Controllable Tracking-Based Video Frame Interpolation

KARLIS MARTINS BRIEDIS, DisneyResearch|Studios, Switzerland and ETH Zürich, Switzerland
ABDELAZIZ DJELOUAH, DisneyResearch|Studios, Switzerland
RAPHAËL ORTIZ, DisneyResearch|Studios, Switzerland
MARKUS GROSS, DisneyResearch|Studios, Switzerland and ETH Zürich, Switzerland
CHRISTOPHER SCHROERS, DisneyResearch|Studios, Switzerland

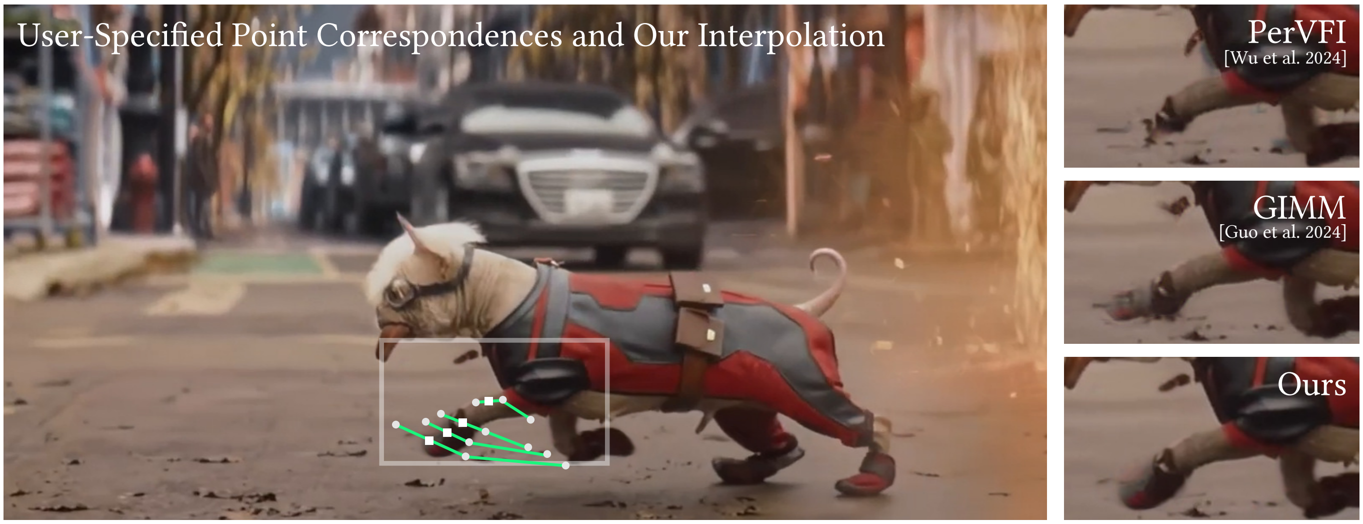

Fig. 1. We present a tracking-based video interpolation method that can take sparse user-specified point tracks as input to improve the interpolation quality and express user's artistic intents. Our method first generates results without any additional inputs and then can be optionally modified to improve interpolation quality. © 2025 Disney

Temporal video frame interpolation has been an active area of research in recent years, with a primary focus on motion estimation, compensation, and synthesis of the final frame. While recent methods have shown good quality results in many cases, they can still fail in challenging scenarios. Moreover, they typically produce fixed outputs with no means of control, further limiting their application in film production pipelines. In this work, we address the less explored problem of user-assisted frame interpolation to improve quality and enable control over the appearance and motion of interpolated frames. To this end, we introduce a tracking-based video frame interpolation method that utilizes sparse point tracks, first estimated and interpolated with existing point tracking methods and then optionally refined by the user. Additionally, we propose a mechanism for controlling the levels of hallucination in interpolated frames through inference-time model weight adaptation, allowing a continuous trade-off between hallucination and blurriness.

Even without any user input, our model achieves state-of-the-art results in challenging test cases. By using points tracked over the whole sequence, we can use better motion trajectory interpolation methods, such as cubic splines, to more accurately represent the true motion and achieve significant improvements in results. Our experiments demonstrate that refining tracks and their trajectories through user interactions significantly improves the quality of interpolated frames.

CCS Concepts: • **Computing methodologies** → *Image processing*; **Reconstruction**.

Additional Key Words and Phrases: Video Frame Interpolation, Point Tracking, User Control

ACM Reference Format:
Karlis Martins Briedis, Abdelaziz Djelouah, Raphaël Ortiz, Markus Gross, and Christopher Schroers. 2025. Controllable Tracking-Based Video Frame Interpolation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3721238.3730598

Authors' Contact Information: Karlis Martins Briedis, karlis.briedis@inf.ethz.ch; Abdelaziz Djelouah, aziz.djelouah@disneyresearch.com; Raphaël Ortiz, raphael.ortiz@disneyresearch.com; Markus Gross, grossm@inf.ethz.ch; Christopher Schroers, christopher.schroers@disneyresearch.com.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730598

1 Introduction

Video frame interpolation (VFI) is a commonly used image post-processing technique with a wide range of applications, such as frame rate adjustment [Castagno et al. 1996], novel-view synthesis [Kalantari et al. 2016], and the generation of artistic slow-motion effects [Jiang et al. 2018].

While the advances made in recent years [Jin et al. 2023; Li et al. 2023; Niklaus and Liu 2020; Zhou et al. 2023] have greatly improved the quality of interpolated frames, finding correspondences in scenes with large or complex displacements between the keyframes and compensating for the motion remains a challenging problem, limiting practical use cases. Additionally, as an ill-posed problem, VFI typically generates a single variant out of many plausible intermediate frames, which may differ from the user expectations. Yet, so far little research has been done in adding control over the interpolated outputs.

On the other hand, significant progress has been made in estimating sparse point correspondences [Luo et al. 2023; Zhang et al. 2024a] and tracking points through a video [Doersch et al. 2023; Karaev et al. 2025; Neoral et al. 2024; Tumanyan et al. 2025; Wang et al. 2023]. Despite this progress, such point tracks have not yet been utilized to improve frame interpolation. Furthermore, frame interpolation methods are typically trained on real-world videos containing various kinds of motion. However, a simple motion model is typically assumed during training, leading to misalignment between the interpolated output and the reference.

In this work, we make the connection between point tracking, and non-linear motion estimation to present a novel tracking-based frame interpolation method, designed around enabling using user control over interpolation outputs. The method uses sparse point tracks as an input, obtains dense flows from keyframes to the target frame, and inverts and refines them into optical flows that are used to synthesize the final frame. The tracks can first be estimated fully automatically with an off-the-shelf tracking algorithm and optionally refined through a user interaction, *e.g.*, to specify correspondences that were missed by the point tracker or to control their trajectories between the keyframes. By training the model with tracks that are estimated from full sequences, including the target frame, we enable it to reconstruct the true motion and avoid temporal misalignment between the model's prediction and the ground truth [Briedis et al. 2021; Kiefhaber et al. 2024; Zhong et al. 2025]. As an additional means of control, we propose an extension to our model to enable test-time trade off between hallucination and blurriness, similar to a low-rank adaptation (LoRA) [Hu et al. 2022b] of the model weights.

Although we focus on adding controllability through point tracks, our base model already achieves competitive performance on the challenging DAVIS dataset. Especially in subjective ratings, our base model excels even when compared to concurrent work. When leveraging point tracks, we can show significant interpolation quality improvements.

To summarize, our main contributions are:

- designing the first frame interpolation architecture that can leverage a set of sparse point tracks for motion estimation, enabling non-linear interpolation during training and inference;
- introducing controllability regarding motion and appearance to help artists address potential imperfections and achieving their artistic intent;
- achieving state-of-the-art frame interpolation results on challenging sequences.

2 Related Work

Classically, frame interpolation has relied on optical flow and image warping [Baker et al. 2011]. Most of the modern learning-based methods build on top of their differential counterparts or estimate the motion implicitly with *phase-based* [Meyer et al. 2018, 2015], *kernel-based* [Lee et al. 2020; Niklaus et al. 2017a,b] or *direction prediction* [Choi et al. 2020] methods. We refer to the survey by Dong et al. [2023] for a more complete list of prior work.

Some of the *motion-based* methods use a pre-trained optical flow estimator [Sun et al. 2018; Teed and Deng 2020] to forward-splat the keyframe features or flow to the target frame [Bao et al. 2019; Hu et al. 2022a; Niklaus et al. 2023; Niklaus and Liu 2018, 2020], or additionally jointly learn the forward motion [Jin et al. 2023]. Other methods predict the motion from the target frame to the keyframes directly to backward-warp them [Huang et al. 2022; Kong et al. 2022; Reda et al. 2022] or combine with forward warping [Danier et al. 2022; Park et al. 2020; Sim et al. 2021]. Our method falls most closely in the last category but instead uses sparse tracks to handle the forward motion while the backward motion is updated in a dense manner at a fixed resolution instead of using coarse-to-fine processing.

More recent paradigms employ the transformer architecture [Lu et al. 2022; Park et al. 2023; Plack et al. 2023; Zhang et al. 2023; Zhou et al. 2023], diffusion models [Danier et al. 2024; Jain et al. 2024], all-pair correlation volumes [Li et al. 2023; Liu et al. 2024], or state space models [Zhang et al. 2024b]. Other works focus on perceptual aspects [Wu et al. 2024], or improving optical flow reversal from the keyframe flows [Guo et al. 2024; Jeong et al. 2024].

Most methods assume linear motion between the keyframes while only a few estimate quadratic motion [Liu et al. 2020; Xu et al. 2019; Zhang et al. 2020] or learn non-linear motion implicitly [Choi et al. 2021; Hu et al. 2024; Kalluri et al. 2023; Park et al. 2021]. While all of these methods can learn a more plausible motion than the linear one, they provide no control and still suffer from misalignment with the reference. Kiefhaber et al. [2024] also address the issue with non-linearities in the frame interpolation training and evaluation data but propose to filter them build a benchmark with only linear motion. Concurrently with our work, Zhong et al. [2025] address training with non-linear data and introduce control over time curves for different segments of an image. User controllability by utilizing conditioned video diffusion models has also been explored [Tanveer et al. 2025; Wang et al. 2025b]. While large generative models perform well at synthesizing new content, they incur very high computational costs and limitations in high-resolution fine-grained detail reconstruction.

The only other existing control for frame interpolation has been proposed as a binary decision between models trained on color or perceptual losses [Niklaus and Liu 2020; Plack et al. 2023; Reda et al. 2022] with no intermediate options. For the task of image upsampling, ESRGAN [Wang et al. 2018] provides control of the appearance using weight interpolation but require interpolating two full sets of weights before inference and does not allow training for intermediate steps. Pan et al. [2023] present an optimization-based method for controlling GAN-generated images using handle points but is limited to images that can be represented in its latent space.

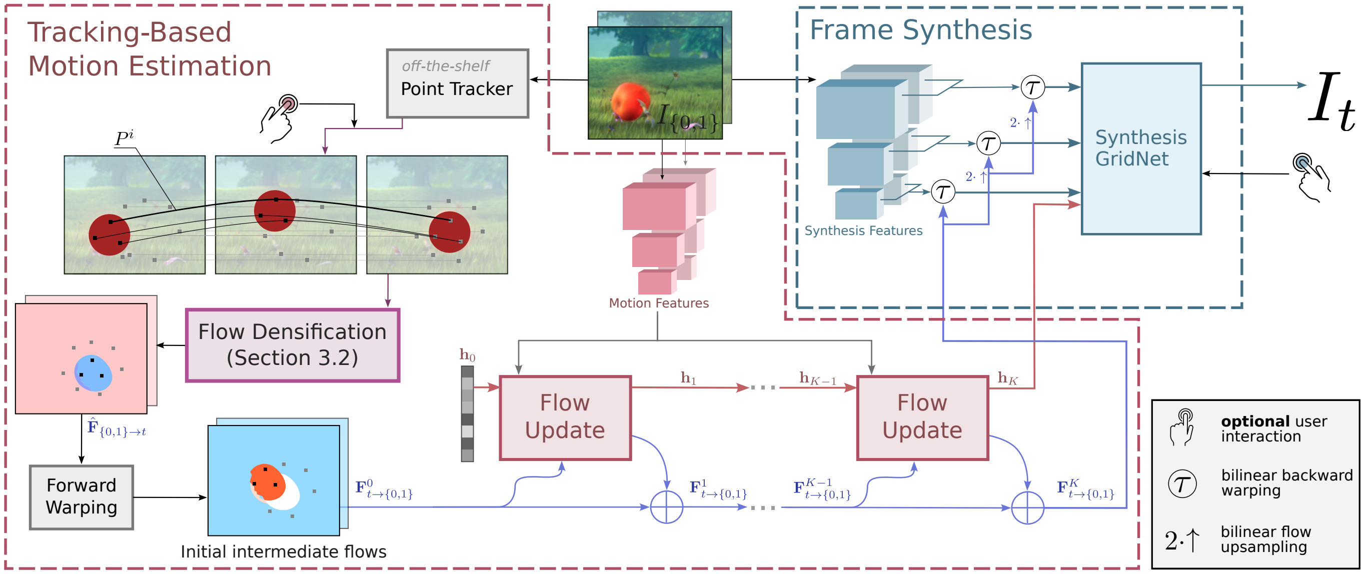

Fig. 2. Method overview. The initial tracks are obtained using an *off-the-shelf* point tracker and optionally adjusted through user interaction, *e.g.* to specify non-linear motion between the keyframes. These tracks are densified at keyframes to obtain approximate forward flows, which are forward-warped to create the initial bilateral flows $\mathbf{F}^0_{t\to\{0,1\}}$. These flows are refined at the target timestep and used to warp keyframe feature pyramids to synthesize the final frame.

Recent point tracking methods [Doersch et al. 2023; Karaev et al. 2025; Neoral et al. 2024; Tumanyan et al. 2025; Wang et al. 2023] have shown great improvement in sparse point tracking but often are prohibitively expensive for dense tracking. Le Moing et al. [2024] propose *dense optical tracking* (DOT), initializing coarse estimate with sparse tracks and refining it with a customized version of the flow estimation model RAFT [Teed and Deng 2020]. As part of our method we solve a related problem of estimating dense flow to an unknown target frame based on sparse point tracks.

3 Tracking-Based Frame Interpolation

The goal of our method is to reconstruct a frame I_t between two keyframes $I_{\{0,1\}}$ by utilizing sparse point tracks extracted from the video or provided as a user input. An overview of the method is shown in Figure 2.

First, we obtain and process points tracks between the input frames. We then use them to compute coarse non-linear optical flows from keyframes the target temporal position, followed by flow reversal and refinement which applies multiple iterations of flow update steps. Finally, we use the refined flows to backward-warp the keyframes and synthesize the final frame.

3.1 Obtaining Point Tracks

At first, we obtain L point tracks, such that l-th track $P^l = \{(\mathbf{x}^l_j, v^l_j) | j \in N\}$ contains the position $\mathbf{x}$ of the same 3D point projected onto the camera in each of the N input frames and $v \in \{0, 1\}$ denotes its visibility. In this work, we consider that the tracks can be obtained automatically using an off-the-shelf method, or refined and provided manually through a user input. For most of our experiments, we automatically extract point tracks using the CoTracker2 [Karaev et al. 2025] method.

To obtain automatically extracted tracks' positions at the target time step t, we linearly interpolate the position of each track. In case the track visibility changes between the two key-frames $\mathbf{I}_{\{0,1\}}$, it is unknown at which intermediate timestep it became occluded. To reflect this, the visibility v for the interpolated track position is set to the minimum of both key point, *i.e.* it is marked as visible only if it is visible in both closest keyframes. Note that as a point can be tracked through the whole video, any discrete higher order point interpolation methods, such as cubic splines, can be used. Additionally, their position and visibility at the target frame can be further adjusted via a user input. Please see the supplementary video for an example how it can be done interactively.

As during training the target frame is known, we extract tracks from all input frames to obtain a better estimation of their position and visibility. It allows to spatially align the outputs with the reference image resulting in a better training supervision signal.

3.2 Flow Densification

Having sparse correspondences from one of the keyframes I_0 to the intermediate target frame I_t and the other keyframe I_1, our goal is to obtain dense flow $\mathbf{F}_{0\to t}$ that follows the given tracks. Figure 3 illustrates the improvements from our densification process. For context, the reference optical flow computed from the ground truth middle frame is shown.

Although naive, spatial nearest-neighbor interpolation of displacements associated with the visible points is a good starting point. More formally, we define this nearest flow field from i to j at pixel $\mathbf{y}$ as

$$\bar{\mathbf{F}}_{i\to j}[\mathbf{y}] = P^{l^*}_j - P^{l^*}_i, \quad l^* = \underset{l\in\{1\ldots L \ | v^l_i\}}{\arg\min} ||P^l_i - \mathbf{y}||_2 . \qquad (1)$$

Fig. 3. Track densification into $\hat{\mathbf{F}}_{0\to t}$. We show the output of the nearest-neighbor and barycentric interpolation approaches compared to our densification method for obtaining the optical flow from a keyframe to the target frame. For reference, we show RAFT output which can not be obtained during inference since the middle frame is unknown.

However, this densification is agnostic to the image content and contains inaccuracies (see Figure 3), even when using more complex approach such as barycentric interpolation.

To improve on this initial result, we want to refine the initial coarse flow estimation $\bar{\mathbf{F}}_{0\to 1}$ by utilizing input frames I_0 and I_1. While any refinement model can be used, for the purposes of our experiments, we leverage DOT [Le Moing et al. 2024], which uses the coarse flow as the initial starting point for a task-adjusted RAFT optical flow model. While we cannot use DOT directly to obtain $\mathbf{F}_{0\to t}$, as I_t is unknown, we employ it to obtain refined keyframe flow $\tilde{\mathbf{F}}_{0\to 1}$ (and $\tilde{\mathbf{F}}_{1\to 0}$ analogously):

$$\tilde{\mathbf{F}}_{0\to 1} = \texttt{DOT}(\bar{\mathbf{F}}_{0\to 1}, I_0, I_1)\ . \tag{2}$$

One can note that on one side, the motion of the tracked points is more reliable, while on the other side the refined keyframe flow better represents the pixel level neighborhood relationship in terms of both content and motion. Our proposal is to leverage both for a better densification. Specifically the similarity in terms of keyframe motion is used to associate pixels and point, to query intermediate flow values. More formally, we define it as:

$$\hat{\mathbf{F}}_{0\to t}[\mathbf{y}] = P_t^{l^*} - P_0^{l^*}\ , \tag{3a}$$

$$l^* = \underset{l\in\mathcal{N}_K^0(\mathbf{y})}{\arg\min}\ |(P_1^l - P_0^l) - \tilde{\mathbf{F}}_{0\to 1}[\mathbf{y}]|\ , \tag{3b}$$

$$\mathcal{N}_K^i(\mathbf{y}) = \left\{k \;\middle|\; k \in \operatorname{argsort}_{l\in\{1\dots L\ |v_i^l\}} ||P_i^l - \mathbf{y}||_2\right\}\ , \tag{3c}$$

where $\mathcal{N}_K^i$ gives the $K = 16$ spatially nearest visible neighbors.

A representation of the these different refinement stages is provided in Figure 4, where we see the transition from the initial set of tracked points, the densification using nearest-neighbors, the refinement of the key-frame flow using DOT, and its usage to query motion vectors from the tracked point to create a better densification for both the target (middle) and key frame.

3.3 Flow Refinement and Frame Synthesis

Having obtained trajectory-adjusted flows $\hat{\mathbf{F}}_{i\to t}$ from the keyframes to the target timestep, we use them for the final frame synthesis. As the densified outputs are still relatively coarse, we choose to refine them by splatting them to the target timestep, applying iterative flow update steps, and, finally, using backward warping to provide keyframe information to the frame synthesis module.

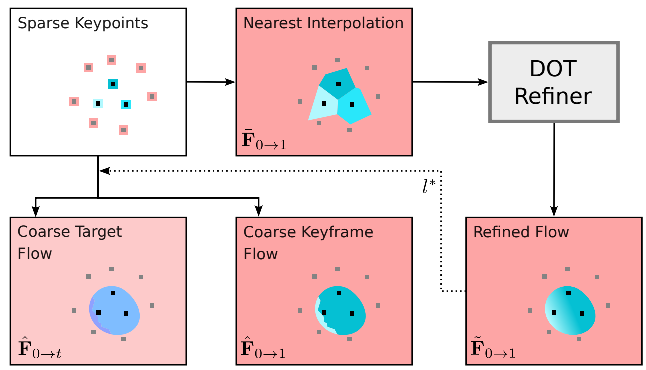

Fig. 4. The different stages for the densification of tracked points, transitioning from the initial set of tracked points to dense target and key frame flows. We use key-frame flow from DOT to query motion vectors from the tracked point to create a better result for both the target (middle) and key frame. See the text (Section 3.2) for more details.

Flow Reversal. To obtain the initial flows $\mathbf{F}^0_{t\to i}$ from the target frame to keyframes, we reverse them with forward warping $\mathcal{W}$:

$$\mathbf{F}^0_{t\to i} = \mathcal{W}_{\hat{\mathbf{F}}_{i\to t}}(-\hat{\mathbf{F}}_{i\to t}\ ,), \tag{4}$$

using exponential weights [Niklaus and Liu 2020]. As we cannot use brightness constancy due to non-linear motion, we opt to use depth-aware weighting [Bao et al. 2019] extracted with the monocular Depth Anything V2 [Yang et al. 2024] model and further refined with a small network together with the input features and flow $\hat{\mathbf{F}}_{i\to t}$.

Flow Refinement. We refine the initial flow fields $\mathbf{F}^0_{t\to\{0,1\}}$ over $K = 4$ iterations into the final flows $\mathbf{F}^K_{t\to\{0,1\}}$, simultaneously solving the interpolation and optical flow problems.

At first, we compute 5-level scale-agnostic feature pyramids [Reda et al. 2022] of the input frames. In every iteration, we backward-warp the bottom 3 levels of scale (1/4, 1/8, 1/16) with the current flow estimates $\mathbf{F}^k_{t\to\{0,1\}}$ and update flows along a hidden state $\mathbf{h}_k$ that is initialized as a learnable vector, repeated across the spatial dimensions.

For the update step, we choose to adopt recurrent update block, re-purposed for multi-level processing [Wang et al. 2025a]. This achieved by starting the processing from the lowest level and concatenating it with the bilinearly upsampled hidden state of the previous level. The flow update is only performed at the 1/4 resolution.

Frame Synthesis. For the final frame synthesis, we construct a pair of feature pyramids and bilinearly backward-warp them with rescaled and bilinearly resampled $\mathbf{F}^K_{t\to\{0,1\}}$. We then apply an occlusion mask to the warped frames of the valid pixels in $\mathbf{F}^0_{t\to\{0,1\}}$. That is, we zero out warped features if not a single value was forward-warped to that pixel.

Warped features, along with the final hidden state $\mathbf{h}_k$ at 1/4 resolution, are concatenated on every level of scale and processed with a 3×6 GridNet [Fourure et al. 2017] to obtain the final interpolated frame.

3.4 Low-Rank Sharpness Adapter

Training with just pixel losses often yields blurry results therefore many methods perform a fine-tuning stage with a perceptual feature loss [Niklaus and Liu 2018, 2020; Niklaus et al. 2017b] as well as style loss [Plack et al. 2023; Reda et al. 2022] to improve the perceptual quality. However, models tuned with such losses can sometimes exhibit artifacts or hallucinate unwanted anomalies such that blurry results can be the preferred behavior. We propose a method extension that allows to control the level of sharpness and hallucination based on low-rank adaption (LoRA) [Hu et al. 2022b].

We first train the model without any perceptual losses and then fine tune only the low rank updates for each convolution [Mangrulkar et al. 2022] of the frame synthesis network lateral blocks while adding VGG feature difference loss [Niklaus and Liu 2018].

The output of each convolution is redefined to

$$y = \Phi(x) + w \cdot \Delta\Phi_r(x) \,, \tag{5a}$$

$$\Delta\Phi_r(x) = K_2^{r\times d} * K_1^{d\times r} * x \,, \tag{5b}$$

where x is the input, Φ is the original convolution and $\Delta\Phi_r$ is a low-rank convolution first mapping inputs to the lower r-dimensional space and then transforming back to the original d-dimensional space, w is the control weight that is uniformly sampled during training $w \sim \mathcal{U}_{[0,1]}$. Both $\Delta\Phi_r$ weights are fine-tuned and the second weight is initialized to as zeros while the original $\Phi(x)$ is frozen.

During inference, we can dynamically change the control weight w to achieve different outputs without retraining the model. Additionally, the weight can be changed spatially to control only some parts of the image. To provide a spatially-varying mask to lower levels of the GridNet we apply average pooling operation.

4 Experiments

Training Details. We train our method on Vimeo-90K [Xue et al. 2019] septuplet dataset. During training, we sample random 256×256 crops from a uniformly-spaced frame triplet. For data augmentation we apply temporal and spatial flips. The model is trained with the Adam [Kingma and Ba 2015] optimizer with batch size of 4 for $500K$ steps. We use the reciprocal square root learning rate schedule [Zhai et al. 2022], performing $100K$ warm up and cooldown steps, with peak learning rate reaching of 10^{-4}. Following Lu et al. [2022], we use L_1 and *Census* losses. It takes approximately 45 hours to train our final model on a single *NVIDIA RTX 4090* GPU using mixed-precision training.

Sharpness LoRA Training. To train the low-rank sharpness adapter, we add perceptual feature loss following [Niklaus and Liu 2018] and train for an additional $200K$ steps using constant learning rate of 10^{-3}. The fine-tuning takes additional 13 hours.

Point Tracking. To obtain the points between the keyframes we use CoTracker2 [Karaev et al. 2025]. For training, we use a single set of pre-generated tracks per sequence, initialized on a regular grid with an edge size of 16 in 3rd and 5th frame and tracked over the whole sequence. During inference, by default we sample 2048 points near motion boundaries similar to Le Moing et al. [2024]. For the user interaction tests we use a regular with an edge size of 32 to have fewer tracks that need to be interacted with.

Evaluation. To evaluate our methods we adopt the commonly used Vimeo-90K test splits and the more challenging DAVIS [Perazzi et al. 2016] dataset at $1080p$ resolution. For DAVIS dataset we interpolate 4-th frame of each of the 50 sequences based on its two neighboring frames. Unless otherwise noted, we use sharpness control value $w = 1.0$.

Runtime and Memory. When measured over the DAVIS test set, tracking 2048 points takes $0.30 \pm 0.00s$, while running the whole interpolation model takes $0.80 \pm 0.06s$. Model inference on a $2K$ dataset uses approximately 8GB of GPU memory, while $4K$ content uses approximately 24GB.

4.1 Comparison with Prior Methods

To evaluate the unassisted baseline performance of our method, *i.e.* without taking any user inputs, we compare it with the traditional state-of-the-art frame interpolation methods. For quantitative comparisons, we re-evaluate the methods with implementations provided by their authors and present the results in Table 1. For *XVFI* we use the variant trained on Vimeo-90K dataset.

A clear benefit from using our method is the possibility to interpolate tracks with cubic splines that represent the motion more accurately. As shown in Table 1 our method shows significant improvement and outperforms prior work by a large margin. Additionally, even our base model shows a strong performance, especially on the more challenging DAVIS dataset, and is on par with the state-of-the art, quantitatively outperforming all prior works apart from the concurrent work GIMM [Guo et al. 2024].

Qualitative comparison with the top-performing two-frame methods is shown in Figure 5. For FILM and our method we show the perceptually trained variants. It can be seen that our method has better quality results when interpolating scenes with complex motion.

4.2 Motion Control Evaluation

To quantitatively evaluate how user-provided correspondence points can improve the interpolation quality, we simulate it by extracting tracks from the sequence, including the target reference frame.

Initially, we extract point tracks between both keyframes as in traditional inference and linearly interpolate them, while also track

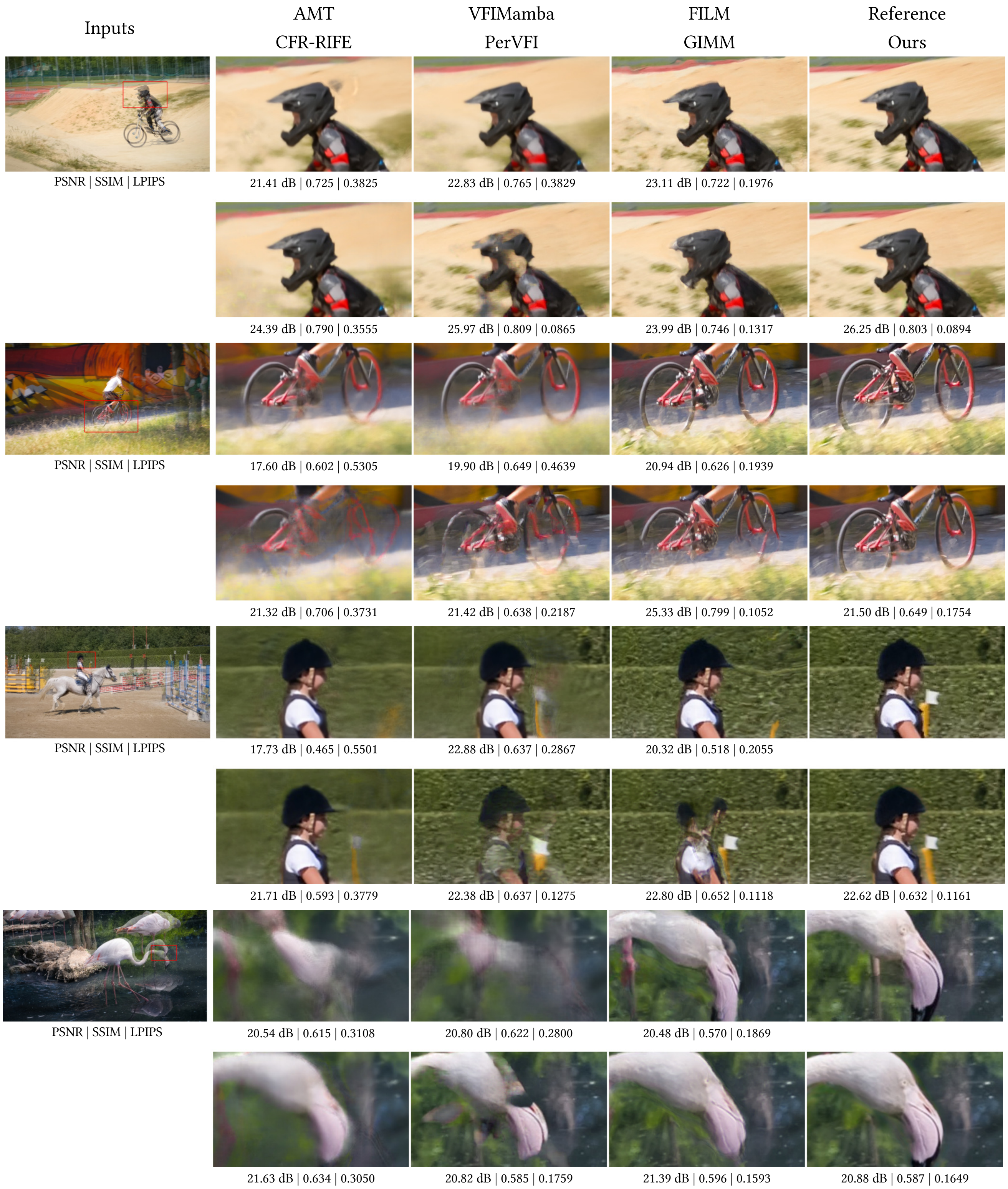

Fig. 5. Qualitative comparison with the prior frame interpolation methods without any user interaction or additional inputs. We show overlaid keyframes as inputs and report metrics per full image.

Table 1. Quantitative evaluation against prior methods, without using any user inputs. We list the methods trained with perceptual losses separately. For our model we report two scores with different sharpness control values $\mathcal{S}_w$. Finally, we report results with non-linear motion estimation from four input frames on the DAVIS dataset. It is not applicable for Vimeo-90K.

	Vimeo-90K 256p			**DAVIS** 1080p		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SoftSplat-$\mathcal{L}_1$ [Niklaus and Liu 2020]	36.09	0.970	0.0220	26.65	0.796	0.1907
XVFI-Vimeo [Sim et al. 2021]	35.06	0.963	0.0234	24.83	0.752	0.2332
ABME [Park et al. 2021]	36.22	0.971	0.0217	27.06	0.811	0.1889
VFIFormer [Lu et al. 2022]	36.55	**0.972**	0.0207	Out of Memory		
RIFE [Huang et al. 2022]	34.28	0.957	0.0192	26.79	0.792	0.1175
FILM-L_1 [Reda et al. 2022]	36.05	0.970	0.0201	27.42	0.811	0.1162
AMT-G [Li et al. 2023]	36.53	**0.972**	0.0195	26.80	0.799	0.1832
UPRNet LARGE [Jin et al. 2023]	36.43	**0.972**	0.0206	25.95	0.782	0.2316
EMA-VFI [Zhang et al. 2023]	**36.65**	**0.972**	0.0205	26.41	0.784	0.2213
SGM 50% [Liu et al. 2024]	35.81	0.968	0.0217	27.14	0.806	0.1760
CFA-RIFE [Zhong et al. 2025]	34.85	0.962	0.0241	27.70	0.823	0.1638
VFIMamba [Zhang et al. 2024b]	36.64	**0.972**	0.0202	27.34	0.814	0.1869
GIMM [Guo et al. 2024]	35.74	0.967	**0.0122**	**28.77**	**0.838**	**0.0738**
Ours-$\mathcal{S}_{0.0}$	35.74	0.968	0.0212	28.16	0.829	0.1176
SoftSplat-$\mathcal{L}_F$ [Niklaus and Liu 2020]	35.45	0.964	**0.0128**	26.20	0.767	0.1337
FILM-$\mathcal{L}_S$ [Reda et al. 2022]	**35.86**	**0.969**	0.0132	27.22	0.802	0.0970
PerVFI [Wu et al. 2024]	34.02	0.954	0.0179	27.38	0.808	0.0912
LDMVFI [Danier et al. 2024]	33.11	0.945	0.0233	24.65	0.727	0.1658
Ours-$\mathcal{S}_{1.0}$	35.49	0.966	0.0142	**27.98**	**0.820**	**0.0839**
FLAVR [Kalluri et al. 2023]		n/a		26.29	0.778	0.2874
Ours-$\mathcal{S}_{0.0}$-cubic splines		n/a		**29.30**	**0.852**	0.1123
Ours-$\mathcal{S}_{1.0}$-cubic splines		n/a		28.95	0.844	**0.0791**

Table 2. Quantitative evaluation of motion control by observing the interpolation improvement depending on the number of reference tracks provided to our method. See the text for more details.

Reference Tracks #	**Vimeo-90K** 256p PSNR ↑	SSIM ↑	LPIPS ↓	**DAVIS** 1080p PSNR ↑	SSIM ↑	LPIPS ↓
0	35.51	0.967	0.0141	27.84	0.820	0.0853
4	35.62	0.967	0.0140	27.91	0.821	0.0840
8	35.72	0.968	0.0139	27.93	0.821	0.0839
16	35.87	0.969	0.0137	27.96	0.822	0.0837
32	36.11	0.970	0.0134	28.05	0.826	0.0831
64	36.46	0.971	0.0130	28.28	0.836	0.0816
128	**36.57**	**0.972**	**0.0129**	28.91	0.860	0.0774
256	**36.57**	**0.972**	**0.0129**	30.24	0.899	0.0699
512	**36.57**	**0.972**	**0.0129**	**31.66**	**0.925**	**0.0645**

the same points across all keyframes. We then evaluate which tracks have the largest error between the true and linearly assumed motion, defined as

$$error_l = ||P_t^l - \frac{P_0^l + P_1^l}{2}||_2 \ . \qquad (6)$$

Subsequently, we select a specified number of reference tracks with the largest error and replace them with their true position in

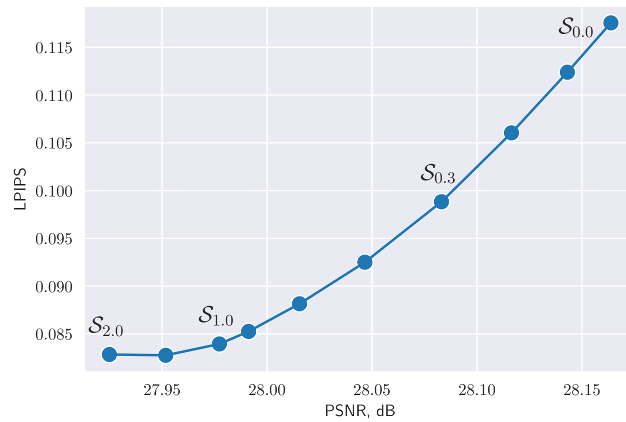

Fig. 6. Sharpness control results. PSNR and LPIPS values for different perceptual control values $\mathcal{S}_w$ over the DAVIS test dataset.

the target frame. This process approximates a scenario where a user notices interpolation errors and corrects them by adjusting nearby tracks.

In Table 2, we show how the number of provided tracks, extracted by using the reference, impacts the final interpolation result on our two benchmarks. It can be observed that by increasing the number of control points, the interpolation quality also improves.

4.3 Sharpness Control Evaluation

Results with two sharpness control values are reported in Table 1. In Figure 6 we show how different control values impact the perception-distortion quality.

4.4 Ablation Study

We present an ablation study in Table 3, evaluating the impact of training data, point densification methods, and network components.

First, we train our model on Vimeo-90K triplet (3F) training dataset as well as the Vimeo-90K septuplet (7F) dataset, adjusting the ratio of tracks with linear assumption. That is, during training with a chosen probability `lin` we replace the true position of all target frame tracks with a linear approximation. These results show the benefits of training with more challenging data, while highlighting the importance of considering the non-linear motion during training.

In the second part, we investigate the impact of different point densification approaches by comparing nearest and barycentric interpolation methods with our algorithm, described in Section 3.2. Additionally, we consider an extension of our algorithm that optimizes continuous blending weights for nearest neighbors and applies them to the target frame tracks.

Finally, we ablate design decisions in our model by training a model without providing depth values to the splatting weight estimation, without masking occluded regions, and using a smaller, efficient model with halved internal layer feature dimensions. While some alternative variants perform better quantitatively, we prioritize the perceptual quality.

Table 3. Ablative experiments on the model design and training data.

		Vimeo-90K 256*p*			DAVIS 1080*p*		
	Scenario	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Training Data	3F	**35.97**	**0.970**	**0.0204**	27.89	0.827	0.1181
	7F, `lin=100%`	35.87	0.967	0.0256	28.63	0.841	0.2109
	7F, `lin=75%`	35.89	0.968	0.0253	28.62	0.842	0.2038
	7F, `lin=50%`	35.93	0.968	0.0254	28.56	0.840	0.2009
	7F, `lin=25%`	35.95	0.968	0.0262	**28.79**	**0.848**	0.1804
	7F, `lin=0%` (Ours)	35.74	0.968	0.0212	28.16	0.829	**0.1176**
Densification	Nearest	35.66	**0.968**	**0.0209**	28.09	0.827	0.1187
	Barycentric	35.53	0.967	0.0217	28.06	0.827	0.1177
	Optimized	**35.74**	**0.968**	0.0211	**28.17**	**0.829**	**0.1175**
	Ours	**35.74**	**0.968**	0.0212	28.16	**0.829**	0.1176
Model	w/o Depth	26.73	0.835	0.1321	25.04	0.751	0.2754
	w/o Occlusion Masking	**35.80**	**0.969**	**0.0209**	28.18	0.829	0.1188
	Smaller Model	35.63	0.968	0.0215	**28.22**	**0.830**	0.1227
	Ours	35.74	0.968	0.0212	28.16	0.829	**0.1176**

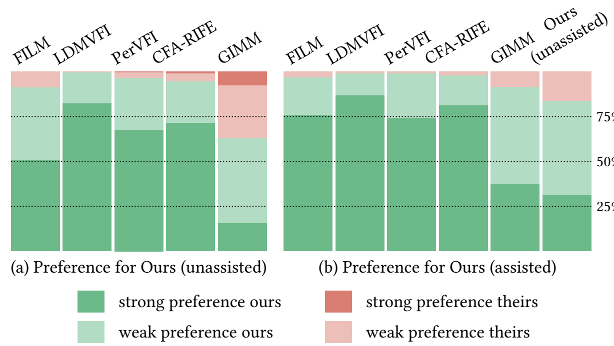

Fig. 7. User study results of comparing our methods - (a) non-assisted and (b) assisted - against prior methods. Dark green and light green represent strong and weak preference for our method, respectively, while dark red and light red indicate strong and weak preference for the compared method.

4.5 Real-World User Controllability

To interact with the model, we developed an interactive desktop application that allows users to load automatically estimated tracks and modify them by adjusting their position and visibility in each keyframe, as well as delete and add new tracks. It allows choosing different point interpolation methods and to specify global *sharpness* weight.

We use this tool to process several sequences from the DAVIS test set and show results in Figure 8. Examples of interaction are shown in the supplementary video.

4.6 User Study

To evaluate the perceptual improvement of our baseline method as well as the impact of user interactions, we conducted a user study where participants had to give a strong or weak preference for one of two 32× interpolated videos. In the study, 26 users provided 1598 votes and the summary of the results is shown in Figure 7.

Our baseline version, without any user inputs, already shows strong results compared to prior art, with only the concurrent GIMM achieving close results. However, our assisted version, obtained by interacting with the method for 6:06 minutes per sequence on average, shows a clear preference in the user ratings, achieving 91.4% preference over the best prior method GIMM, and 83.7% preference over our unassisted version.

For more details on the user study design and results, please see the supplementary document.

5 Discussion and Limitations

Although we propose a user-oriented frame interpolation method that shows strong unassisted interpolation results and is first to enable full control for improving interpolation outputs and motion trajectories, there are still a few limitations and open areas for future work.

As our work prioritizes quality and controllability over computational efficiency, it adds an overhead to the interactive workflow. While we find it generally sufficient to make edits in a low interpolation factor preview and only then rendering the high framerate version only once, the high framerate video can sometimes show problems that are not very apparent in the low-framerate video. Future work on increasing interpolation efficiency without compromising quality could allow to interactively preview the final rendering.

Additionally, more investigation into the graphical interface to interact with the model could improve user efficiency and quality outputs. For example, to change trajectories of an object, all points have to be manually selected. This could be improved by use of segmentation models or other tools to automatically guess the region user might want to edit. There is also currently no control over the depth and occlusion values, which makes some sequences, such as *dog-agility* in results, challenging to improve (Figure 9). Another important interaction aspect is removal of incorrect point matches and specifying new ones. Further advances in point tracking algorithms would alleviate some of this burden.

Finally, as our method is inherently based on explicit flow representations, it can fail to interpolate complex elements where the motion cannot be approximated with a single displacement vector, *e.g.* volumes. Adoption of implicit motion modeling (GIMM) [Guo et al. 2024] for flow reversal could further improve the performance of the model.

6 Conclusion

In this paper we have presented a tracking-based frame interpolation method that utilize sparse point tracks to improve the interpolation quality and enable artist control over interpolation results. Using only point tracks, estimated from the keyframes, our method achieves state-of-the-art results on a challenging test dataset.

Additionally, we have shown that by allowing a user to interact with the model, it allows to improve the quality and significantly outperforms prior methods in user preference.

Fig. 8. Qualitative comparison of frame interpolation before and after user interaction, along with GIMM [Guo et al. 2024] results. We show overlaid keyframes as inputs and report metrics per full image.

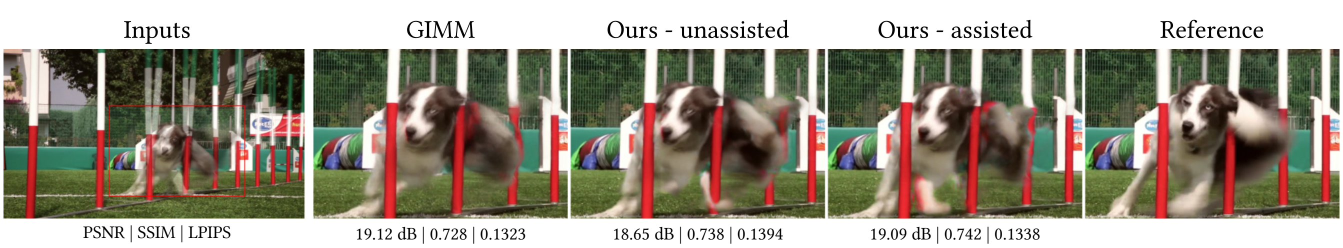

Fig. 9. User control limitation. With no explicit control over depth and occlusions, it is difficult to improve samples with changing depth (left front paw).

Acknowledgments

Figure 2 contains images by Blender Foundation. Figures 3, 5, 8, 9 contain images from the DAVIS dataset [Perazzi et al. 2016].

References

Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. 2011. A Database and Evaluation Methodology for Optical Flow. *International journal of computer vision* 92, 1 (2011), 1–31.

Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. 2019. Depth-Aware Video Frame Interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Karlis Martins Briedis, Abdelaziz Djelouah, Mark Meyer, Ian McGonigal, Markus Gross, and Christopher Schroers. 2021. Neural Frame Interpolation for Rendered Content. *ACM Transactions on Graphics* 40, 6 (Dec. 2021), 239:1–239:13. doi:10.1145/3478513.3480553

R. Castagno, P. Haavisto, and G. Ramponi. 1996. A Method for Motion Adaptive Frame Rate Up-Conversion. *IEEE Transactions on Circuits and Systems for Video Technology* 6, 5 (Oct. 1996), 436–446. doi:10.1109/76.538926

Jinsoo Choi, Jaesik Park, and In So Kweon. 2021. High-Quality Frame Interpolation via Tridirectional Inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 596–604.

Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. 2020. Channel Attention Is All You Need for Video Frame Interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, Number 07. 10663–10671.

Duolikun Danier, Fan Zhang, and David Bull. 2022. ST-MFNet: A Spatio-Temporal Multi-Flow Network for Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3521–3531.

Duolikun Danier, Fan Zhang, and David Bull. 2024. LDMVFI: Video Frame Interpolation with Latent Diffusion Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 2 (March 2024), 1472–1480. doi:10.1609/aaai.v38i2.27912

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. 2023. Tapir: Tracking Any Point with per-Frame Initialization and Temporal Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10061–10072.

Jiong Dong, Kaoru Ota, and Mianxiong Dong. 2023. Video Frame Interpolation: A Comprehensive Survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 19, 2s (May 2023), 78:1–78:31. doi:10.1145/3556544

Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. 2017. Residual Conv-Deconv Grid Network for Semantic Segmentation. In *Proceedings of the British Machine Vision Conference, 2017*.

Zujin Guo, Wei Li, and Chen Change Loy. 2024. Generalizable Implicit Motion Modeling for Video Frame Interpolation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022b. LoRA: Low-rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Mengshun Hu, Kui Jiang, Zhihang Zhong, Zheng Wang, and Yinqiang Zheng. 2024. IQ-VFI: Implicit Quadratic Motion Estimation for Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6410–6419.

Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. 2022a. Many-to-Many Splatting for Efficient Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. 2022. Real-Time Intermediate Flow Estimation for Video Frame Interpolation. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Vol. 13674. Springer Nature Switzerland, Cham, 624–642. doi:10.1007/978-3-031-19781-9_36

Siddhant Jain, Daniel Watson, Eric Tabellion, Aleksander Holynski, Ben Poole, and Janne Kontkanen. 2024. Video Interpolation with Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7341–7351.

Jisoo Jeong, Hong Cai, Risheek Garrepalli, Jamie Menjay Lin, Munawar Hayat, and Fatih Porikli. 2024. OCAI: Improving Optical Flow Estimation by Occlusion and Consistency Aware Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19352–19362.

Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. 2018. Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9000–9008.

Xin Jin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm. 2023. A Unified Pyramid Recurrent Network for Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1578–1587.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-Based View Synthesis for Light Field Cameras. *ACM Transactions on Graphics* 35, 6 (Dec. 2016), 193:1–193:10. doi:10.1145/2980179.2980251

Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. 2023. FLAVR: Flow-Agnostic Video Representations for Fast Frame Interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2071–2082.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. 2025. CoTracker: It Is Better to Track Together. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 18–35.

Simon Kiefhaber, Simon Niklaus, Feng Liu, and Simone Schaub-Meyer. 2024. Benchmarking Video Frame Interpolation. doi:10.48550/arXiv.2403.17128 arXiv:2403.17128 [cs]

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA, USA.

Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. 2022. IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. 2024. Dense Optical Tracking: Connecting the Dots. In *CVPR*.

Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. 2020. AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5316–5325.

Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. 2023. AMT: All-Pairs Multi-Field Transforms for Efficient Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9801–9810.

Chunxu Liu, Guozhen Zhang, Rui Zhao, and Limin Wang. 2024. Sparse Global Matching for Video Frame Interpolation with Large Motion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 19125–19134. doi:10.1109/CVPR52733.2024.01809

Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. 2020. Enhanced Quadratic Video Interpolation. In *Computer Vision – ECCV 2020 Workshops*, Adrien Bartoli and Andrea Fusiello (Eds.). Springer International Publishing, Cham, 41–56. doi:10.1007/978-3-030-66823-5_3

Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. 2022. Video Frame Interpolation With Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3532–3542.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. 2023. Diffusion Hyperfeatures: Searching through Time and Space for Semantic Correspondence. In *Advances in Neural Information Processing Systems*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. 2018. PhaseNet for Video Frame Interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. 2015. Phase-Based Frame Interpolation for Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1410–1418.

Michal Neoral, Jonáš Šerých, and Jiří Matas. 2024. MFT: Long-Term Tracking of Every Pixel. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 6837–6847.

Simon Niklaus, Ping Hu, and Jiawen Chen. 2023. Splatting-Based Synthesis for Video Frame Interpolation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 713–723.

Simon Niklaus and Feng Liu. 2018. Context-Aware Synthesis for Video Frame Interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1701–1710.

Simon Niklaus and Feng Liu. 2020. Softmax Splatting for Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5437–5446.

Simon Niklaus, Long Mai, and Feng Liu. 2017a. Video Frame Interpolation via Adaptive Convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 670–679.

Simon Niklaus, Long Mai, and Feng Liu. 2017b. Video Frame Interpolation via Adaptive Separable Convolution. In *Proceedings of the IEEE International Conference on Computer Vision*. 261–270.

Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. 2023. Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*. ACM, Los Angeles

CA USA, 1–11. doi:10.1145/3588432.3591500

Junheum Park, Jintae Kim, and Chang-Su Kim. 2023. BiFormer: Learning Bilateral Motion Estimation via Bilateral Transformer for 4K Video Frame Interpolation. In *Computer Vision and Pattern Recognition*.

Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. 2020. BMBC: Bilateral Motion Estimation with Bilateral Cost Volume for Video Interpolation. In *European Conference on Computer Vision*.

Junheum Park, Chul Lee, and Chang-Su Kim. 2021. Asymmetric Bilateral Motion Estimation for Video Frame Interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14539–14548.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. 2016. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *Computer Vision and Pattern Recognition*.

Markus Plack, Karlis Martins Briedis, Abdelaziz Djelouah, Matthias B. Hullin, Markus Gross, and Christopher Schroers. 2023. Frame Interpolation Transformer and Uncertainty Guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9811–9821.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. 2022. FILM: Frame Interpolation for Large Motion. In *European Conference on Computer Vision (ECCV)*.

Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. 2021. XVFI: Extreme Video Frame Interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14489–14498.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. PWC-Net: Cnns for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8934–8943.

Maham Tanveer, Yang Zhou, Simon Niklaus, Ali Mahdavi Amiri, Hao Zhang, Krishna Kumar Singh, and Nanxuan Zhao. 2025. MotionBridge: Dynamic Video Inbetweening with Flexible Controls. doi:10.48550/arXiv.2412.13190 arXiv:2412.13190 [cs]

Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 402–419. doi:10.1007/978-3-030-58536-5_24

Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. 2025. DINO-tracker: Taming DINO for Self-Supervised Point Tracking in a Single Video. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 367–385.

Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. 2023. Tracking Everything Everywhere All at Once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19795–19806.

Wen Wang, Qiuyu Wang, Kecheng Zheng, Hao OUYANG, Zhekai Chen, Biao Gong, Hao Chen, Yujun Shen, and Chunhua Shen. 2025b. Framer: Interactive Frame Interpolation. In *The Thirteenth International Conference on Learning Representations*.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

Yihan Wang, Lahav Lipson, and Jia Deng. 2025a. SEA-RAFT: Simple, Efficient, Accurate RAFT for Optical Flow. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Vol. 15065. Springer Nature Switzerland, Cham, 36–54. doi:10.1007/978-3-031-72667-5_3

Guangyang Wu, Xin Tao, Changlin Li, Wenyi Wang, Xiaohong Liu, and Qingqing Zheng. 2024. Perception-Oriented Video Frame Interpolation via Asymmetric Blending. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2753–2762.

Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. 2019. Quadratic Video Interpolation. In *NeurIPS*.

Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. 2019. Video Enhancement with Task-Oriented Flow. *International Journal of Computer Vision (IJCV)* 127, 8 (2019), 1106–1125.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything V2. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2022. Scaling Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12104–12113.

Guozhen Zhang, Chunxu Liu, Yutao Cui, Xiaotong Zhao, Kai Ma, and Limin Wang. 2024b. VFIMamba: Video Frame Interpolation with State Space Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. 2023. 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)*. 5682–5692.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. 2024a. Telling Left from Right: Identifying Geometry-Aware Semantic Correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3076–3085.

Youjian Zhang, Chaoyue Wang, and Dacheng Tao. 2020. Video Frame Interpolation without Temporal Priors. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 13308–13318.

Zhihang Zhong, Gurunandan Krishnan, Xiao Sun, Yu Qiao, Sizhuo Ma, and Jian Wang. 2025. Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Vol. 15091. Springer Nature Switzerland, Cham, 346–363. doi:10.1007/978-3-031-73414-4_20

Kun Zhou, Wenbo Li, Xiaoguang Han, and Jiangbo Lu. 2023. Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22169–22179.