RePix-VFI: Re-rendering Observed Pixels for Video Frame Interpolation with Generative Priors from One-Step Diffusion

Dayeon Kim
KAIST
dyk4501@kaist.ac.kr

Jeonghyeok Do†
KAIST
ehwjdgur0913@kaist.ac.kr

Munchurl Kim†
KAIST
mkimee@kaist.ac.kr

https://kaist-viclab.github.io/RePix-VFI_site

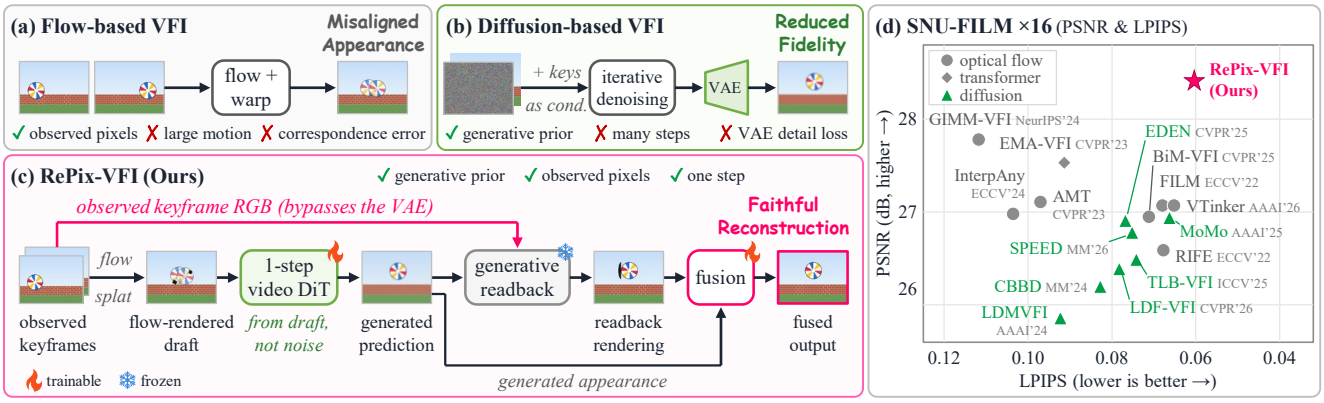

Figure 1. **RePix-VFI re-renders observed pixels using a one-step generated video as the correspondence reference.** (a) Flow-based VFI transports observed appearance but can misplace details under inaccurate correspondences. (b) Diffusion-based VFI draws on generative priors but typically requires iterative denoising and can lose observed details in VAE reconstruction. (c) Our RePix-VFI refines a flow-rendered draft in a single forward pass. The completed prediction provides correspondence references for re-rendering original keyframe RGB outside the VAE and supplies complementary appearance for fusion. (d) On SNU-FILM ×16, RePix-VFI achieves both the highest PSNR and the lowest LPIPS among the compared methods (upper right is better).

Abstract

*Multi-frame video frame interpolation (VFI) aims to reconstruct dense video from sparse keyframes, requiring accurate intermediate motion and faithful appearance. As temporal gaps grow, flow-based methods can retain observed textures but misplace them under ambiguous motion, whereas iterative latent-diffusion models can synthesize plausible content but incur substantial sampling costs and may alter details during reconstruction with a variational autoencoder (VAE). We present **RePix-VFI**, named for **Re**-rendering observed **Pix**els, a multi-frame VFI framework that couples draft-conditioned one-step video diffusion with reconstruction from observed keyframes. A pretrained video diffusion model refines a flow-rendered draft of the whole window in a single forward pass. For each target frame, our generative readback matches its two bounding keyframes to the generated prediction using an optical-flow estimator. The resulting correspondences guide re-rendering of original keyframe RGB pixels outside the VAE. A lightweight appearance fusion head combines these renderings with generated appearance. Experiments on benchmark datasets demonstrate significant improvements in reconstruction and perceptual quality. On SNU-FILM at ×8 and ×16, RePix-VFI achieves the highest PSNR and the lowest LPIPS, FloLPIPS, and tOF among the compared methods, exceeding the best evaluated diffusion-based baseline by 1.71 and 1.48 dB in PSNR, respectively. It also runs 32–42× faster than the state-of-the-art multi-frame VFI diffusion model.*

1. Introduction

Video frame interpolation (VFI) reconstructs the intermediate frames between observed keyframes, increasing the temporal resolution of a video. It is a fundamental low-level vision task, widely used for slow-motion synthesis [14], frame-rate up-conversion, and video compression [40], but

†Co-corresponding authors (equal advising).

remains challenging because motion and appearance at unobserved times must be inferred from the available frames. Multi-frame VFI goes beyond a single frame pair, reconstructing a dense video window jointly from several sparse keyframes. Accurate reconstruction requires placing content correctly at intermediate times while preserving observed appearance. As the temporal gap grows, large displacements, nonlinear motion, and occlusions make intermediate motion and visibility ambiguous.

To reconstruct intermediate frames while retaining observed appearance, flow-based VFI methods estimate intermediate motion and warp observed pixels or features accordingly [8, 11, 16, 23, 27, 29]. As illustrated in Fig. 1(a), this provides direct access to keyframe textures, but inaccurate correspondences can misplace detail and produce ghosting under large gaps. Moreover, pixel transport alone cannot reconstruct content absent from all observations.

Diffusion-based VFI offers complementary information through learned priors for intermediate appearance or motion [6, 9, 15]. These priors can support plausible predictions when motion is ambiguous or observations are incomplete. Video diffusion models extend these priors to multi-frame VFI, jointly predicting a whole window from multiple keyframes [25]. However, as shown in Fig. 1(b), iterative latent-diffusion models incur substantial sampling costs through repeated network evaluations, and reconstructing frames with a variational autoencoder (VAE) can alter details already available in the keyframes.

Prior methods use interpolated estimates to refine motion or match input textures, or derive continuous trajectories from generated latent anchors [21, 24, 39]. We instead use the RGB predictions from one-step video generation as references for matching the observed keyframes at each target time. This separates the reference used to estimate correspondences from the observed RGB transported by the renderer. Our controlled comparison shows that a generated prediction can provide a better rendering reference even when its RGB PSNR is lower than that of the flow-rendered draft.

Based on this observation, we present **RePix-VFI**, named for **Re**-rendering observed **Pix**els, a multi-frame VFI framework that couples draft-conditioned one-step video diffusion with reconstruction from observed keyframes (Fig. 1(c)). A flow-based renderer first constructs a dense draft from the observations, providing an initial estimate of intermediate motion and appearance. Starting from the encoded flow-rendered draft, a pretrained video diffusion transformer, fine-tuned as a one-step generator, predicts a latent correction in a single forward pass per window. For each intermediate frame, our *generative readback* uses a frozen optical-flow estimator to match its two bounding keyframes to the generated prediction. The resulting correspondences guide re-rendering of original keyframe RGB along a path that bypasses the VAE. A lightweight appearance fusion head combines these renderings with the generated predictions.

Experiments on SNU-FILM and DAVIS demonstrate the effectiveness of the complete reconstruction framework. Under the SNU-FILM entire-video protocol at $\times 8$ and $\times 16$, RePix-VFI achieves the highest PSNR and the lowest LPIPS, FloLPIPS, and tOF among the compared methods, exceeding the best evaluated diffusion-based baseline by 1.71 and 1.48 dB in PSNR, respectively. It also attains the highest PSNR and lowest LPIPS on DAVIS at $\times 4$–$\times 16$. Controlled comparisons support the design of generative readback. With the flow estimator and renderer fixed, replacing the flow-rendered draft with the generated prediction as the correspondence reference improves rendering PSNR by 0.45–0.72 dB. The full model also lowers LPIPS relative to Fusion w/o readback at all three factors, and fusion exceeds the better of the re-rendering and the generated prediction by 0.39–1.46 dB. Under the same 49-frame protocol, end-to-end inference is 32–42$\times$ faster than LDF-VFI [25], the state-of-the-art multi-frame VFI diffusion model.

Our contributions are summarized as follows:

- We propose **RePix-VFI**, a multi-frame VFI framework that adapts a pretrained video diffusion transformer for draft-conditioned one-step prediction and reuses its RGB outputs for correspondence estimation and appearance fusion.
- We introduce *generative readback*, which uses the generated predictions at intermediate times as correspondence references for a frozen optical-flow estimator. The resulting correspondences guide re-rendering of original keyframe RGB outside the VAE, followed by lightweight appearance fusion with the generated predictions.
- Experiments demonstrate superior reconstruction and perceptual performance, together with improved temporal consistency on SNU-FILM and DAVIS. Controlled ablations examine the roles of generated correspondence references and observation rendering in the complete reconstruction.

2. Related Work

2.1. Video Frame Interpolation (VFI)

Flow-based video frame interpolation. Flow-based VFI methods synthesize intermediate frames by warping or splatting observed pixels and features [14, 23]. Recent works improve motion estimation and reconstruction across scales and interpolation times [11, 27, 44]. GIMM-VFI [8] uses coordinate-based implicit motion modeling, while BiM-VFI [29] introduces bidirectional motion descriptors for non-uniform motion. Some methods match the inputs against intermediate estimates: ABME [24] refines motion using an interpolated anchor, while VTinker [39] maps input texture features onto a proxy derived from flow-warped inputs.

Diffusion-based video frame interpolation. Diffusion-

based VFI methods model intermediate appearance or motion using generative priors. LDMVFI [6] performs conditional latent diffusion with a decoder that predicts deformable kernels for sampling, whereas MoMo [15] and HFD [9] generate optical flows for subsequent frame synthesis. For multi-frame VFI, LDF-VFI [25] jointly synthesizes temporal chunks and connects them autoregressively, and conditions its VAE decoder on features from the observed video. SpeedVFI [7] distills multi-frame interpolation into one-step generation from noise latents conditioned on encoded keyframes, and SPEED [48] generates each intermediate frame with one-step diffusion in pixel space, avoiding latent encoding and decoding.

Generation and observation reuse. Existing methods connect motion estimation, generation, and observation reuse at different stages. MoG [45] injects backward-warped input latents and features to constrain generated motion. The training-free SNM-VFI [13] initializes and iteratively guides video diffusion with flow-derived drafts, then blends flow-based and generated frames with confidence from forward–backward flow consistency. Both use motion estimated from the observations to guide generation. Other methods exploit generated intermediate representations: MADiff [12] updates motion hints from intermediate predictions during iterative sampling and uses them for conditional generation and input-feature alignment within its decoder. RDVFI [21] generates low-resolution sparse latent keyframes with one-step diffusion, estimates continuous pixel trajectories, and synthesizes target frames from warped inputs and generated latent features.

Distinguished from previous methods, our RePix-VFI couples draft-conditioned one-step video diffusion with reconstruction from observed keyframes. A single generator evaluation refines the encoded flow-rendered draft into a dense RGB prediction. Whereas RDVFI [21] derives trajectories from sparse latent anchors, our *generative readback* matches observed keyframes directly to each generated target-time RGB prediction using a frozen optical-flow estimator. These correspondences transport original keyframe RGB outside the VAE, while the generated prediction remains an appearance candidate for fusion.

2.2. One-step Diffusion for Restoration

One-step restoration methods adapt pretrained diffusion priors for efficient image and video reconstruction [7, 21]. OSEDiff [41] starts from the encoded degraded image instead of noise and regularizes the output with variational score distillation (VSD). DOVE [2] adapts a pretrained video diffusion model through latent-space training followed by pixel-space refinement, while SeedVR2 [35] combines progressive distillation with adversarial post-training. PSSR [38] trains a source-initialized base model in latent and pixel space, then supplements lightweight model evaluation with iterative refinement. We adopt latent-to-pixel training for draft-conditioned video generation and connect its completed predictions to correspondence estimation and reconstruction from observed keyframes.

3. RePix-VFI

Fig. 2 illustrates the proposed four-stage pipeline. A flow-rendered draft provides an initial estimate of intermediate motion and appearance. A pretrained video diffusion transformer then refines the draft's latent representation in a single forward pass per window. The decoded target frames connect generation to observation reconstruction: each prediction serves as both an RGB reference for estimating keyframe-to-target correspondences and an appearance candidate for fusion. The rendering path thus uses generated predictions for matching while drawing its RGB pixels from the observations, outside the VAE. A lightweight appearance fusion head combines these renderings with generated appearance.

Let $\mathbf{I} = \{I_t\}_{t=0}^{T-1}$ denote a target ground-truth video of T frames. For an interpolation factor s, the observed keyframes are $\mathcal{A} = \{I_k\}_{k \in \mathcal{K}_s}$, where $\mathcal{K}_s$ indexes frames sampled at intervals of s. The goal is to reconstruct the missing frames indexed by $\mathcal{M}_s = \{0, \dots, T-1\} \setminus \mathcal{K}_s$, which we refer to as target frames.

3.1. Flow-Rendered Draft

Optical flow estimation. For a target time $t \in \mathcal{M}_s$, let $\{k, k+s\} \subset \mathcal{K}_s$ be its bounding keyframe indices corresponding to two consecutive keyframes. In Stage 1 of Fig. 2, a frozen optical-flow estimator $\mathcal{F}$ [33] computes flows from I_k to its adjacent observed keyframes, $I_{k\pm s}$. Following quadratic motion interpolation [42], the flow from the left keyframe is given by

$$\begin{aligned}\mathbf{F}^{\text{draft}}_{k\to t} =& \frac{\tau}{2}\left(\mathbf{F}_{k\to k+s} - \mathbf{F}_{k\to k-s}\right) \\ &+ \frac{\tau^2}{2}\left(\mathbf{F}_{k\to k+s} + \mathbf{F}_{k\to k-s}\right),\end{aligned} \tag{1}$$

where $\tau = (t-k)/s \in (0,1)$ is the normalized interpolation time. The right-keyframe field $\mathbf{F}^{\text{draft}}_{k+s\to t}$ is estimated symmetrically using flows from I_{k+s} to I_k and I_{k+2s}. When the required outer keyframe is unavailable or the quadratic fit is invalid, we use linear motion from the corresponding bounding keyframe.

Draft rendering. Let $\mathbf{F}^{\text{draft}}$ stack the left- and right-keyframe intermediate flows for all target frames. We construct a draft $\mathbf{I}^{\text{fs}}$ via forward splatting to serve as the starting point for one-step diffusion generation:

$$\mathbf{I}^{\text{fs}} = \mathcal{R}(\mathcal{A}, \mathbf{F}^{\text{draft}}), \tag{2}$$

where $\mathcal{R}$ denotes the rendering operator. At each target time t, $\mathcal{R}$ bilinearly forward-splats the two bounding keyframes

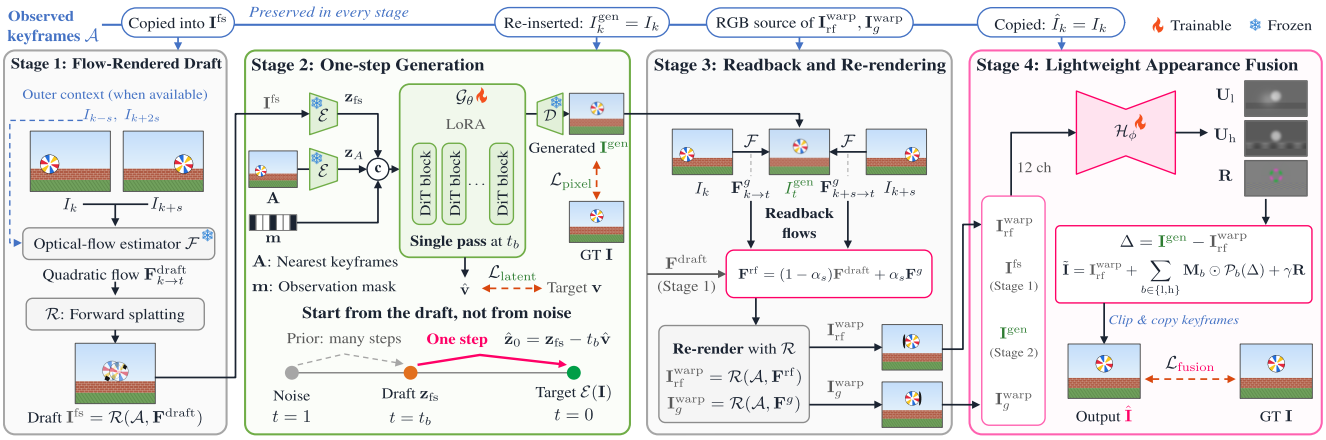

Figure 2. **Overview of RePix-VFI.** (i) **Stage 1**: A flow-based renderer constructs a draft from observed keyframes. (ii) **Stage 2**: A generator refines the draft latent in one forward pass, conditioned on the observations and their temporal mask. (iii) **Stage 3**: Mixing draft and readback flows yields $\mathbf{I}_{\text{rf}}^{\text{warp}}$, while readback flows alone yield $\mathbf{I}_g^{\text{warp}}$. (iv) **Stage 4**: A lightweight appearance fusion head combines four RGB inputs to predict spatial mixing between $\mathbf{I}_{\text{rf}}^{\text{warp}}$ and $\mathbf{I}^{\text{gen}}$, together with a residual correction.

along their displacement fields [23]. The splatted RGB contributions are weighted by $(1-\tau, \tau)$ and normalized by the corresponding weighted splat mass. Pixels receiving no contribution are filled with the corresponding values from the unwarped linear blend $(1-\tau)I_k + \tau I_{k+s}$. Observed keyframes are copied unchanged into $\mathbf{I}^{\text{fs}}$.

3.2. Draft-Conditioned One-Step Generation

Conditioning. As shown in Stage 2 of Fig. 2, the encoded flow-rendered draft serves as the initial latent state for a pretrained video diffusion transformer $\mathcal{G}_\theta$ [32]. We adapt the transformer for deterministic one-step refinement, with additional conditioning from the observed keyframes. Let $\mathcal{E}$ and $\mathcal{D}$ denote the frozen VAE encoder and decoder, respectively. We construct $\mathbf{A}$ by copying the temporally nearest observation to each target index and encode the two videos as $\mathbf{z}_{\text{fs}} = \mathcal{E}(\mathbf{I}^{\text{fs}})$ and $\mathbf{z}_A = \mathcal{E}(\mathbf{A})$. Inspired by the channel-wise video conditioning scheme [25], we concatenate $\mathbf{z}_{\text{fs}}$, $\mathbf{z}_A$, and an observation mask $\mathbf{m}$ along the channel dimension. The mask encodes the observed frame positions in a layout compatible with the VAE's temporal compression, distinguishing the observed keyframes from the copied ones for the target frames.

One-step refinement. We use $\mathbf{z}_{\text{fs}}$ directly, without adding diffusion noise, and evaluate $\mathcal{G}_\theta$ at a fixed normalized timestep t_b. With the pretrained rectified-flow velocity parameterization [17, 18], the predicted velocity defines a single latent correction toward the refined latent $\hat{\mathbf{z}}_0$ as:

$$\hat{\mathbf{v}} = \mathcal{G}_\theta\big([\mathbf{z}_{\text{fs}}, \mathbf{z}_A, \mathbf{m}], t_b\big), \qquad \hat{\mathbf{z}}_0 = \mathbf{z}_{\text{fs}} - t_b\hat{\mathbf{v}} \tag{3}$$

Then, we obtain $\tilde{\mathbf{I}}^{\text{gen}} = \mathcal{D}(\hat{\mathbf{z}}_0)$ after decoding. In $\tilde{\mathbf{I}}^{\text{gen}}$, we replace the generated frames at the observed frame positions with the corresponding observed frames, yielding $\mathbf{I}^{\text{gen}} = \{I_t^{\text{gen}}\}_{t=0}^{T-1}$ so that $I_k^{\text{gen}} = I_k$ for all $k \in \mathcal{K}_s$. Note that $\mathcal{G}_\theta$ shares the same parameters across interpolation factors.

Generation objective. We train LoRA adapters [10] in the attention and feed-forward layers of $\mathcal{G}_\theta$, together with the patch embedding and output projection; the remaining backbone parameters and the VAE are frozen. Newly introduced patch-embedding weights are zero-initialized. Building on source-initialized latent refinement and latent-to-pixel adaptation [38], we train the same generator in two successive stages. The latent stage supervises the draft-to-target correction using the target velocity

$$\mathbf{v} = \big(\mathbf{z}_{\text{fs}} - \mathcal{E}(\mathbf{I})\big)/t_b, \tag{4}$$

where the latent-stage objective is

$$\mathcal{L}_{\text{latent}} = \mathcal{L}_{\text{mse}}(\hat{\mathbf{v}}, \mathbf{v}) + \lambda_{\text{vsd}}\mathcal{L}_{\text{vsd}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}, \tag{5}$$

where $\lambda_{\text{vsd}} = 1$ and $\lambda_{\text{adv}} = 0.1$ are empirically set for all our experiments. The VSD loss compares predictions from a frozen reference score model and a learned auxiliary score model on re-noised generated latents [37, 43]. The adversarial loss uses a latent-space discriminator trained with a hinge loss. In the pixel stage, we subsequently refine the same generator using

$$\mathcal{L}_{\text{pixel}} = \mathcal{L}_{\text{mse}}^{\mathcal{M}_s}(\mathbf{I}^{\text{gen}}, \mathbf{I}) + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}}^{\mathcal{M}_s}(\mathbf{I}^{\text{gen}}, \mathbf{I}), \tag{6}$$

where $\lambda_{\text{lpips}} = 2$ and the superscript $\mathcal{M}_s$ restricts both loss functions to missing frames that we target. The two training stages supervise the draft-to-target latent correction and its decoded appearance, respectively.

3.3. Readback and Re-rendering

Generative readback. After decoding, each target prediction I_t^{gen} provides an explicit RGB reference at the unobserved time. In Stage 3 of Fig. 2, we reuse the frozen optical-flow estimator $\mathcal{F}$ to match both bounding keyframes directly

to this prediction:

$$\mathbf{F}^{g}_{k\to t} = \mathcal{F}(I_k, I_t^{\text{gen}}), \quad \mathbf{F}^{g}_{k+s\to t} = \mathcal{F}(I_{k+s}, I_t^{\text{gen}}). \quad (7)$$

Whereas the draft estimates target-time motion from correspondences between observations, readback estimates correspondences against the generated target-time RGB. This uses the flow estimator without VFI-specific correspondence training. Stacking these fields gives $\mathbf{F}^g$, which we blend with the draft flows to control the readback update:

$$\mathbf{F}^{\text{rf}} = \big(1-\alpha_s\big)\mathbf{F}^{\text{draft}} + \alpha_s \mathbf{F}^{g}, \quad (8)$$

where $\alpha_s \in [0,1]$ is calibrated separately for each interpolation factor s. Grid search on designated fusion-training windows selects $\alpha_s = 0.8, 0.9, 1.0$ for $s = 4, 8, 16$.

Re-rendering. We apply the rendering operator from Sec. 3.1 to the refined and readback-only flows:

$$\mathbf{I}^{\text{warp}}_{\text{rf}} = \mathcal{R}(\mathcal{A}, \mathbf{F}^{\text{rf}}), \qquad \mathbf{I}^{\text{warp}}_{g} = \mathcal{R}(\mathcal{A}, \mathbf{F}^{g}). \quad (9)$$

Both renderings draw their RGB values from the original keyframes, while the generated prediction guides their transport through the estimated correspondences. We use $\mathbf{I}^{\text{warp}}_{\text{rf}}$ as the reconstruction base and $\mathbf{I}^{\text{warp}}_{g}$ as an auxiliary fusion input; they coincide when $\alpha_s = 1$.

3.4. Lightweight Appearance Fusion

Stage 4 of Fig. 2 combines generated appearance $\mathbf{I}^{\text{gen}}$ with the observation rendering $\mathbf{I}^{\text{warp}}_{\text{rf}}$ to produce the final reconstruction. A framewise 2D U-Net $\mathcal{H}_\phi$ [28] predicts their spatial mixing and a residual correction, using the draft $\mathbf{I}^{\text{fs}}$ and readback-only rendering $\mathbf{I}^{\text{warp}}_{g}$ as additional conditioning. The four RGB inputs are concatenated into 12 channels, from which the head predicts two mixing logits, $\mathbf{U}_{\text{l}}$ and $\mathbf{U}_{\text{h}}$, and a 3-channel residual $\mathbf{R}$:

$$[\mathbf{U}_{\text{l}}, \mathbf{U}_{\text{h}}, \mathbf{R}] = \mathcal{H}_\phi\big([\mathbf{I}^{\text{warp}}_{\text{rf}}, \mathbf{I}^{\text{fs}}, \mathbf{I}^{\text{gen}}, \mathbf{I}^{\text{warp}}_{g}]\big). \quad (10)$$

Let $\Delta = \mathbf{I}^{\text{gen}} - \mathbf{I}^{\text{warp}}_{\text{rf}}$ denote the difference between the two appearance candidates. Separate spatial maps modulate its low- and high-frequency components, while the residual provides an additional RGB correction:

$$\tilde{\mathbf{I}} = \mathbf{I}^{\text{warp}}_{\text{rf}} + \sum\nolimits_{b\in\{\text{l},\text{h}\}} \mathbf{M}_b \odot \mathcal{P}_b(\Delta) + \gamma\mathbf{R}, \quad (11)$$

where $\mathbf{M}_b = \text{sigmoid}(\mathbf{U}_b)$. We empirically set $\gamma = 0.03$, allowing only small residual corrections beyond the adaptive mixing of re-rendered observations and generated appearance. The head includes calibrated logit offsets and a bounded residual transformation. Here, $\mathcal{P}_{\text{l}}$ is a Gaussian low-pass filter with a standard deviation of 2 pixels and $\mathcal{P}_{\text{h}} = \mathcal{I} - \mathcal{P}_{\text{l}}$, where $\mathcal{I}$ is the identity operator. At inference, we clip $\tilde{\mathbf{I}}$ to $[0,1]$ and copy the observed keyframes into the result to obtain the final output $\hat{\mathbf{I}}$, ensuring $\hat{I}_k = I_k$ for all $k \in \mathcal{K}_s$.

Fusion objective. With the generator $\mathcal{G}_\theta$ frozen, we optimize the appearance fusion head $\mathcal{H}_\phi$ using draft-relative reconstruction and perceptual losses:

$$\mathcal{L}_{\text{fusion}} = \mathcal{L}^{\text{rel},\mathcal{M}_s}_{\text{mse}}(\tilde{\mathbf{I}}, \mathbf{I}) + \lambda_{\text{lpips}}\mathcal{L}^{\text{rel},\mathcal{M}_s}_{\text{lpips}}(\hat{\mathbf{I}}, \mathbf{I}), \quad (12)$$

where rel denotes normalization by the corresponding draft error, and the perceptual objective includes a weighted component with temporal-distance. More details are provided in the *Appendix*.

4. Experiments

4.1. Experimental Setup

Training data and implementation. We fine-tune Wan2.2-TI2V-5B [32] on LAVIB [30] using rank-32 LoRA [10] and 49-frame, 512×512 crops, with keyframe intervals sampled uniformly from 2 to 16 and random temporal phases. Generator adaptation comprises 5k latent-stage and 30k pixel-stage updates at $t_b = 0.599$, while the appearance fusion head is trained for 13.5k updates in total using frozen generator outputs. Optimization, fusion initialization, and other details are provided in the *Appendix*.

Benchmarks and baselines. We evaluate at interpolation factors $s \in \{4, 8, 16\}$ using the SNU-FILM entire-video protocol [3, 25] and DAVIS 2016 [26] at native 1080p resolutions. At each factor, every s-th frame is retained as a keyframe, and the remaining frames are reconstructed. Table 1 compares our RePix-VFI with 15 VFI baselines using released weights, covering non-diffusion and diffusion-based VFI. All methods reconstruct the same target frames from the same subsampled videos using their native temporal contexts.

Metrics. We measure reconstruction fidelity using PSNR and SSIM [36], perceptual similarity using AlexNet-LPIPS [46], and motion-aware perceptual quality using FloLPIPS [5]. We additionally evaluate temporal consistency using a RAFT-based implementation of tOF [4, 33]. PSNR, SSIM, LPIPS, and FloLPIPS are computed on missing frames, whereas tOF uses all adjacent frame pairs, including observed-frame transitions. The metrics are averaged equally over sequences.

4.2. Quantitative Comparisons

SNU-FILM. As shown in Table 1, our RePix-VFI achieves the highest PSNR at all three factors and the lowest LPIPS and FloLPIPS at $\times 8$ and $\times 16$. At these two factors, PSNR improves over GIMM-VFI [8] by $0.86, 0.64$ dB and over MoMo [15], the strongest evaluated diffusion-based baseline, by $1.71, 1.48$ dB. For the temporal consistency metric tOF, our RePix-VFI reduces the error relative to the best competing result with large margins of 12.0%, 11.6%, and 4.8% at $\times 4$, $\times 8$, and $\times 16$, respectively. Together, these results show improved reconstruction accuracy and perceptual quality, accompanied by lower temporal motion discrepancy.

Table 1. **Quantitative comparison on SNU-FILM under the entire-video protocol.** Frame metrics are evaluated on missing frames from 31 clips; tOF uses all adjacent frame pairs, including observed-frame transitions. Scores are averaged equally over clips. **Bold** and underline indicate the best and second-best results. NFE is reported per synthesized frame, except per chunk for LDF-VFI and per window for RePix-VFI. Midpoint-only methods are applied recursively.

Method	Venue	NFE	SNU-FILM ×4					SNU-FILM ×8					SNU-FILM ×16				
			PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓
Non-diffusion																	
FILM [27]	ECCV'22	1	36.62	0.9815	0.0186	**0.0300**	0.2856	31.91	0.9501	0.0344	0.0565	0.4129	27.07	0.8879	0.0680	0.1094	0.6814
RIFE [11]	ECCV'22	1	35.72	0.9790	0.0211	0.0340	0.3248	31.19	0.9479	0.0364	0.0618	0.4601	26.59	0.8854	0.0678	0.1114	0.7617
AMT [16]	CVPR'23	1	36.73	0.9823	0.0291	0.0470	0.3766	32.04	0.9525	0.0495	0.0855	0.5086	27.11	0.8926	0.0971	0.1659	0.7852
EMA-VFI [44]	CVPR'23	1	36.63	0.9827	0.0296	0.0492	0.2924	32.27	0.9543	0.0496	0.0840	0.4073	27.53	0.8962	0.0914	0.1537	0.6055
GIMM-VFI [8]	NeurIPS'24	1	36.88	0.9828	0.0336	0.0527	0.2811	32.45	0.9554	0.0607	0.0970	0.3947	27.78	**0.9016**	0.1118	0.1762	0.5674
InterpAny [49]	ECCV'24	2	36.24	0.9809	0.0278	0.0477	0.3246	31.60	0.9496	0.0510	0.0926	0.4807	26.98	0.8898	0.1036	0.1717	0.7592
BiM-VFI [29]	CVPR'25	1	36.43	0.9812	0.0230	0.0371	0.3265	31.69	0.9493	0.0394	0.0650	0.4713	26.95	0.8886	0.0712	0.1149	0.6522
VTinker [39]	AAAI'26	1	36.04	0.9805	0.0186	0.0306	0.2902	31.67	0.9492	0.0340	0.0569	0.4141	27.07	0.8890	0.0652	0.1072	0.6186
Diffusion-based																	
LDMVFI [6]	AAAI'24	200	34.79	0.9746	0.0249	0.0450	0.3408	30.15	0.9364	0.0480	0.0877	0.5340	25.86	0.8727	0.0923	0.1521	0.8668
CBBD [20]	MM'24	50	35.95	0.9799	0.0194	0.0339	0.3000	31.02	0.9451	0.0390	0.0721	0.5148	26.20	0.8779	0.0828	0.1371	0.9153
MoMo [15]	AAAI'25	8	36.20	0.9801	**0.0178**	**0.0300**	0.3011	31.60	0.9480	0.0336	0.0577	0.4207	26.94	0.8869	0.0663	0.1110	0.6247
EDEN [47]	CVPR'25	2	35.44	0.9780	0.0203	0.0372	0.2913	31.14	0.9462	0.0392	0.0708	0.4185	26.91	0.8885	0.0768	0.1276	0.6337
TLB-VFI [19]	ICCV'25	10	35.94	0.9802	0.0182	0.0316	0.3092	30.96	0.9455	0.0367	0.0686	0.5193	26.49	0.8820	0.0742	0.1152	0.7655
SPEED [48]	MM'26	1	34.37	0.9721	0.0257	0.0391	0.2888	30.70	0.9420	0.0430	0.0703	0.4086	26.78	0.8857	0.0752	0.1196	0.5770
LDF-VFI [25]	CVPR'26	16	33.47	0.9670	0.0320	0.0428	0.3133	30.79	0.9442	0.0452	0.0634	0.4099	26.39	0.8794	0.0783	0.1160	0.5911
RePix-VFI	–	1	**36.98**	**0.9842**	0.0198	0.0310	**0.2474**	**33.31**	**0.9628**	**0.0322**	**0.0493**	**0.3490**	**28.42**	0.9014	**0.0604**	**0.0871**	**0.5400**

Table 2. **Quantitative comparison on DAVIS 2016.** PSNR and LPIPS are evaluated on missing frames from all 50 sequences at their native resolutions and averaged equally over sequences.

Method	×4		×8		×16	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
GIMM-VFI [8]	23.81	0.2737	20.59	0.3903	18.22	0.5046
BiM-VFI [29]	22.86	0.1643	19.71	0.2451	17.40	0.3413
VTinker [39]	22.71	0.1742	19.47	0.2680	17.39	0.3575
MoMo [15]	22.50	0.1718	19.33	0.2638	17.26	0.3523
EDEN [47]	22.34	0.3260	20.06	0.4239	18.38	0.4940
TLB-VFI [19]	20.62	0.2367	18.30	0.3097	17.14	0.3630
LDF-VFI [25]	22.74	0.2444	20.11	0.3055	17.71	0.3974
RePix-VFI	**24.56**	**0.1422**	**21.64**	**0.2099**	**19.17**	**0.2962**

DAVIS. Table 2 reports the VFI results averaged over all 50 DAVIS 2016 sequences. Our RePix-VFI achieves the highest PSNR and lowest LPIPS at all three interpolation factors, with PSNR gains of 0.75, 1.04, and 0.79 dB over the best competing results at ×4, ×8, and ×16, respectively. This extends the reconstruction and perceptual-quality improvement on high-resolution DAVIS sequences.

4.3. Qualitative Comparisons

Fig. 3 compares reconstructed frames at ×16 on SNU-FILM and DAVIS. In the SNU-FILM examples, our RePix-VFI better preserves the relative positions of overlapping people's legs and the configuration of the skateboarder's legs and board. Several competing predictions exhibit duplicated contours, distorted structures, or oversmoothed details in these regions. This advantage in intermediate placement is also evident in the fast-motion DAVIS motorbike example, where our reconstruction more closely matches the GT positions of the motorcycle and foreground planter. Beyond placement, the car-roundabout example shows more faithful rear-wheel geometry and car-body contours, together with the visible directional arrow on the partially occluded road sign behind the vehicle. These examples demonstrate the framework's ability to recover intermediate scene configurations while preserving local structure and appearance.

Table 3. **Reconstruction ablations on SNU-FILM.** Scores follow the entire-video protocol. $\mathbf{I}^{\mathrm{fs}}$-reference rendering replaces $\mathbf{I}^{\mathrm{gen}}$ with $\mathbf{I}^{\mathrm{fs}}$ as the correspondence reference under fixed rendering settings. With the generator frozen, *Fusion w/o readback* and *Full reconstruction* are trained with the same training strategy.

Output	×4		×8		×16	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Draft $\mathbf{I}^{\mathrm{fs}}$	35.15	0.0374	31.33	0.0598	27.12	0.0966
Generation $\mathbf{I}^{\mathrm{gen}}$ ($\mathcal{L}_{\mathrm{latent}}$)	33.24	0.0378	30.35	0.0599	26.59	0.0965
Generation $\mathbf{I}^{\mathrm{gen}}$ ($+\mathcal{L}_{\mathrm{pixel}}$)	34.83	0.0286	32.19	0.0402	28.03	0.0676
$\mathbf{I}^{\mathrm{fs}}$-reference rendering	35.07	0.0358	31.21	0.0538	26.89	0.0888
Re-rendering $\mathbf{I}_{\mathrm{rf}}^{\mathrm{warp}}$	35.52	0.0349	31.93	0.0519	27.46	0.0857
Fusion w/o readback	36.84	0.0228	33.04	0.0359	28.22	0.0646
Full reconstruction (Ours)	**36.98**	**0.0198**	**33.31**	**0.0322**	**28.42**	**0.0604**

4.4. Ablation Studies

Generator Adaptation. The two *Generation* $\mathbf{I}^{\mathrm{gen}}$ rows in Table 3 compare generator outputs before and after pixel-space adaptation on SNU-FILM. This additional adaptation improves PSNR by 1.44–1.84 dB and lowers LPIPS. The resulting $\mathbf{I}^{\mathrm{gen}}$ also achieves lower LPIPS than *Draft* $\mathbf{I}^{\mathrm{fs}}$ at all three factors, providing an appearance candidate with improved perceptual similarity for subsequent reconstruction.

Correspondence Reference for Re-rendering. To assess

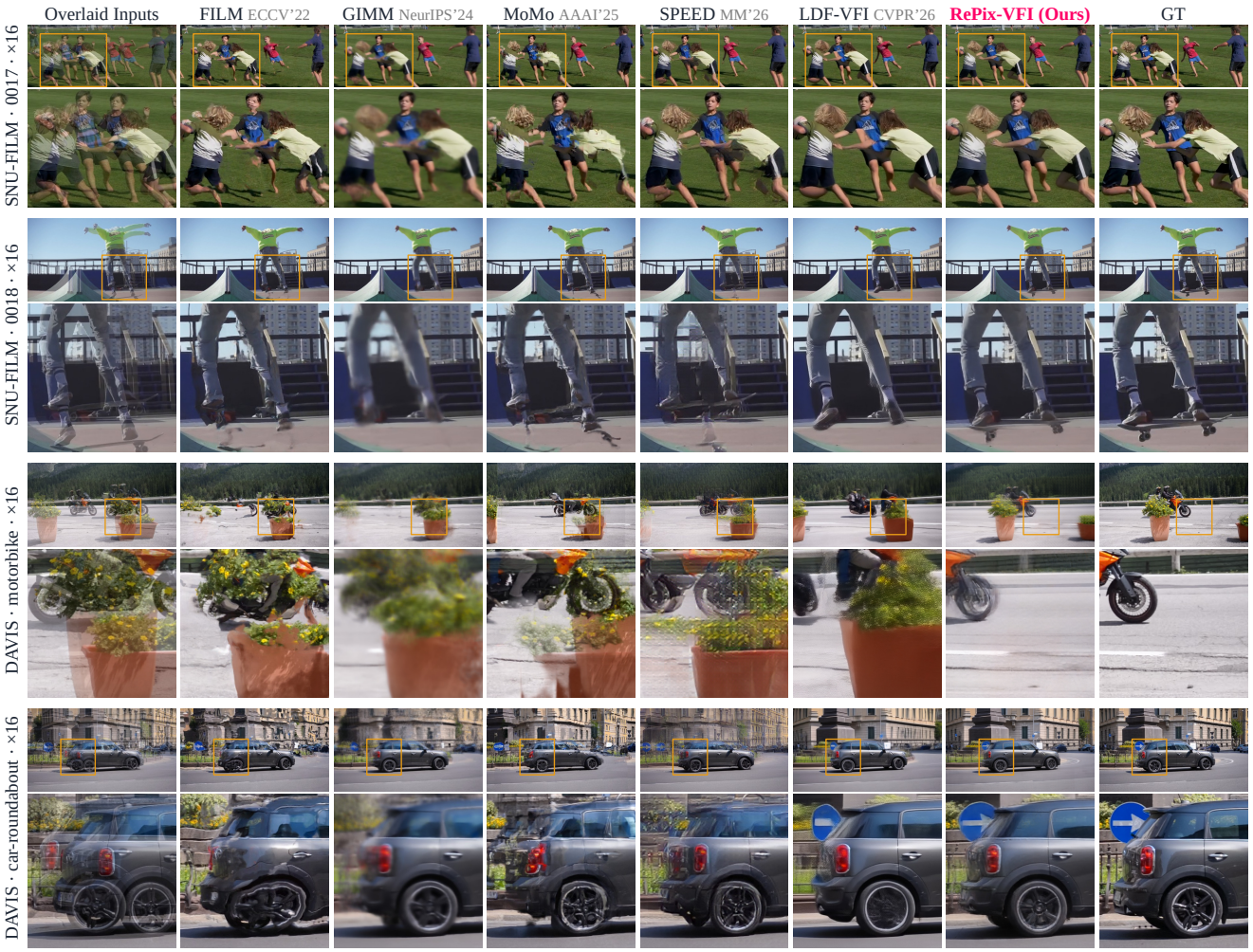

Figure 3. **Qualitative comparisons at ×16.** The top two examples are from SNU-FILM and the bottom two from DAVIS, all at the midpoint between observed keyframes ($\tau = 8/16$). The first column shows an equal-weight overlay of the bounding observations; subsequent columns show predictions and GT. Boxes indicate the regions enlarged below.

whether generation also benefits observation rendering, we compare $\mathbf{I}^{\text{fs}}$-*reference rendering* with *Re-rendering* $\mathbf{I}^{\text{warp}}_{\text{rf}}$ in Table 3. These use $\mathbf{I}^{\text{fs}}$ and $\mathbf{I}^{\text{gen}}$ as their respective correspondence references, with the observed RGB, frozen flow estimator, rendering resolution, renderer, and α_s held fixed. Using the generated reference improves rendering PSNR by 0.45, 0.72, and 0.57 dB at ×4, ×8, and ×16 and lowers LPIPS. At ×4, the generated prediction has lower PSNR than the draft, yet provides a better reference for observation rendering. Thus, RGB PSNR alone does not determine the usefulness of a prediction as a correspondence reference, supporting the use of generated predictions beyond their role as appearance candidates.

Fusion with Re-rendered Observations. We next evaluate whether re-rendered observations improve reconstruction beyond fusing the draft with generated appearance (Table 3). *Fusion w/o readback* retains the same fusion architecture and generated appearance candidate, replacing both re-rendered inputs $\mathbf{I}^{\text{warp}}_{\text{rf}}$ and $\mathbf{I}^{\text{warp}}_{g}$ with the draft $\mathbf{I}^{\text{fs}}$. Full reconstruction improves PSNR over this control by 0.14, 0.27, and 0.20 dB and reduces LPIPS by 0.0030, 0.0037, and 0.0042 at ×4, ×8, and ×16. It also outperforms generation alone and re-rendering alone by 0.39–2.15 dB and 0.96–1.46 dB, respectively, with lower LPIPS.

Fig. 4 illustrates these complementary roles on DAVIS car-turn. $\mathbf{I}^{\text{gen}}$ reduces the draft's duplicated car contour but oversmooths license-plate and headlight details, whereas re-rendering retains observed details with residual ghosting near the front tire. The *Full reconstruction* combines these cues, reducing visible ghosting while preserving clearer license-plate, headlight, and grille details than *Fusion w/o readback*.

Appearance Fusion Design. Keeping the generated and re-rendered inputs fixed, we examine how the fusion head combines them (Table 4). Free residual fusion predicts an unconstrained RGB correction to $\mathbf{I}^{\text{warp}}_{\text{rf}}$ without explicit mixing. The other controls replace the spatial mixing maps with

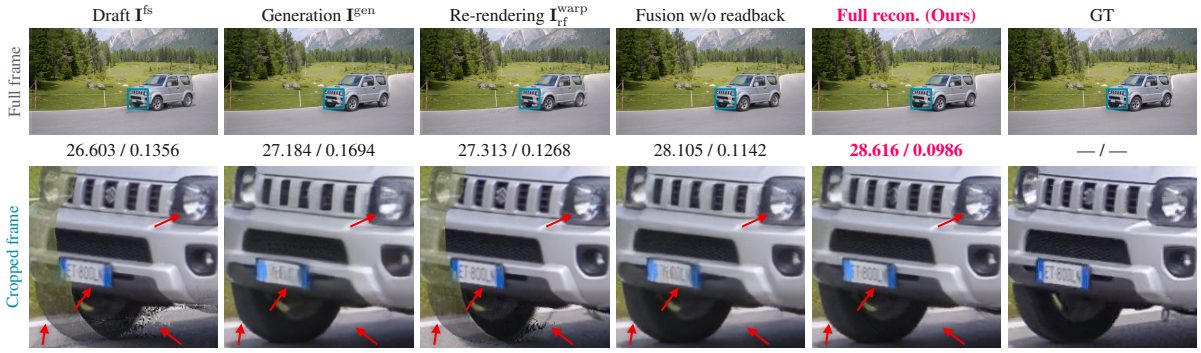

Figure 4. **Complementary roles of generation, re-rendering, and fusion on DAVIS car-turn ($\times$8, $\tau = 1/8$).** The top row shows full frames with PSNR↑ / LPIPS↓ computed over the entire image; the bottom enlarges the same cropped region for each output. Arrows highlight duplicated contours near the front tire and fine details around the license plate and headlight.

Table 4. **Appearance fusion design.** Evaluation on 89 LAVIB parent videos, averaged over three paired training runs. Within each run, variants share the generator, inputs, pretrained U-Net body, and adaptation budget. The initial output is $\mathbf{I}^{\mathrm{warp}}_{\mathrm{rf}}$ for free residual fusion and calibrated mixing for the other variants.

Variant	×4		×8		×16	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Spatial mixing, no residual	35.85	0.0267	32.80	0.0393	27.92	0.0695
Fixed mixing maps + residual	35.83	0.0262	32.77	0.0405	27.90	0.0752
Free residual fusion	35.86	0.0253	32.70	0.0401	27.82	0.0768
Full reconstruction (Ours)	**35.98**	**0.0240**	**32.85**	**0.0369**	**27.93**	**0.0676**

Table 5. **End-to-end inference cost.** Latency is total inference time per synthesized frame, including auxiliary processing and excluding model loading on a B200 GPU. NFE counts primary-network evaluations in the stated units. Memory is the maximum peak allocation across factors.

Method	NFE	s / frame↓				Peak Memory (GiB)
		×2	×4	×8	×16	
LDMVFI [6]	200 / frame	4.283	4.783	4.338	4.361	13.42
CBBD [20]	50 / frame	1.172	1.054	1.237	1.190	17.87
MoMo [15]	8 / frame	0.087	0.090	0.088	0.085	11.02
EDEN [47]	2 / frame	0.101	0.093	0.093	0.102	7.56
TLB-VFI [19]	10 / frame	0.441	0.445	0.438	0.431	34.24
SPEED [48]	1 / frame	0.013	0.019	0.020	0.018	6.38
LDF-VFI [25]	16 / chunk	10.304	7.048	6.093	7.305	11.91
RePix-VFI	1 / window	0.319	0.217	0.186	0.175	42.05

calibrated constants or remove residual correction. The full head achieves the highest PSNR and lowest LPIPS at all three factors, improving PSNR over free residual fusion by 0.12, 0.15, and 0.11 dB at ×4, ×8, and ×16, respectively. Fixing the mixing weights or removing the residual also worsens both metrics, although the ×16 PSNR difference without the residual is small (0.01 dB).

4.5. Efficiency

RePix-VFI evaluates the video generator once per window, sharing this computation across the target frames. Under the 49-frame, 720p/B200 protocol (Table 5), the complete pipeline requires 0.175–0.319 s per synthesized frame, including flow estimation, rendering, and fusion, achieving a 32–42× speedup over the evaluated 16-step LDF-VFI [25]. This reduction comes with higher peak allocated memory (42.05 vs. 11.91 GiB), while MoMo, EDEN, and SPEED have lower latency. At ×8, enabling readback adds 0.050 s per synthesized frame, resulting in a total latency of 0.186 s/frame. For this factor, Table 3 shows that *Full reconstruction (Ours)* improves PSNR by 0.27 dB and reduces LPIPS by 0.0037 over *Fusion w/o readback* on SNU-FILM.

4.6. Limitations

Our reconstruction remains sensitive to the accuracy of the generated reference and the estimated correspondences, as errors in either can propagate to the re-rendered observations and persist after fusion. The framework also retains substantial computational and memory requirements: although one-step generation avoids repeated sampling, the video backbone and subsequent reconstruction stages remain costly compared with lightweight VFI methods.

5. Conclusion

We presented RePix-VFI, a multi-frame VFI framework that couples draft-conditioned one-step video diffusion with reconstruction from observed keyframes. Generative readback matches observed keyframes to generated predictions using a frozen optical-flow estimator, providing correspondences for re-rendering their original RGB pixels. A lightweight appearance fusion head combines these renderings with generated appearance. Experiments on SNU-FILM and DAVIS demonstrate improved reconstruction, perceptual quality, and temporal consistency, while controlled ablations further support the use of generated correspondence references. This design allows generated predictions to guide the placement of observed pixels without requiring the generator to reproduce all available appearance details.

Table 6. **Overview of the *Appendix*.**

Section	Contents
Sec. A	Implementation and evaluation details
Sec. B	Component analysis
Sec. C	Benchmark results and efficiency
Sec. D	Visual results
Sec. E	Limitations and future work

This supplement analyzes how one-step video predictions guide correspondence estimation for re-rendering original keyframe RGB outside the VAE. It includes implementation details, component controls, benchmark and visual comparisons, efficiency measurements, and limitations (Table 6).

A. Implementation and Evaluation Details

Draft and generator. We construct the flow-rendered draft using flows estimated by frozen RAFT-Large at half the output height and width. Readback flows are estimated at full output resolution. The generator takes VAE encodings of the flow-rendered draft $\mathbf{I}^{\text{fs}}$ and the observation sequence $\mathbf{A}$, concatenated with the observation mask $\mathbf{m}$ to form a 100-channel input $(48 + 48 + 4)$. Text conditioning uses a fixed empty-prompt embedding. We adapt the generator using AdamW with a learning rate of 5×10^{-5} and a batch size of 24, maintaining an EMA with decay 0.999. During latent-stage adaptation, the frozen Wan model [32] provides the reference score; an auxiliary model with separate LoRA adapters is trained with a flow-matching objective on detached generated latents. A framewise 2D discriminator operating in latent space supplies the adversarial loss. Pixel-stage adaptation uses VGG-LPIPS [46].

Fusion training. We train the 1.19M-parameter appearance fusion head using AdamW with a learning rate of 10^{-4}. Training consists of 10.5k RGB-residual pretraining updates followed by 3k adaptation updates. For the pair in Table 3 of the main paper, *Full reconstruction* and *Fusion w/o readback* share the frozen generator, training data, optimization settings, and adaptation budget. Both are initialized from the same U-Net body, pretrained using the inputs in Eq. (10) of the main paper. The output projection is zero-initialized, so adaptation starts from the mixing weights given by the fixed logit offsets and from a zero residual $\mathbf{R}$, which is bounded by $\tanh$. During adaptation, the perceptual term in Eq. (12) of the main paper combines relative VGG-LPIPS with $\lambda(d)$-weighted relative AlexNet-LPIPS. Here, $d = \min_{k \in \mathcal{K}_s} |t - k|$ is the distance in frames to the nearest observed keyframe. The weight $\lambda(d)$ increases linearly from zero at $d = 1$ to one at $d = 4$ and remains one thereafter.

Calibration. Calibration uses the same LAVIB [30] validation segments as fusion training. The 48 parent videos in Table 7 and the 89 parent videos in Table 4 of the main paper are disjoint from each other and from the generator-adaptation, fusion-training, and calibration sets at the LAVIB parent-ID level. Grid search selects α_s by minimizing relative MSE plus twice relative VGG-LPIPS, and the logit offsets by maximizing mean PSNR. *Fusion w/o readback* is calibrated separately. Both α_s and the logit offsets remain fixed during adaptation and evaluation.

Evaluation. RePix-VFI uses native-resolution 49-frame windows (stride 48), with shorter windows ending at observed keyframes for short clips. For each factor, all methods are evaluated on the same target frames, excluding frames after the last observed keyframe. We compute tOF [4] using RAFT-Small [33]. The tLP metric [4] is the absolute difference between adjacent-frame AlexNet-LPIPS distances in the reconstructed and GT videos, averaged over all pairs and scaled by 100. For FVD and temporal metrics on DAVIS, REDS, and Adobe240, observed-frame outputs are replaced with GT, while SNU-FILM tOF uses each method's observed-frame outputs.

Table 7. **Linear versus flow-rendered draft initialization.** Results on 48 held-out LAVIB parent videos. Generated rows use 5k latent-stage updates only (one run per source); output differences are draft minus linear.

Output	Metric	×2	×4	×8	×16
Linear source (input)	PSNR↑	28.19	25.12	22.64	20.45
	LPIPS↓	0.1058	0.1161	0.1550	0.2192
Generated from linear source	PSNR↑	30.31	28.45	27.82	24.79
	LPIPS↓	0.0674	0.0769	0.0871	0.1255
Flow-rendered draft $\mathbf{I}^{\text{fs}}$ (input)	PSNR↑	37.61	35.31	32.11	26.68
	LPIPS↓	0.0262	0.0350	0.0554	0.1116
Generated from $\mathbf{I}^{\text{fs}}$	PSNR↑	35.60	33.95	31.35	26.36
	LPIPS↓	0.0258	0.0351	0.0561	0.1122
Generated-output difference	ΔPSNR	**+5.29**	**+5.50**	**+3.53**	**+1.57**
	ΔLPIPS	**-0.0416**	**-0.0418**	**-0.0310**	**-0.0133**

B. Component Analysis

B.1. Source Initialization

Table 7 compares a linear RGB blend of the bounding observations with the flow-rendered draft $\mathbf{I}^{\text{fs}}$ as the generator's source initialization. Conditioning and optimization settings are shared, following Sec. A. At a fixed budget of 5k latent-stage updates, the flow-rendered source yields higher generated-frame PSNR (by 1.57–5.50 dB) and lower LPIPS than linear RGB initialization at every factor. This comparison uses 49-frame 512×512 windows and precedes pixel-stage adaptation, readback, and appearance fusion. Separately, pixel-stage adaptation raises the PSNR of $\mathbf{I}^{\text{gen}}$ by 1.44–1.84 dB on SNU-FILM (Table 3 of the main paper).

Table 8. **Correspondence-reference and RGB-source controls on SNU-FILM.** Pre-fusion quality differences in $\mathbf{I}_{\mathrm{rf}}^{\mathrm{warp}}$. (a) fixes the flow estimator and rendering settings; (b) additionally fixes the estimated flows. Positive ΔPSNR and negative ΔLPIPS favor the first named choice.

Metric	×4	×8	×16
(a) Generated − draft correspondence reference			
ΔPSNR (dB)↑	+0.45	+0.72	+0.57
ΔLPIPS↓	-0.0009	-0.0019	-0.0031
(b) Original − VAE-reconstructed keyframe RGB			
ΔPSNR (dB)↑	+1.98	+1.08	+0.50
ΔLPIPS↓	-0.0190	-0.0172	-0.0150

B.2. Correspondence Reference and Keyframe RGB Source

The controls in Table 8 support using generated predictions for correspondence estimation and original observations for re-rendering. Table 8(a) summarizes the reference control in Table 3 of the main paper. Table 8(b) replaces original keyframe RGB I_k with $\mathcal{D}(\mathcal{E}(I_k))$, applying the frozen VAE to each keyframe as a one-frame video. With flows and the renderer fixed, using original keyframe RGB rather than its VAE reconstruction improves pre-fusion PSNR by 0.50–1.98 dB and lowers LPIPS at every factor.

B.3. Fusion Training and Reference Choice

Table 9(a) tests two fusion-training configurations with three paired runs each. One pretrains each U-Net body with its own inputs; the other shares no-readback pretraining and uses full-resolution draft flows for fusion. Both retain the generator's input draft. Each pair shares the generator, training data, objective, and adaptation budget. Readback lowers mean LPIPS at every factor; PSNR improves at ×4 and ×8 and changes by less than 0.04 dB at ×16. The reference control in Table 8(a) is measured before appearance fusion and does not depend on these configurations.

Table 9(b) uses GIMM-VFI predictions as correspondence references while retaining $\mathbf{I}^{\mathrm{gen}}$ for appearance fusion. Rendering settings, fusion initialization, the adaptation budget, and the calibration procedure are shared (Sec. A).

GIMM-VFI uses native two-frame inputs and has higher RGB PSNR than $\mathbf{I}^{\mathrm{gen}}$ at ×4 and ×8 (Tables 1 and 3 of the main paper). Yet generated references yield higher final PSNR at these factors and lower LPIPS at all factors. Thus, RGB PSNR alone does not determine a prediction's usefulness as a correspondence reference.

B.4. Fusion Design on Public Benchmarks

In Table 10, the full appearance fusion head achieves the lowest mean LPIPS at all SNU-FILM and Adobe240 settings, and the highest mean PSNR at ×4 and ×8 on both datasets.

Table 9. **Readback and reference controls on SNU-FILM.** Mean paired differences after appearance fusion over three training runs.

Metric	×4	×8	×16
(a) Readback − no readback			
Separate pretraining; half-res. draft flows			
ΔPSNR (dB)↑	+0.033	+0.160	+0.007
ΔLPIPS↓	-0.0009	-0.0023	-0.0026
Shared no-readback pretraining; full-res. draft flows			
ΔPSNR (dB)↑	+0.009	+0.077	-0.037
ΔLPIPS↓	-0.0006	-0.0024	-0.0026
(b) Generated − GIMM-VFI reference			
ΔPSNR (dB)↑	+0.089	+0.153	-0.005
ΔLPIPS↓	-0.0004	-0.0019	-0.0030

Table 10. **Fusion design on public benchmarks.** Three-run means on (a) SNU-FILM (31 clips) and (b) Adobe240 (17 sequences), using the heads from Table 4 of the main paper without further adaptation. **Bold** and underline indicate the best and second-best results.

Variant	×4		×8		×16	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
(a) SNU-FILM						
Spatial mixing, no residual	36.81	0.0249	33.27	0.0368	**28.44**	0.0642
Fixed mixing maps + residual	36.67	0.0220	33.16	0.0349	28.37	0.0639
Free residual fusion	36.75	0.0211	33.11	0.0340	28.31	0.0636
Full reconstruction (Ours)	**36.95**	**0.0203**	**33.30**	**0.0325**	28.42	**0.0607**
(b) Adobe240						
Spatial mixing, no residual	35.21	0.0280	33.13	0.0339	**27.46**	0.0585
Fixed mixing maps + residual	35.31	0.0235	33.16	0.0301	27.43	0.0558
Free residual fusion	35.36	**0.0230**	33.18	0.0294	27.41	0.0550
Full reconstruction (Ours)	**35.39**	**0.0230**	**33.22**	**0.0292**	27.44	**0.0544**

C. Benchmark Results and Efficiency

C.1. Extended Metrics on SNU-FILM and DAVIS

Table 11 adds FVD [34] to the SNU-FILM [3] comparison, and Table 13 extends the DAVIS [26] comparison to all 15 baselines and seven metrics. On DAVIS, RePix-VFI leads in PSNR, SSIM, LPIPS, FloLPIPS, tOF, and tLP at all three factors, while ranking second or third in FVD. On SNU-FILM, it ranks second in FVD at ×8 and ×16.

Observed-frame control. The SNU-FILM tOF advantage over LDF-VFI remains when observed frames are replaced with GT: LDF-VFI obtains 0.3177, 0.4118, and 0.5917 at ×4, ×8, and ×16, respectively, versus RePix-VFI's 0.2474, 0.3490, and 0.5400.

C.2. Generalization to REDS and Adobe240

We evaluate on the REDS validation set [22] and the Adobe240 [31] test split from VideoINR [1], without retraining or recalibration (Tables 14 and 15). RePix-VFI achieves the highest PSNR and lowest tOF on both datasets

Table 11. **Video-level FVD comparison on SNU-FILM.** Lower FVD is better. **Bold** and underline indicate the best and second-best results.

Method	×4	×8	×16
Non-diffusion			
FILM [27]	**4.17**	15.90	36.90
RIFE [11]	5.95	18.68	37.35
AMT [16]	6.63	19.39	60.22
GIMM-VFI [8]	7.20	22.10	70.67
InterpAny [49]	9.24	22.28	55.59
BiM-VFI [29]	5.68	17.20	35.79
VTinker [39]	5.64	18.23	42.60
EMA-VFI [44]	5.73	20.12	53.12
Diffusion-based			
LDMVFI [6]	8.29	26.81	66.89
CBBD [20]	5.09	19.56	55.22
MoMo [15]	4.75	19.68	44.68
EDEN [47]	9.49	25.20	59.35
TLB-VFI [19]	<u>4.42</u>	17.91	38.60
SPEED [48]	6.53	22.97	51.91
LDF-VFI [25]	12.50	**14.98**	**25.89**
RePix-VFI	5.35	<u>15.42</u>	<u>35.53</u>

Table 12. **Reconstruction stage costs.** Separately synchronized profiling in seconds per 49-frame, 720p window at ×8 on an NVIDIA B200. The median is computed separately for each stage and for the total runtime.

Stage	Time (s)
Draft construction	0.71
One-step generation and VAE	4.40
Readback and re-rendering	2.21
Fusion and remaining processing	0.57
Total (instrumented run)	7.90

at all three factors. It also obtains the lowest FloLPIPS on REDS and the best results in all five metrics on Adobe240 at ×8 and ×16. Where it does not achieve the lowest LPIPS, its score is within 0.001 of the best result.

C.3. Runtime and Memory

We follow the 49-frame, 720p/B200 protocol in Table 5 of the main paper: median end-to-end latency after warm-up, excluding model loading. Per-frame latency is computed by dividing window time by the number of synthesized frames: 24, 36, 42, and 45 at ×2, ×4, ×8, and ×16, respectively. Methods retain their native numerical precision. LDF-VFI uses 16 sampling steps and fixed 40-frame increments in the 49-frame wrapper; quality evaluation uses its native entire-video script. For RePix-VFI, the median end-to-end latency at ×8 is 7.81 s/window (0.186 s/synthesized frame). A separate profiling run with synchronization between stages gives a median total of 7.90 s (Table 12). Generation and VAE processing account for approximately 56% of the total, and readback and re-rendering for 28%. Peak memory is reported in Table 5 of the main paper and measures allocated CUDA memory, excluding unused reservations.

D. Visual Results

Fig. 5 extends the DAVIS motorbike example in Fig. 3 of the main paper across target times. RePix-VFI closely follows the changing object positions in GT as the motorcycle passes behind the planter. At ×8, RePix-VFI preserves distinct ball and mouth contours on SNU-FILM, leg separation on DAVIS, and legible lettering on Adobe240 and REDS (Fig. 6). At ×16, it preserves wing shape, vehicle contours, and separation between nearby pedestrians (Fig. 7).

Accompanying demo videos show all qualitative examples from the main paper and this supplement at playback rates of 24, 8, and 4 fps.

E. Limitations and future work

Reconstructing fine structures under fast motion remains challenging, as illustrated by the wheel blur at $\tau = 0.750$ in Fig. 5. In the accompanying DAVIS car-roundabout and car-shadow demo videos, wheels can show less rotation than in GT while the vehicle moves. A temporal loss focused on moving object parts could help recover such local motion. A smaller video backbone and reduced-resolution readback could reduce the dominant costs in Table 12.

Table 13. **Extended quantitative comparison on DAVIS 2016.** Evaluation on all 50 sequences at native resolutions. Temporal metrics and FVD use GT observed frames (Sec. A). **Bold** and underline indicate the best and second-best results.

Method / ×4	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	tLP↓	FVD↓
Non-diffusion							
FILM [27]	21.80	0.7587	0.2048	0.2509	23.79	6.286	169.41
RIFE [11]	22.83	0.7882	0.1683	0.2171	17.80	4.291	118.50
AMT [16]	21.76	0.7682	0.2999	0.3687	28.20	6.546	237.96
GIMM-VFI [8]	23.81	0.8209	0.2737	0.3293	15.67	6.322	145.37
InterpAny [49]	22.27	0.7812	0.3049	0.3785	23.77	6.349	242.35
BiM-VFI [29]	22.86	0.7910	0.1643	0.2130	12.67	3.059	**81.34**
VTinker [39]	22.71	0.7818	0.1742	0.2218	18.68	4.410	112.24
EMA-VFI [44]	22.61	0.7876	0.2743	0.3332	23.07	6.324	163.35
Diffusion-based							
LDMVFI [6]	20.87	0.7362	0.2499	0.3001	26.24	9.270	356.79
CBBD [20]	20.60	0.7339	0.2634	0.3118	27.50	7.869	339.36
MoMo [15]	22.50	0.7780	0.1718	0.2231	15.80	3.195	104.36
EDEN [47]	22.34	0.7635	0.3260	0.3717	25.72	9.031	369.95
TLB-VFI [19]	20.62	0.7322	0.2367	0.2838	27.53	10.192	306.43
SPEED [48]	23.15	0.7951	0.1570	0.2022	12.74	4.219	114.93
LDF-VFI [25]	22.74	0.7894	0.2444	0.2885	18.82	5.668	103.08
RePix-VFI	**24.56**	**0.8263**	**0.1422**	**0.1785**	**8.11**	**2.672**	90.44

Method / ×8	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	tLP↓	FVD↓
Non-diffusion							
FILM [27]	18.87	0.6755	0.2918	0.3348	28.38	12.758	402.86
RIFE [11]	19.71	0.7035	0.2553	0.3015	24.74	9.505	274.82
AMT [16]	19.12	0.6974	0.4137	0.4706	31.22	10.274	463.73
GIMM-VFI [8]	20.59	0.7437	0.3903	0.4424	22.94	10.607	299.87
InterpAny [49]	19.53	0.7081	0.4108	0.4680	28.00	9.134	432.54
BiM-VFI [29]	19.71	0.7048	0.2451	0.2933	20.65	5.083	186.78
VTinker [39]	19.47	0.6910	0.2680	0.3160	25.83	10.453	304.52
EMA-VFI [44]	19.76	0.7146	0.3933	0.4470	27.53	9.802	373.10
Diffusion-based							
LDMVFI [6]	18.67	0.6675	0.3200	0.3612	29.49	16.519	524.84
CBBD [20]	18.29	0.6601	0.3492	0.3899	29.73	12.802	553.53
MoMo [15]	19.33	0.6888	0.2638	0.3145	24.04	6.053	278.77
EDEN [47]	20.06	0.6998	0.4239	0.4605	32.12	12.981	705.41
TLB-VFI [19]	18.30	0.6591	0.3097	0.3508	29.72	16.402	506.87
SPEED [48]	20.12	0.7126	0.2426	0.2881	21.31	8.222	283.50
LDF-VFI [25]	20.11	0.7190	0.3055	0.3535	24.68	7.539	**182.83**
RePix-VFI	**21.64**	**0.7549**	**0.2099**	**0.2455**	**13.53**	**4.784**	192.52

Method / ×16	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	tLP↓	FVD↓
Non-diffusion							
FILM [27]	17.06	0.6136	0.3692	0.4039	29.72	18.118	584.19
RIFE [11]	17.54	0.6331	0.3475	0.3843	28.39	15.606	468.43
AMT [16]	17.53	0.6471	0.5107	0.5515	31.21	15.768	793.09
GIMM-VFI [8]	18.22	0.6752	0.5046	0.5437	27.51	17.109	559.78
InterpAny [49]	17.81	0.6544	0.5028	0.5439	29.57	13.458	651.71
BiM-VFI [29]	17.40	0.6280	0.3413	0.3810	26.06	9.471	364.75
VTinker [39]	17.39	0.6221	0.3575	0.3949	28.98	16.870	516.47
EMA-VFI [44]	17.99	0.6606	0.4990	0.5417	29.22	15.252	662.71
Diffusion-based							
LDMVFI [6]	17.23	0.6159	0.3813	0.4115	30.08	21.370	711.21
CBBD [20]	16.77	0.6051	0.4291	0.4591	30.10	15.504	756.59
MoMo [15]	17.26	0.6189	0.3523	0.3936	28.33	10.935	509.24
EDEN [47]	18.38	0.6488	0.4940	0.5217	32.87	17.676	1355.58
TLB-VFI [19]	17.14	0.6195	0.3630	0.3984	29.40	17.767	543.80
SPEED [48]	17.85	0.6413	0.3399	0.3764	27.30	13.932	547.51
LDF-VFI [25]	17.71	0.6480	0.3974	0.4405	26.97	9.376	**348.81**
RePix-VFI	**19.17**	**0.6848**	**0.2962**	**0.3275**	**19.92**	**8.464**	391.26

Table 14. **Quantitative comparison on REDS.** Evaluation on all 30 official 120-fps validation sequences at native 1280 × 720. **Bold** and underline indicate the best and second-best results.

Method	REDS ×4					REDS ×8					REDS ×16				
	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓
Non-diffusion															
FILM [27]	29.90	0.8963	0.0369	0.0397	0.5516	24.21	0.7542	0.0752	0.0822	1.0115	20.69	0.6471	0.1382	0.1573	1.5903
RIFE [11]	29.09	0.8891	0.0410	0.0437	0.5907	24.11	0.7544	0.0755	0.0808	1.0221	20.69	0.6477	0.1342	0.1473	1.6355
AMT [16]	30.05	0.8999	0.0456	0.0512	0.5873	24.33	0.7604	0.0903	0.1070	1.1133	20.81	0.6576	0.1835	0.2279	1.9164
GIMM-VFI [8]	30.59	0.9086	0.0576	0.0618	0.5072	25.00	0.7820	0.1224	0.1297	0.9322	21.55	**0.6866**	0.2332	0.2487	1.4521
InterpAny [49]	29.82	0.8979	0.0514	0.0573	0.5669	24.37	0.7639	0.1046	0.1183	1.0552	20.99	0.6653	0.2036	0.2343	1.6916
BiM-VFI [29]	29.79	0.8943	0.0396	0.0432	0.5878	24.19	0.7529	0.0768	0.0831	1.0253	20.77	0.6484	0.1366	0.1500	1.5256
VTinker [39]	29.81	0.8954	0.0368	0.0391	0.5459	24.27	0.7546	**0.0735**	0.0786	0.9834	20.87	0.6501	0.1328	0.1451	1.5073
EMA-VFI [44]	30.22	0.9031	0.0487	0.0533	0.5438	24.49	0.7646	0.0922	0.1007	0.9959	21.02	0.6627	0.1719	0.1938	1.5353
Diffusion-based															
LDMVFI [6]	28.77	0.8787	0.0565	0.0640	0.6950	23.70	0.7411	0.1072	0.1228	1.2991	20.47	0.6348	0.1830	0.2060	2.0141
CBBD [20]	29.65	0.8944	0.0390	0.0457	0.6275	23.87	0.7499	0.0840	0.1037	1.3145	20.37	0.6405	0.1616	0.1903	2.1002
MoMo [15]	29.70	0.8924	**0.0365**	0.0392	0.5554	24.10	0.7502	0.0743	0.0802	1.0030	20.70	0.6451	0.1347	0.1481	1.5137
EDEN [47]	29.44	0.8905	0.0453	0.0478	0.5713	24.43	0.7628	0.0939	0.0994	1.0423	21.24	0.6666	0.1699	0.1807	1.6710
TLB-VFI [19]	29.58	0.8941	0.0373	0.0436	0.6310	23.75	0.7467	0.0811	0.1004	1.3140	20.58	0.6486	0.1471	0.1654	1.6808
SPEED [48]	28.39	0.8764	0.0538	0.0552	0.5562	24.06	0.7556	0.0941	0.0976	0.9861	20.98	0.6605	0.1584	0.1670	1.4789
LDF-VFI [25]	28.37	0.8775	0.0637	0.0660	0.4886	24.26	0.7658	0.0920	0.0956	0.8834	20.80	0.6520	0.1492	0.1572	1.4186
RePix-VFI	**33.01**	**0.9478**	0.0367	**0.0385**	**0.3280**	**25.84**	**0.8010**	0.0740	**0.0766**	**0.7898**	**21.67**	0.6764	**0.1304**	**0.1360**	**1.3559**

Table 15. **Quantitative comparison on Adobe240.** Evaluation on all 17 test sequences at native 1280 × 720. **Bold** and underline indicate the best and second-best results.

Method	Adobe240 ×4					Adobe240 ×8					Adobe240 ×16				
	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	tOF↓
Non-diffusion															
FILM [27]	34.89	0.9730	0.0225	0.0248	1.0921	31.57	0.9460	0.0326	0.0366	1.1693	26.07	0.8592	0.0629	0.0734	1.3693
RIFE [11]	34.00	0.9685	0.0256	0.0277	1.1484	30.93	0.9421	0.0348	0.0384	1.2311	25.74	0.8575	0.0624	0.0699	1.4789
AMT [16]	34.99	0.9738	0.0308	0.0360	1.1750	31.63	0.9473	0.0437	0.0523	1.2454	26.16	0.8630	0.0838	0.1051	1.5145
GIMM-VFI [8]	35.10	0.9746	0.0332	0.0377	1.0876	31.93	0.9506	0.0475	0.0536	1.1574	26.72	0.8740	0.0896	0.0986	1.2916
InterpAny [49]	34.57	0.9717	0.0300	0.0367	1.1427	31.26	0.9446	0.0438	0.0551	1.2211	25.95	0.8616	0.0865	0.1056	1.4323
BiM-VFI [29]	34.88	0.9730	0.0253	0.0282	1.1210	31.52	0.9454	0.0349	0.0392	1.2054	26.13	0.8598	0.0632	0.0708	1.3493
VTinker [39]	34.57	0.9719	0.0225	0.0249	1.1082	31.39	0.9451	0.0323	0.0358	1.1775	26.18	0.8607	0.0600	0.0675	1.3280
EMA-VFI [44]	34.92	0.9735	0.0311	0.0364	1.1053	31.76	0.9483	0.0421	0.0490	1.1714	26.39	0.8664	0.0770	0.0909	1.3411
Diffusion-based															
LDMVFI [6]	34.00	0.9671	0.0273	0.0311	1.1343	30.42	0.9327	0.0445	0.0524	1.2819	25.26	0.8420	0.0856	0.0995	1.5250
CBBD [20]	34.54	0.9711	0.0235	0.0267	1.1121	31.03	0.9408	0.0366	0.0456	1.2806	25.46	0.8495	0.0746	0.0919	1.5791
MoMo [15]	34.69	0.9718	**0.0219**	**0.0242**	1.0868	31.41	0.9444	0.0320	0.0359	1.1657	26.08	0.8584	0.0606	0.0691	1.3151
EDEN [47]	34.64	0.9713	0.0231	0.0251	1.1007	31.40	0.9436	0.0359	0.0396	1.1899	26.33	0.8625	0.0704	0.0775	1.3598
TLB-VFI [19]	34.47	0.9712	0.0225	0.0253	1.1234	30.99	0.9411	0.0350	0.0436	1.2888	25.69	0.8559	0.0688	0.0795	1.4191
SPEED [48]	33.64	0.9664	0.0265	0.0281	1.0994	30.82	0.9401	0.0383	0.0412	1.1731	26.10	0.8606	0.0676	0.0733	1.3114
LDF-VFI [25]	32.54	0.9606	0.0335	0.0369	1.1376	30.76	0.9441	0.0406	0.0447	1.2109	25.97	0.8660	0.0671	0.0726	1.3626
RePix-VFI	**35.40**	**0.9753**	0.0228	0.0247	**1.0738**	**33.22**	**0.9610**	**0.0289**	**0.0318**	**1.1146**	**27.42**	**0.8840**	**0.0541**	**0.0577**	**1.2586**



Figure 5. **Multiple target times on DAVIS motorbike at ×16.** Targets 00018–00030 in steps of two lie between observed keyframes 00016 and 00032 ($\tau = 0.125, 0.250, \ldots, 0.875$). The first column shows an equal-weight overlay of the bounding observations. All methods and target times use the same fixed crop.

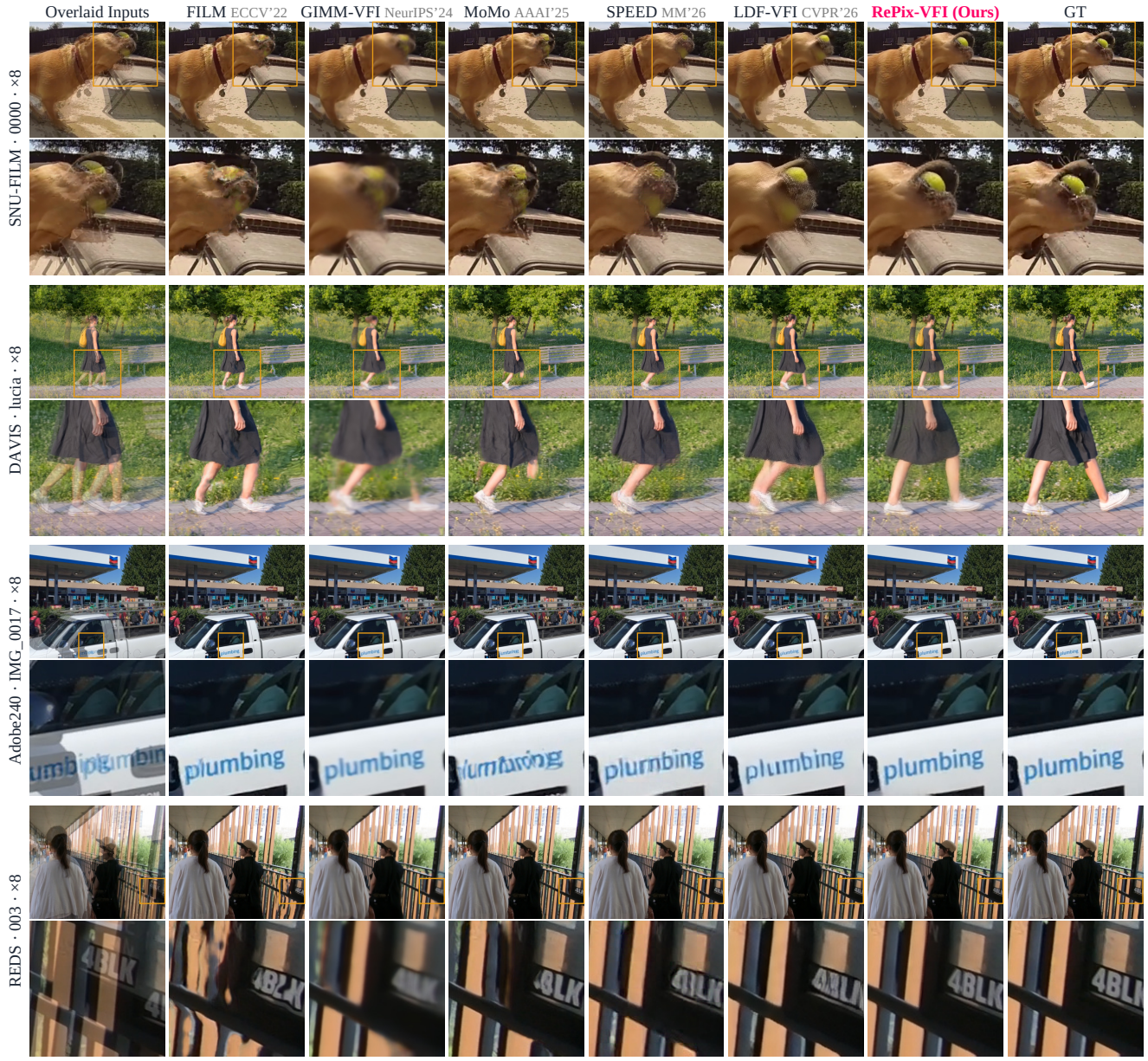

Figure 6. **Qualitative comparisons at $\times$8.** Midpoint targets ($\tau = 0.5$) from SNU-FILM, DAVIS, Adobe240, and REDS, from top to bottom. The first column shows an equal-weight overlay of the bounding observations; subsequent columns show predictions and GT. Boxes indicate the regions enlarged below.

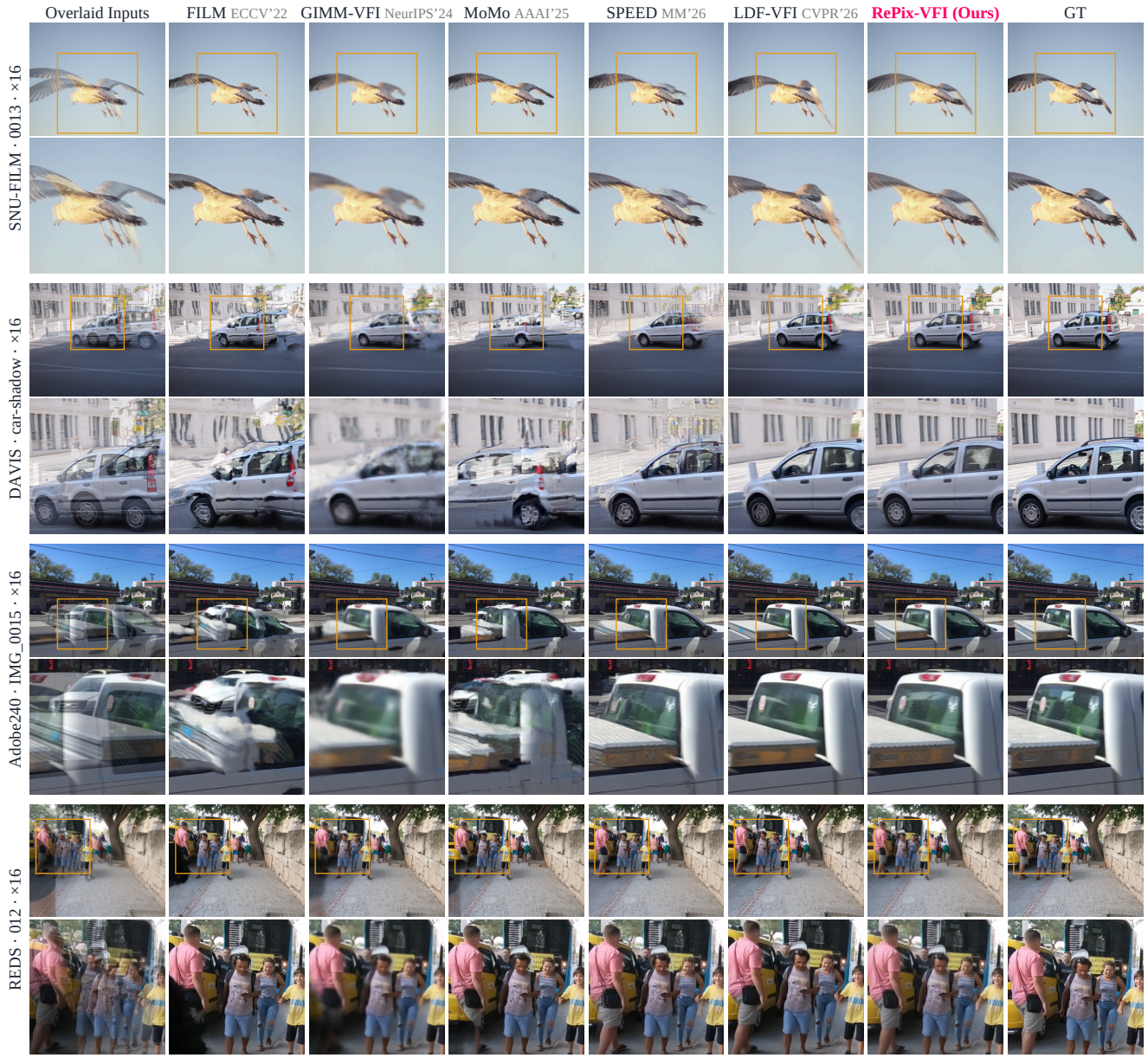

Figure 7. **Qualitative comparisons at ×16.** Midpoint targets ($\tau = 0.5$) from SNU-FILM, DAVIS, Adobe240, and REDS, from top to bottom. The first column shows an equal-weight overlay of the bounding observations; subsequent columns show predictions and GT. Boxes indicate the regions enlarged below.

References

[1] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. VideoINR: Learning video implicit neural representation for continuous space-time super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2047–2057, 2022. 10

[2] Zheng Chen, Zichen Zou, Kewei Zhang, Xiongfei Su, Xin Yuan, Yong Guo, and Yulun Zhang. DOVE: Efficient one-step diffusion model for real-world video super-resolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 85218–85237, 2025. 3

[3] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10663–10671, 2020. 5, 10

[4] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for GAN-based video generation. *ACM Transactions on Graphics*, 39(4), 2020. 5, 9

[5] Duolikun Danier, Fan Zhang, and David Bull. FloLPIPS: A bespoke video quality metric for frame interpolation. In *Picture Coding Symposium (PCS)*, pages 283–287, 2022. 5

[6] Duolikun Danier, Fan Zhang, and David Bull. LDMVFI: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1472–1480, 2024. 2, 3, 6, 8, 11, 12, 13

[7] Ganggui Ding, Xiaogang Xu, Hao Chen, and Chunhua Shen. SpeedVFI: One-step diffusion for efficient video frame interpolation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 25263–25281. PMLR, 2026. 3

[8] Zujin Guo, Wei Li, and Chen Change Loy. Generalizable implicit motion modeling for video frame interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 63747–63770, 2024. 2, 5, 6, 11, 12, 13

[9] Yang Hai, Guo Wang, Tan Su, Wenjie Jiang, and Yinlin Hu. Hierarchical flow diffusion for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22943–22952, 2025. 2, 3

[10] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 4, 5

[11] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision (ECCV)*, pages 624–642, 2022. 2, 6, 11, 12, 13

[12] Zhilin Huang, Yijie Yu, Ling Yang, Chujun Qin, Bing Zheng, Xiawu Zheng, Zikun Zhou, Yaowei Wang, and Wenming Yang. Motion-aware latent diffusion models for video frame interpolation. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 1043–1052, 2024. 3

[13] Jisoo Jeong, Hong Cai, Jamie Menjay Lin, Hanno Ackermann, Hyeonjun Sim, Yinhao Zhu, Yunxiao Shi, and Fatih Porikli. SNM-VFI: Symmetric nonlinear motion-guided generative video frame interpolation. In *European Conference on Computer Vision Workshops (ECCVW)*, 2026. 3

[14] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018. 1, 2

[15] Jaihyun Lew, Jooyoung Choi, Chaehun Shin, Dahuin Jung, and Sungroh Yoon. Disentangled motion modeling for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 4607–4615, 2025. 2, 3, 5, 6, 8, 11, 12, 13

[16] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. AMT: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9801–9810, 2023. 2, 6, 11, 12, 13

[17] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 4

[18] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 4

[19] Zonglin Lyu and Chen Chen. TLB-VFI: Temporal-aware latent Brownian bridge diffusion for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16260–16269, 2025. 6, 8, 11, 12, 13

[20] Zonglin Lyu, Ming Li, Jianbo Jiao, and Chen Chen. Frame interpolation with consecutive Brownian bridge diffusion. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 3449–3458, 2024. 6, 8, 11, 12, 13

[21] Yongrui Ma, Shijie Zhao, Mingde Yao, Junlin Li, Li Zhang, Xiaohong Liu, Qi Dou, Jinwei Gu, and Tianfan Xue. Realtime video frame interpolation using one-step diffusion sampling. In *International Conference on Learning Representations (ICLR)*, 2026. 2, 3

[22] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 10

[23] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5437–5446, 2020. 2, 4

[24] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[25] Xinyu Peng, Han Li, Yuyang Huang, Ziyang Zheng, Yaoming Wang, Xin Chen, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Towards holistic modeling for video frame interpolation with auto-regressive diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 2, 3, 4, 5, 6, 8, 11, 12, 13

[26] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016. 5, 10

[27] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame interpolation for large motion. In *European Conference on Computer Vision (ECCV)*, pages 250–266, 2022. 2, 6, 11, 12, 13

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015. 5

[29] Wonyong Seo, Jihyong Oh, and Munchurl Kim. BiM-VFI: Bidirectional motion field-guided frame interpolation for video with non-uniform motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7244–7253, 2025. 2, 6, 11, 12, 13

[30] Alexandros Stergiou. LAVIB: A large-scale video interpolation benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 29091–29105, 2024. 5, 9

[31] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 10

[32] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 4, 5, 9

[33] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 3, 5, 9

[34] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 10

[35] Jianyi Wang, Shanchuan Lin, Zhijie Lin, Yuxi Ren, Meng Wei, Zongsheng Yue, Shangchen Zhou, Hao Chen, Yang Zhao, Ceyuan Yang, Xuefeng Xiao, Chen Change Loy, and Lu Jiang. SeedVR2: One-step video restoration via diffusion adversarial post-training. In *International Conference on Learning Representations (ICLR)*, pages 40987–41009, 2026. 3

[36] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[37] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8406–8441, 2023. 4

[38] Aiqiu Wu, Zhaofan Qiu, Ting Yao, and Tao Mei. PS-SR: Pseudo-single-step video super-resolution via speculative diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38218–38227, 2026. 3, 4

[39] Chenyang Wu, Jiayi Fu, Chun-Le Guo, Shuhao Han, and Chongyi Li. VTinker: Guided flow upsampling and texture mapping for high-resolution video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10638–10645, 2026. 2, 6, 11, 12, 13

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

[41] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 92529–92553, 2024. 3

[42] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1647–1656, 2019. 3

[43] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6613–6623, 2024. 4

[44] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5682–5692, 2023. 2, 6, 11, 12, 13

[45] Guozhen Zhang, Yuhan Zhu, Yutao Cui, Xiaotong Zhao, Kai Ma, and Limin Wang. Motion-aware generative frame interpolation. *arXiv preprint arXiv:2501.03699*, 2025. 3

[46] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 5, 9

[47] Zihao Zhang, Haoran Chen, Haoyu Zhao, Guansong Lu, Yanwei Fu, Hang Xu, and Zuxuan Wu. EDEN: Enhanced diffusion for high-quality large-motion video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2105–2115, 2025. 6, 8, 11, 12, 13

[48] Zihao Zhang, Haoyu Zhao, Siqian Yang, Yidi Wu, Yudong Jiang, and Zuxuan Wu. SPEED: One-step pixel diffusion for high-quality video frame interpolation. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2026. 3, 6, 8, 11, 12, 13

[49] Zhihang Zhong, Gurunandan Krishnan, Xiao Sun, Yu Qiao, Sizhuo Ma, and Jian Wang. Clearer frames, anytime: Resolving velocity ambiguity in video frame interpolation. In *European Conference on Computer Vision (ECCV)*, pages 346–363, 2024. 6, 11, 12, 13