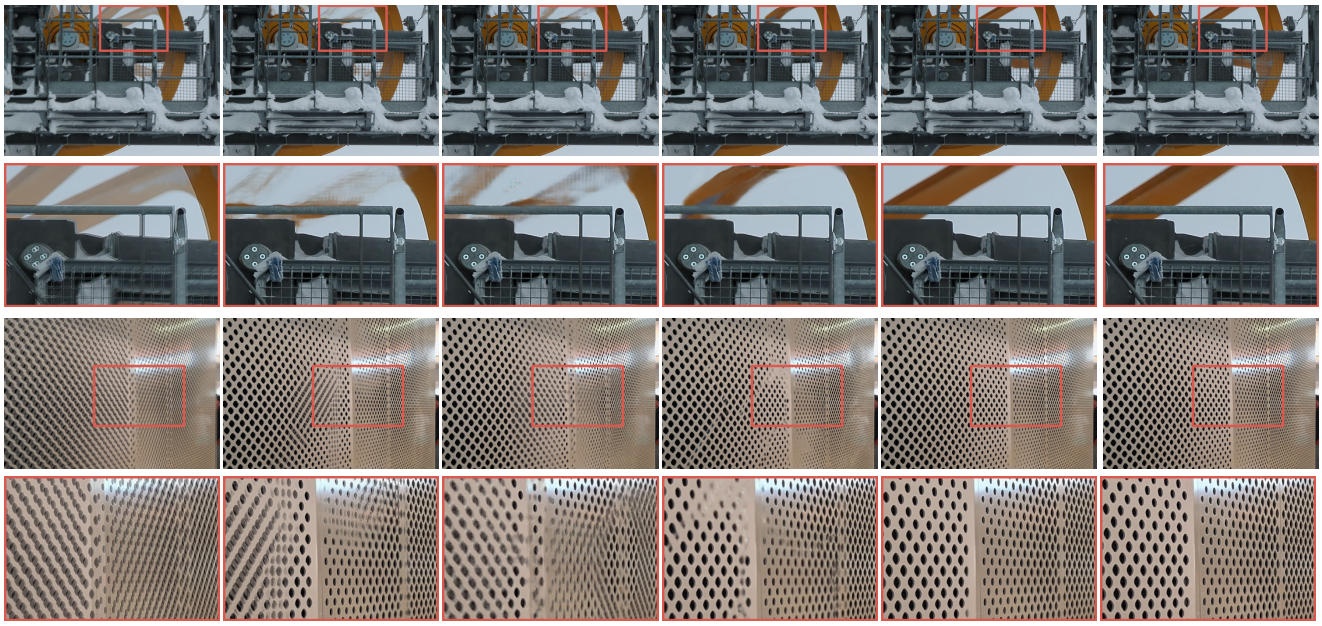


Figure 7: **Qualitative comparison on LaMoR.** The proposed HiFI is particularly effective at very challenging cases including repetitive textures and large motion.

Table 4: **Architecture analysis.** Both attention layers and diffusion process contribute to substantial accuracy gain. Comparing to a domain-specific architecture, FILM, our approach scales up better when training on the same large-scale video dataset.

Method	Vimeo-90K		X-TEST 2K		X-TEST 4K	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours, base model	**35.44**	**0.978**	**30.32**	**0.879**	**28.57**	**0.876**
w/o attention layers	35.13	0.977	29.73	0.861	27.75	0.854
w/o diffusion	33.78	0.965	28.05	0.852	27.56	0.861
FILM (Reda et al. 2022)	34.02	0.970	28.15	0.854	27.24	0.856

Table 5: **Effect of the sampling steps** on PSNR for the base model and patch-based cascade model. More steps tends to be better for higher resolution and large motion datasets.

(a) Base model

Steps	Vimeo-90K	X-TEST 4K
1	34.87	27.95
2	35.37	27.92
4	**35.44**	28.57
8	35.21	29.67
16	34.58	30.34
32	33.53	**30.40**
64	32.68	30.02

(b) Patch-based cascade

Steps	Vimeo-90K	X-TEST 4K
1	36.13	32.32
2	**36.15**	32.83
4	36.12	**32.92**
8	36.06	**32.92**
16	35.98	32.84
32	35.92	32.68
64	35.86	32.64

ing the diffusion process also leads to significant performance degradation. We also test one widely-used traditional method FILM (Reda et al. 2022), which relies on a "scale-agnostic" motion estimator to handle large motion. FILM trained on the same large dataset is substantially worse than HiFI, suggesting that traditional, hand-designed methods do not scale up well *w.r.t.* data.

Number of sampling steps. The optimal number of sampling steps also differ between the base and the patch-based cascade model. In general, we find that the model needs more sampling steps for large motion (*e.g.*, X-TEST) than for small motion (*e.g.*, Vimeo-90K (Xue et al. 2019)). However, the patch-based cascade model is able to achieve better numbers across different datasets with fewer sampling steps.

Discussions. Despite the performance gains on standard benchmarks, some extremely complicated motion types, *e.g.*, fluid dynamics, are still challenging for HiFI. Furthermore, diffusion models are computationally heavy and need distillation (Salimans and Ho 2022) for applications with a limited computational budget.

Conclusion

We have introduced a diffusion-based method for high resolution frame interpolation, named HiFI. Our proposed patch-based cascade achieves state-of-the-art performance on several high-resolution frame interpolation benchmarks up to 8K resolution, while improving efficiency for training and inference. We also establish a new benchmark, LaMoR, which focuses on challenging cases, *e.g.* large motion and repeated textures at high resolution. Our method substantially outperforms all methods on the benchmark as well.

Acknowledgement. We thank Tianhao Zhang, Yisha Sun, Tristan Greszko, Christopher Farro, Fuhao Shi for their help in collecting the dataset, and Yifan Zhuang, Ming-Hsuan Yang, David Salesin, and the anonymous reviewers for their constructive feedback and discussions.

References

Al Shoura, T.; Dehaghi, A. M.; Razavi, R.; Far, B.; and Moshirpour, M. 2023. SEPE Dataset: 8K Video Sequences and Images for Analysis and Development. In *Conference on ACM Multimedia Systems*, 463–468.

Bar-Tal, O.; Yariv, L.; Lipman, Y.; and Dekel, T. 2023. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *ICML*.

Baranchuk, D.; Voynov, A.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2022. Label-Efficient Semantic Segmentation with Diffusion Models. In *ICLR*.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; Jampani, V.; and Rombach, R. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv:2311.15127 [cs.CV]*.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 18392–18402.

Cheng, X.; and Chen, Z. 2020. Video frame interpolation via deformable separable convolution. In *AAAI*, 10607–10614.

Choi, M.; Kim, H.; Han, B.; Xu, N.; and Lee, K. M. 2020. Channel attention is all you need for video frame interpolation. In *AAAI*, 10663–10671.

Danier, D.; Zhang, F.; and Bull, D. 2024. LDMVFI: Video frame interpolation with latent diffusion models. In *AAAI*, 1472–1480.

Ding, Z.; Zhang, M.; Wu, J.; and Tu, Z. 2024. Patched denoising diffusion models for high-resolution image synthesis. In *ICLR*.

Ge, S.; Nah, S.; Liu, G.; Poon, T.; Tao, A.; Catanzaro, B.; Jacobs, D.; Huang, J.-B.; Liu, M.-Y.; and Balaji, Y. 2023. Preserve your own correlation: A noise prior for video diffusion models. In *ICCV*, 22930–22941.

Ho, J.; Saharia, C.; Chan, W.; Fleet, D. J.; Norouzi, M.; and Salimans, T. 2022a. Cascaded diffusion models for high fidelity image generation. *JMLR*, 23(47): 1–33.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022b. Video diffusion models. *NeurIPS*, 35: 8633–8646.

Hu, M.; Jiang, K.; Zhong, Z.; Wang, Z.; and Zheng, Y. 2024. IQ-VFI: Implicit Quadratic Motion Estimation for Video Frame Interpolation. In *CVPR*, 6410–6419.

Hu, P.; Niklaus, S.; Sclaroff, S.; and Saenko, K. 2022. Many-to-many splatting for efficient video frame interpolation. In *CVPR*, 3553–3562.

Huang, Z.; Zhang, T.; Heng, W.; Shi, B.; and Zhou, S. 2022. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 624–642.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*.

Jain, S.; Watson, D.; Tabellion, E.; Hołyński, A.; Poole, B.; and Kontkanen, J. 2024. Video Interpolation with Diffusion Models. In *CVPR*.

Jiang, H.; Sun, D.; Jampani, V.; Yang, M.-H.; Learned-Miller, E.; and Kautz, J. 2018. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 9000–9008.

Jin, X.; Wu, L.; Chen, J.; Chen, Y.; Koo, J.; and Hahm, C.-h. 2023. A unified pyramid recurrent network for video frame interpolation. In *CVPR*, 1578–1587.

Kalluri, T.; Pathak, D.; Chandraker, M.; and Tran, D. 2023. FLAVR: Flow-agnostic video representations for fast frame interpolation. In *WACV*, 2071–2082.

Ke, B.; Obukhov, A.; Huang, S.; Metzger, N.; Daudt, R. C.; and Schindler, K. 2024. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*.

Kiefhaber, S.; Niklaus, S.; Liu, F.; and Schaub-Meyer, S. 2024. Benchmarking Video Frame Interpolation. *arXiv:2403.17128 [cs.CV]*.

Kim, Y.; Hwang, G.; and Park, E. 2024. DiffuseHigh: Training-free Progressive High-Resolution Image Synthesis through Structure Guidance. *arXiv preprint arXiv:2406.18459*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Lee, H.; Kim, T.; Chung, T.-y.; Pak, D.; Ban, Y.; and Lee, S. 2020. AdaCof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, 5316–5325.

Li, Z.; Zhu, Z.-L.; Han, L.-H.; Hou, Q.; Guo, C.-L.; and Cheng, M.-M. 2023. AMT: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, 9801–9810.

Liu, C.; Zhang, G.; Zhao, R.; and Wang, L. 2024a. Sparse Global Matching for Video Frame Interpolation with Large Motion. In *CVPR*, 19125–19134.

Liu, Y.; Lin, C.; Zeng, Z.; Long, X.; Liu, L.; Komura, T.; and Wang, W. 2024b. SyncDreamer: Generating multiview-consistent images from a single-view image. In *ICLR*.

Meyer, S.; Djelouah, A.; McWilliams, B.; Sorkine-Hornung, A.; Gross, M.; and Schroers, C. 2018. PhaseNet for video frame interpolation. In *CVPR*, 498–507.

Meyer, S.; Wang, O.; Zimmer, H.; Grosse, M.; and Sorkine-Hornung, A. 2015. Phase-based frame interpolation for video. In *CVPR*, 1410–1418.

Nam, J.; Lee, G.; Kim, S.; Kim, H.; Cho, H.; Kim, S.; and Kim, S. 2024. Diffusion Model for Dense Matching. In *ICLR*.

Niklaus, S.; and Liu, F. 2018. Context-aware synthesis for video frame interpolation. In *CVPR*, 1701–1710.

Niklaus, S.; and Liu, F. 2020. Softmax splatting for video frame interpolation. In *CVPR*, 5437–5446.

Niklaus, S.; Mai, L.; and Liu, F. 2017a. Video frame interpolation via adaptive convolution. In *CVPR*, 670–679.

Niklaus, S.; Mai, L.; and Liu, F. 2017b. Video frame interpolation via adaptive separable convolution. In *ICCV*, 261–270.

Niklaus, S.; Mai, L.; and Wang, O. 2021. Revisiting adaptive convolutions for video frame interpolation. In *WACV*, 1099–1109.

Park, J.; Kim, J.; and Kim, C.-S. 2023. BiFormer: Learning bilateral motion estimation via bilateral transformer for 4K video frame interpolation. In *CVPR*, 1568–1577.

Park, J.; Ko, K.; Lee, C.; and Kim, C.-S. 2020. BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation. In *ECCV*, 109–125.

Park, J.; Lee, C.; and Kim, C.-S. 2021. Asymmetric bilateral motion estimation for video frame interpolation. In *ICCV*, 14539–14548.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *ICCV*, 4195–4205.

Plack, M.; Briedis, K. M.; Djelouah, A.; Hullin, M. B.; Gross, M.; and Schroers, C. 2023. Frame Interpolation Transformer and Uncertainty Guidance. In *CVPR*, 9811–9821.

Qian, G.; Mai, J.; Hamdi, A.; Ren, J.; Siarohin, A.; Li, B.; Lee, H.-Y.; Skorokhodov, I.; Wonka, P.; Tulyakov, S.; and Ghanem, B. 2024. Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors. In *ICLR*.

Ranjan, A.; and Black, M. J. 2017. Optical Flow Estimation using a Spatial Pyramid Network. In *CVPR*.

Reda, F.; Kontkanen, J.; Tabellion, E.; Sun, D.; Pantofaru, C.; and Curless, B. 2022. FILM: Frame interpolation for large motion. In *ECCV*, 250–266.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 36479–36494.

Salimans, T.; and Ho, J. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *ICLR*.

Saxena, S.; Herrmann, C.; Hur, J.; Kar, A.; Norouzi, M.; Sun, D.; and Fleet, D. J. 2023. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *NeurIPS*.

Saxena, S.; Hur, J.; Herrmann, C.; Sun, D.; and Fleet, D. J. 2024. Zero-Shot Metric Depth with a Field-of-View Conditioned Diffusion Model. In *ECCVW*.

Shi, S.; Li, W.; Zhang, Y.; He, J.; Gong, B.; and Zheng, Y. 2024a. ResMaster: Mastering High-Resolution Image Generation via Structural and Fine-Grained Guidance. *arXiv:2406.16476 [cs.CV]*.

Shi, Y.; Wang, P.; Ye, J.; Mai, L.; Li, K.; and Yang, X. 2024b. MVDream: Multi-view Diffusion for 3D Generation. In *ICLR*.

Shi, Z.; Xu, X.; Liu, X.; Chen, J.; and Yang, M.-H. 2022. Video frame interpolation transformer. In *CVPR*, 17482–17491.

Sim, H.; Oh, J.; and Kim, M. 2021. XVFI: extreme video frame interpolation. In *ICCV*, 14489–14498.

Skorokhodov, I.; Menapace, W.; Siarohin, A.; and Tulyakov, S. 2024. Hierarchical Patch Diffusion Models for High-Resolution Video Generation. In *CVPR*, 7569–7579.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 8934–8943.

Teed, Z.; and Deng, J. 2020. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 402–419.

Xu, J.; Liu, S.; Vahdat, A.; Byeon, W.; Wang, X.; and De Mello, S. 2023. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2955–2966.

Xue, T.; Chen, B.; Wu, J.; Wei, D.; and Freeman, W. T. 2019. Video enhancement with task-oriented flow. *IJCV*, 127: 1106–1125.

Yang, S.; Hwang, H.; and Ye, J. C. 2023. Zero-shot contrastive loss for text-guided diffusion image style transfer. In *ICCV*, 22873–22882.

Zhang, G.; Zhu, Y.; Wang, H.; Chen, Y.; Wu, G.; and Wang, L. 2023. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, 5682–5692.

Zheng, Q.; Guo, Y.; Deng, J.; Han, J.; Li, Y.; Xu, S.; and Xu, H. 2024. Any-size-diffusion: Toward efficient text-driven synthesis for any-size HD images. In *AAAI*, 7571–7578.

Zhou, K.; Li, W.; Han, X.; and Lu, J. 2023. Exploring motion ambiguity and alignment for high-quality video frame interpolation. In *CVPR*, 22169–22179.

Supplementary material

Overview

Here, we provides further implementation details, analyses on our design choices, further discussions on results on Vimeo-90K and SEPE 8K benchmark datasets, and analyses on computational complexity. We also provide more qualitative comparison including interactive tools and videos with state-of-the-art methods in our website. For more details, please browse our project webpage: ***https://hifi-diffusion.github.io***

Implementation details

We include further implementation details continuing from the main paper. Our implementation is based on JAX framework. We use a fixed random seed for reproducibility. We use 256 TPUv5e with 16 GB memory for training. For inference, our model runs on one A100 40GB and processes up to 8K resolution without a memory problem. Our novel HiFI cascade enables this high-resolution processing where most of the methods have difficulties in.

Effect of the patch size and overlap

We provide an analysis on how patch size in the cascade model affects the accuracy during inference. Due to the self-attention layers at two bottom levels, the performance of our method could vary, when a patch size that is different from the training resolution (*i.e.*, 224×288) is used. Also, bigger patch sizes can give better accuracy due to a larger context window, but it is not so clear if it always holds. Given our standard setup (*i.e.*, a three-stage cascade, 4 sampling steps, an average of 4 samples), we try different patch size and evaluate on X-TEST 4K dataset (Sim, Oh, and Kim 2021).

Table 6 reports PSNR and SSIM on X-TEST 4K dataset. Although the training resolution is at 224×288, the method is not very sensitive to the choice of patch size at inference time. The smallest and the biggest patch size (*i.e.*, 256×384 and 768×1152) show marginal difference in both PSNR and SSIM metrics. The patch size 512×768 gives the best accuracy on X-TEST 4K.

We also analyze the impact of varying patch overlap on X-TEST 4K, using a patch size of 512×768. Increase of overlap size between patches can have a similar effect of averaging more samples. By default at inference, we place patches to cover the entire image with minimal, equally distributed overlap, which is automatically determined. In this study, we gradually increase the number of patches at each row or column with equal distanced, compute an overlap ratio, and also report PSNR X-TEST 4K benchmark. The overlap ratio is the value obtained by dividing the sum of all areas processed by the patches by the total image size, computed by $\frac{\text{patch size} \times \text{the number of patches}}{\text{image size}}$.

As in Table 7, overlap size does not significantly affect the performance, showing standard deviation of 0.052 for PSNR and 0.00059 for SSIM. More overlap marginally improves the performance by averaging multiple samples but with the cost of runtime increase. Overlap ratio with 1.33 is our default setup.

Table 6: **Effect of different patch size:** The usage of different patch sizes does not show significant accuracy difference on X-TEST 4K dataset.

Patch size	PSNR	SSIM
256×384	32.70	0.931
384×576	32.82	**0.932**
512×768	**32.92**	0.931
640×960	32.79	0.930
768×1152	32.75	0.930

Table 7: **Effect of overlapping ratio between patches:** Overlap size marginally affect the accuracy. The overlap ratio with 1.33 is our default setup.

Overlap ratio	1.33	1.56	1.60	1.78	1.87	2.13	2.18	2.49
PSNR	32.92	32.99	33.00	33.02	32.98	33.10	33.03	33.07
SSIM	0.931	0.932	0.932	0.932	0.932	0.933	0.932	0.933

Overlaid inputs Intermediate W/o dropout W/ dropout

Figure 8: **Dropout** forces the network to use both the low resolution intermediate and the inputs. Without dropout, the network takes a shortcut and only tries to upsamples the intermediate with missing details. The second row shows close views of highlighted areas in the images at the first row.

Dropout for patch-based cascade training

We find that the image-level dropout is crucial for making the patch-based cascade model behave as intended, especially for challenging large motion scenes, as in Fig. 8. Without dropout, the model finds a shortcut, sharpening the low resolution intermediate from the base model without referring to input images. This results in losing fine details, *e.g.*, thin structures and letters. With dropout, the model refers to both the low resolution intermediate for coarse structure and the high-resolution input for fine details. In this study, we train the model on X-TRAIN (Sim, Oh, and Kim 2021) only, as the network without dropout does not converge with a full training dataset. This suggests that the dropout also stabilizes large scale training for the cascade model.

To further analyze the effect of the image-level dropout, we prepare two models that are with or without the dropout and see the models' behavior by inputting a different image as the low resolution intermediate during the inference. We test a two-stage cascade model as in Fig. 9c.

The model with dropout (*i.e.*, Fig. 9d) successfully outputs high-resolution prediction when actual low resolution intermediates are inputted. When a different intermediate is inputted, the model tries to add appearance (*e.g.*, color or texture) from input frames on top of object structures from the low resolution intermediate. Though it produces non-

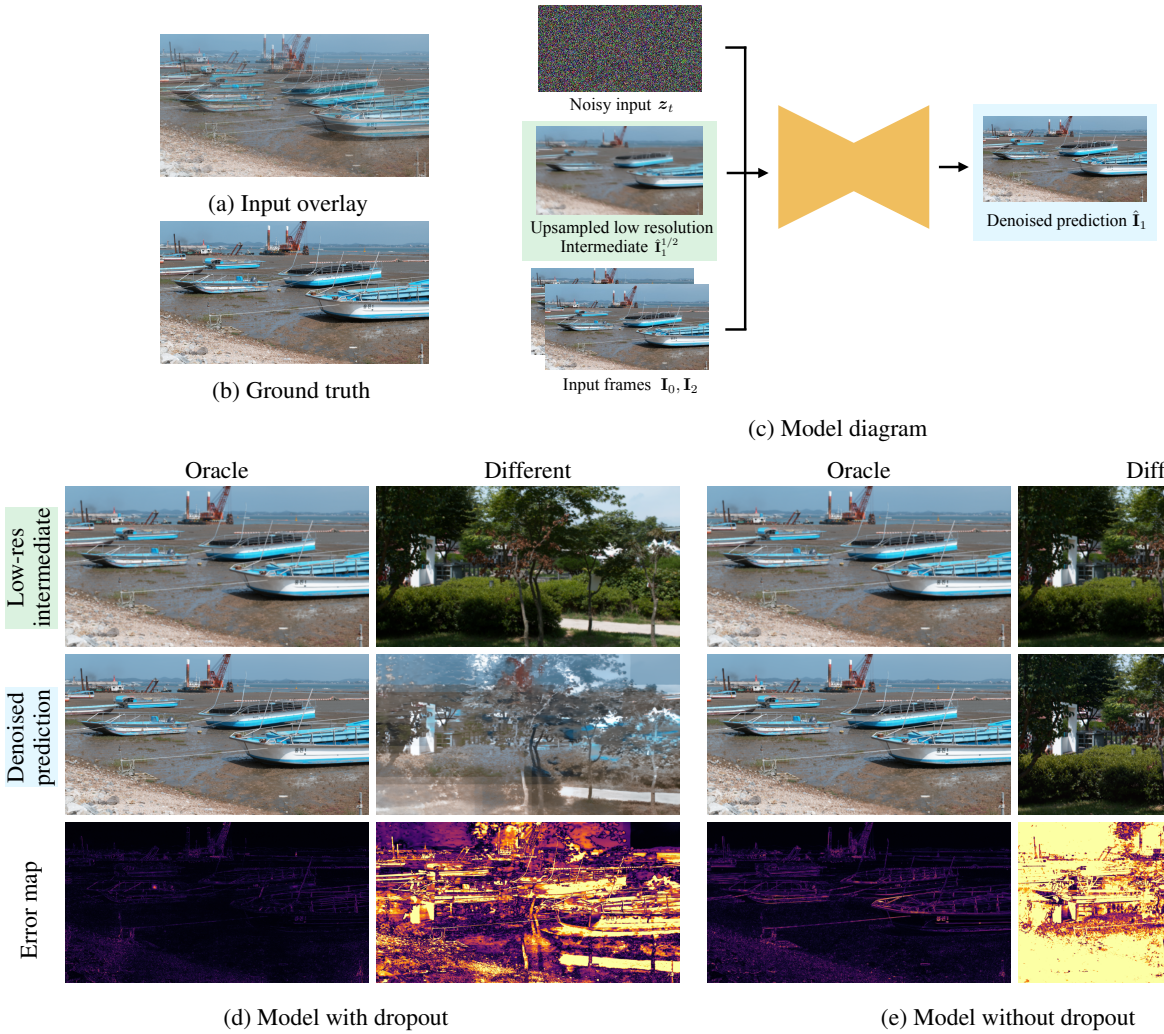


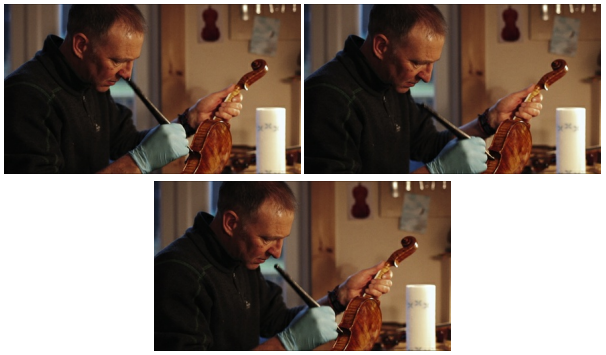
Figure 9: **Effect of dropout**. In the two-stage cascade formulation, we train our cascade model with or without dropout and visualize results when inputting oracle/different low resolution intermediate respectively. These inputs are downsampled and upsampled back to the original resolution to mimic the low resolution intermediate from the previous level. *(d)* With dropout the model effectively utilizes coarse structure from the intermediate and fine details from the high resolution input. This holds true even with different low resolution intermediate: the model add color and texture to the coarse structure. *(e)* On the other hand, the model without dropout solely relies on the intermediate, primarily sharpening it. This leads to a loss of fine details, *e.g.* around object boundaries (see the error map). The behavior becomes looking more apparent with a different intermediate; the model ignores input frames.

sensible output, this proves that the model is able to exploit both input frame and low resolution intermediate during the inference.

On the other hand, the model trained without dropout (*i.e.*, Fig. 9e) only refers to the low resolution intermediate. Even when inputting a different image as the low resolution condition (*e.g.*, the right column in Fig. 9e), the denoised prediction completely ignores the input frames and takes a shortcut to sharpen the low resolution condition (*i.e.*, tree image). Our probabilistic image-level dropout prevents the model from taking this shortcut and learns to refer to both input and condition cues.

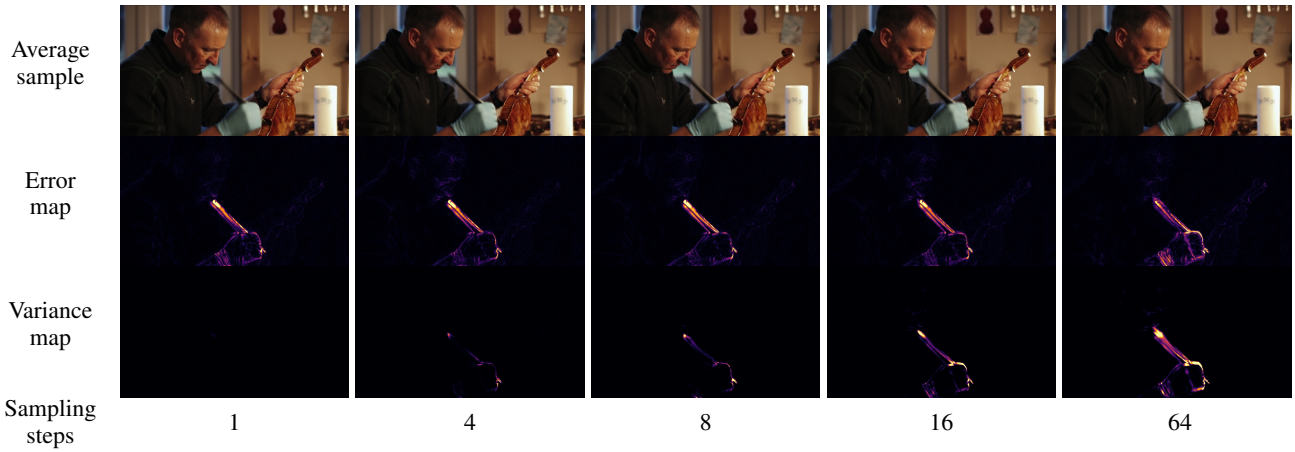
Effect of the number of sampling steps

In Fig. 10, we visualize how the number of sampling steps affects results. The input frame in Fig. 10a shows a person using a stick, and the stick moves fast between the frames. The variance map in Fig. 10c shows that with more sampling steps, the model outputs more diverse motion of the stick and the hand of the person, as highlighted in the various map. With the lower number of sampling steps, our model produces close-to-mean prediction. Fig. 10b visualizes the four samples drawn from 64 sampling steps; our model predicts diverse, plausible samples with non-linear motion, such as in different trajectories or at different acceleration and deceleration rate. This follows the same ob-



(a) Two input frames (above) and ground truth (below)

(b) Multiple samples with 64 sampling steps



(c) Average sample, error map, and variance map *w.r.t.* the different number of sampling steps.

Figure 10: **Effect of the number of sampling steps**: We visualize how the number of sampling steps affects results. (a) Given two input frames, we try different sampling steps and visualize (b) each individual sample as well as (c) averaged sample, error map, and variance map. With more sampling steps, the model predicts multiple plausible diverse samples with non-linear motion (*i.e.*, the fast moving stick).

servation from DDVM (Saxena et al. 2023) that the diffusion model is able to predict plausible multi-mode samples.

Randomness and robustness

With the stochastic nature, HiFI predicts plausible diverse samples with non-linear motion as in Fig. 10, as a unique capability. To see if this unique property can also affect accuracy, we tested 10 runs with different random seeds and compute mean and standard deviation of results on four difference benchmark datasets. As reported in Table 8, it does not yield much variation on accuracy.

Table 8: **Randomness and robustness**: the unique tochastic nature from diffusion does not yield much variation on accuracy.

PSNR	Vimeo	Xiph 2K	X-TEST 4K	LaMoR
mean	36.12	37.34	32.92	28.15
std. dev.	0.0052	0.0024	0.0196	0.0233

Evaluation on SEPE 8K benchmark

SEPE 8K dataset (Al Shoura et al. 2023) provides 40 raw videos with 300 frames, 8K resolution, and 29.97 FPS for benchmarking various downstream computer vision tasks such as video quality assessment, super-resolution, compression, *etc*. To utilize the dataset for benchmarking frame interpolation methods, especially for high resolution with large motion, we select a triple of frames (145th, 150th, and 155th) from each video and target to predict the middle frame from the rest two frames as input.

Table 9 includes all state-of-the-art methods that we compare on SEPE 8K benchmark. Except M2M (Hu et al. 2022) and SGM-VFI (Liu et al. 2024a), the other methods are not able to process 8K resolution image due to the out-of-memory (OOM) problem on A100 40GB GPU. Figure 11 provide qualitative comparison between our method and M2M (Hu et al. 2022). Unlike M2M (Hu et al. 2022), our method is able to recover fine details on such challenging

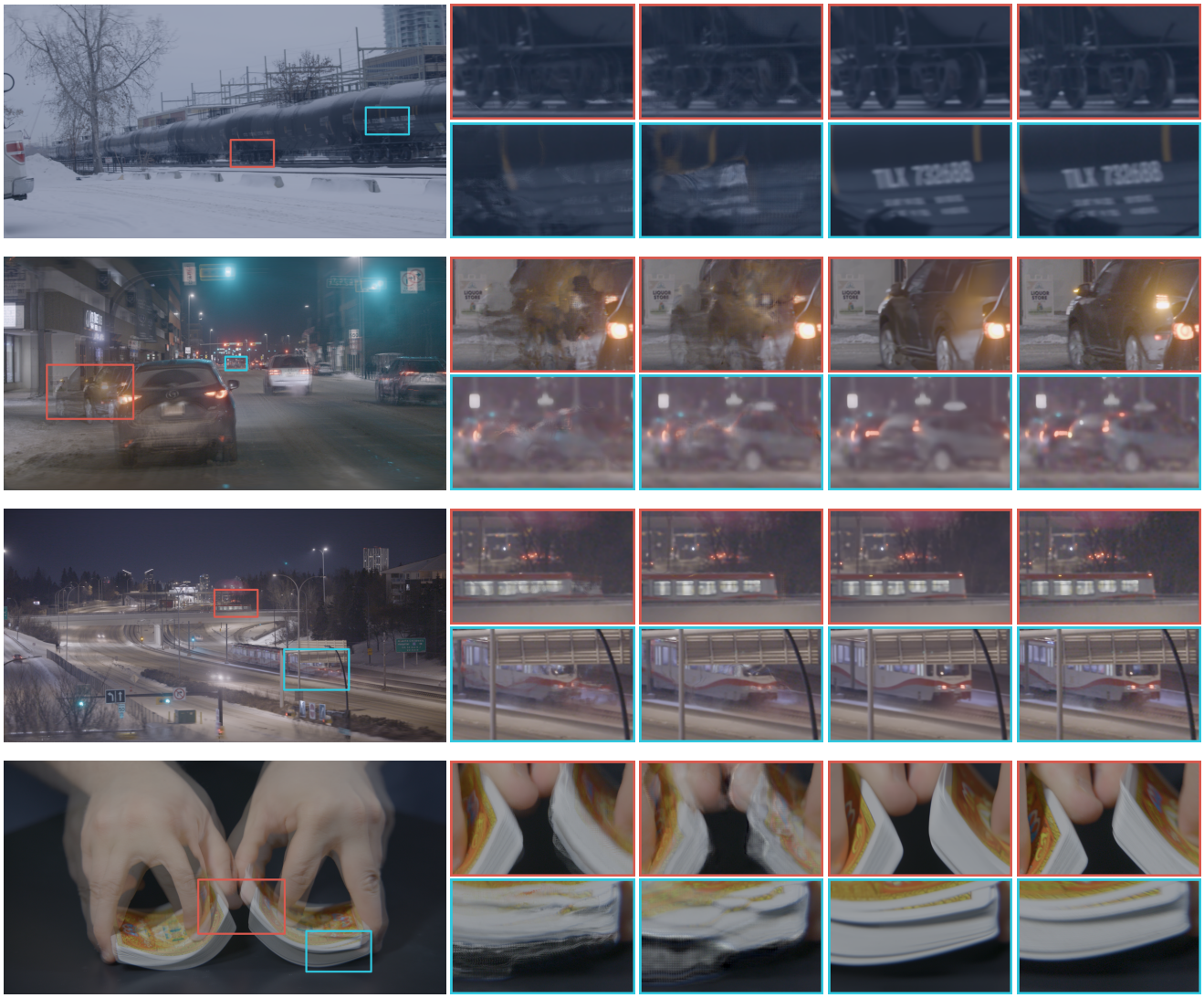


Figure 11: **Qualitative comparison on SEPE 8K**: We provide qualitative comparison between our method and M2M (Hu et al. 2022) which is able to run at 8K resolution. Compared to M2M (Hu et al. 2022) and SGM-VFI (Liu et al. 2024a) which have a difficulty in handling large motion, our method is able to recover fine details on challenging cases at 8K resolution.

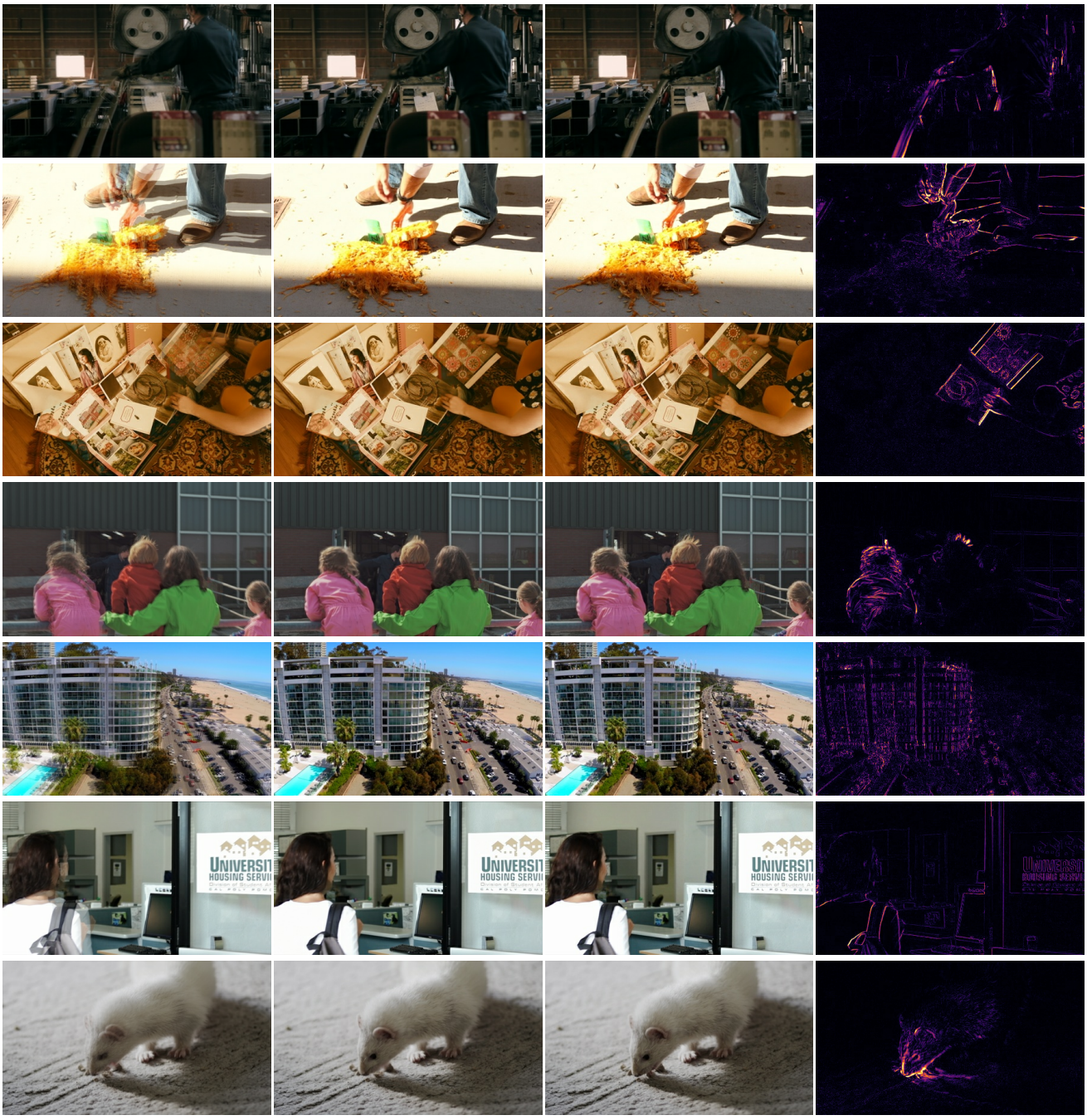
Table 9: **Results on SEPE 8K dataset.** HiFI outperforms M2M (Hu et al. 2022). Most of the methods have the OOM (out of memory) problem at 8K resolution.

Method	PSNR	SSIM
LDMVFI (Danier, Zhang, and Bull 2024)	OOM	
EMA-VFI (Zhang et al. 2023)	OOM	
UPR-Net (Jin et al. 2023)	OOM	
BiFormer (Park, Kim, and Kim 2023)	OOM	
M2M (Hu et al. 2022)	28.34	0.883
SGM-VFI (Liu et al. 2024a)	28.43	0.880
HiFI (Ours)	**29.78**	**0.900**

large motion cases even at 8K resolution.

Discussion on Vimeo-90K

While HiFI achieves the best accuracy on multiple high resolution benchmark datasets, it performs comparably on highly-saturated Vimeo-90K benchmark (Xue et al. 2019). To analyze the behavior, in Fig. 12 we show random-sampled results of our method that gives high errors on Vimeo-90K. As visualized in the error map, erroneous prediction mostly arises from motion boundaries even if HiFI is able to interpolate frames with fine details, as shown in the second column. This is specifically due to the dataset property: Vimeo-90K prefers linear motion prediction (Kiefhaber et al. 2024) whereas our model predicts diverse non-linear motion of objects. Non-linear motion prediction yields a subtle misalignment between predicted posi-



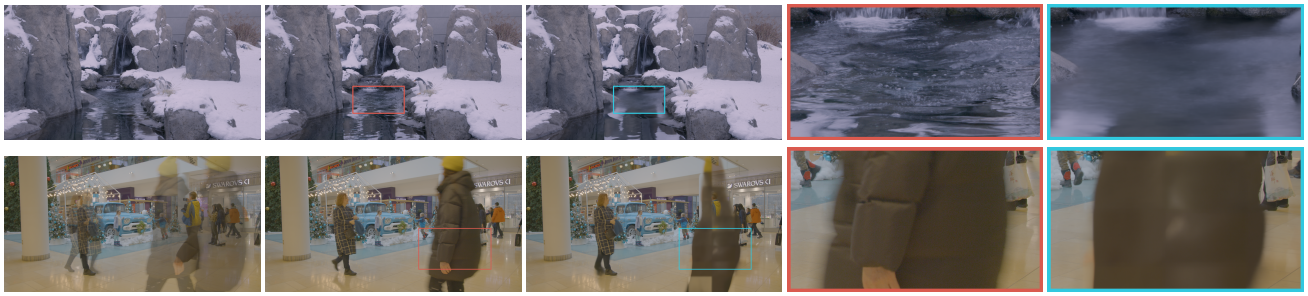
Input overlay | Ours | Ground truth | Error map

Figure 12: **Error map visualization on Vimeo-90K**: We randomly sample ours results with high errors on Vimeo-90k and visualize them with input overlay, ground truth, and error map. The error mostly originates from motion boundaries where the predicted objects' motion sometimes do not align well with true motion. This is because Vimeo-90K dataset prefers linear motion prediction whereas our model can predict plausible non-linear motion (Kiefhaber et al. 2024).

tion and true position of object, and it causes intensity difference mostly near image edges. Thus errors mostly arise near object or motion boundaries. In Fig. 10b, we visualize multiple non-linear motion examples that our model produces.

Comparison with diffusion-based approach

Between two existing diffusion-based methods, VIDIM (Jain et al. 2024) and LDMVFI (Danier, Zhang, and Bull 2024), we provide comparisons with LDMVFI (Danier, Zhang, and Bull 2024) in Table 10. VIDIM (Jain et al. 2024)



Input overlay | Ground truth | **HiFI (Ours)** | Ground truth close view | **HiFI (Ours)** close view

Figure 13: **Some extremely challenging cases** that even our model struggles with: *(above)* fluid motion where more generative solution can be preferred or *(below)* extremely large motion, around 1500 pixels in the example, that is much bigger than the patch size at inference time.

Table 10: **Comparison with diffusion-based approaches:** Our method consistently outperforms LDMVFI (Danier, Zhang, and Bull 2024) on both X-TEST 4K and LaMoR datasets by a large margin.

	X-TEST 4K			LaMoR		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**32.92**	**0.931**	**0.2136**	**28.141**	**0.912**	**0.1880**
LDMVFI	23.36	0.770	0.3285	21.952	0.828	0.2438

focuses on a task closer to conditional video generation, a long-range interpolation on 256 × 256 resolution, which is different from our target task.

Our method consistently outperforms LDMVFI (Danier, Zhang, and Bull 2024) by a large margin, even on LPIPS metric despite that LDMVFI (Danier, Zhang, and Bull 2024) uses LPIPS-based training loss. Furthermore, LDMVFI (Danier, Zhang, and Bull 2024) does not run on SEPE 8K due to the out of memory problem.

Computational complexity

Table 11 provides comparison on computational complexity and runtime performance among different methods. We test each method on X-TEST 4K benchmark and report its accuracy, inference time, and peak memory, using one A100 40GB GPU. Despite of its demanding model size and runtime, our method requires only 7.53 GB peak memory for 4K image processing and achieves the best accuracy compared with others.

We also provide analysis on the impact of cascade levels (Table 12) and patch size (Table 13) on accuracy and runtime (X-TEST 4K, A100 40GB). As in Table 12, the direct input of 4K images causes out-of-memory errors. Simple adoption of patch-based processing (*i.e.* (Patch-based) Base) reduces memory, though it struggles with large motion, resulting in lower PSNR and SSIM. Our patch-based cascades eventually handle challenging cases with improved accuracy, with peak-memory nearly unchanged. Table 13 provide an analysis that both peak memory and runtime increase *w.r.t.* the patch size mainly due to the self-attention layers in the bottleneck. We find that the patch size of 512 × 768 achieves the best performance without having too much increase of runtime and memory.

Table 11: **Comparison on computational complexity:** We measure computational complexity and accuracy of each model on X-TEST 4K benchmark. Our method requires only 7.53 GB peak memory for 4K image processing and achieves the best accuracy compared with others.

Method	Inference time (s)	Model parameters (M)	Peak memory (GB)	X-TEST 4K PSNR (dB)
UPR-Net	**1.96**	**6.56**	33.44	30.68
M2M	2.51	7.61	12.65	30.81
BiFormer	2.34	11.17	21.50	31.32
EMA-VFI	3.27	65.66	30.30	31.46
LDMVFI	11.63	416.46	37.23	23.36
HiFI (ours)	164.18	647.74	**7.53**	**32.92**

Table 12: **Efficiency analysis of cascade processing:** Our patch-based cascades handle challenging cases with improved accuracy, with peak-memory nearly unchanged.

Method	Peak memory (GB)	runtime (s)	PSNR	SSIM
Whole-image processing	OOM	–	–	–
(Patch-based) Base	7.47	94.62	28.57	0.876
Patch-based cascade × 2	7.52	136.53	32.77	0.930
Patch-based cascade × 3	7.53	164.18	32.92	0.931

We leave the reduce of computational cost as future work. Runtime can be substantially reduced by using fewer denoising steps. Having just two steps can yield up to 2 times speedup with only a 0.16% average accuracy drop across five benchmark datasets. Batch processing of samples and patches will further accelerate the runtime while maintaining the same accuracy. Model parameters can also be reduced via model distillation.

Table 13: **Efficiency analysis of patch-wise processing:** The usage of the patch size of 512 × 768 achieves the best performance with a reasonable increase of runtime and memory.

	Patch size	Peak memory (GB)	Runtime (s)	PNSR	SSIM
Patch-based	384×576	6.49	139.68	32.82	0.932
	512×768	7.53	164.18	32.92	0.931
	640×960	9.34	189.57	32.79	0.930
	768×1152	12.51	209.80	32.75	0.930
Whole image	2160×4096	OOM	–	–	–

Challenges and future work

Our method achieves the state of the art on multiple high-resolution benchmark datasets, yet there still exists some extremely difficult cases that challenge our method. Figure 13 shows a few examples from the SEPE 8K dataset. In case of fluid motion in the first row, HiFI tends to output blurry results; more generative solution can be preferred. Also as in the second row, when motion is extremely larger (*e.g*. 1500 pixels) than the patch size, most of the content goes out of image boundary. HiFI cannot establish reliable correspondence, and thus is not able to interpolate their motion. The usage of a bigger patch size can resolve the issues but with increased computational cost.

Furthermore, our model predicts one middle frame given two input frames as a basic setup, following the conventional setup. As future work, our model can be easily extended to the multi-frame setup by conditioning on multiple frames and interpolating multiple middle frames together, trained on raw videos. This multi-frame setup can also effectively represent non-linear motion.