Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation

Zhihang Zhong[1,⋆] Gurunandan Krishnan[2] Xiao Sun[1] Yu Qiao[1]
Sizhuo Ma[2,†] Jian Wang[2,†]

[1] Shanghai Artificial Intelligence Laboratory
[2] Snap Inc.

Abstract. Existing video frame interpolation (VFI) methods blindly predict where each object is at a specific timestep t ("time indexing"), which struggles to predict precise object movements. Given two images of a baseball, there are infinitely many possible trajectories: accelerating or decelerating, straight or curved. This often results in blurry frames as the method averages out these possibilities. Instead of forcing the network to learn this complicated time-to-location mapping implicitly together with predicting the frames, we provide the network with an explicit hint on how far the object has traveled between start and end frames, a novel approach termed "distance indexing". This method offers a clearer learning goal for models, reducing the uncertainty tied to object speeds. We further observed that, even with this extra guidance, objects can still be blurry especially when they are equally far from both input frames (*i.e.*, halfway in-between), due to the directional ambiguity in long-range motion. To solve this, we propose an iterative reference-based estimation strategy that breaks down a long-range prediction into several short-range steps. When integrating our plug-and-play strategies into state-of-the-art learning-based models, they exhibit markedly sharper outputs and superior perceptual quality in arbitrary time interpolations, using a uniform distance indexing map in the same format as time indexing. Additionally, distance indexing can be specified pixel-wise, which enables temporal manipulation of each object independently, offering a novel tool for video editing tasks like re-timing. The code is available at https://zzh-tech.github.io/InterpAny-Clearer/.

Keywords: Frame interpolation · Manipulation · Velocity ambiguity

1 Introduction

Video frame interpolation (VFI) plays a crucial role in creating slow-motion videos [1], video generation [8], prediction [44], and compression [43]. Directly warping the starting and ending frames using the optical flow between them can only model linear motion, which often diverges from actual motion paths,

⋆ First author, †Co-corresponding authors. This work was partially completed while Z. Zhong was a Snap Research intern (and a student at The University of Tokyo).

leading to artifacts such as holes. To solve this, learning-based methods have emerged as leading solutions to VFI, which aim to develop a model, represented as $\mathcal{F}$, that uses a starting frame I_0 and an ending frame I_1 to generate a frame for a given timestep, described by:

$$I_t = \mathcal{F}\left(I_0, I_1, t\right). \tag{1}$$

Two paradigms have been proposed: In fixed-time interpolation [1,23], the model only takes the two frames as input and always tries to predict the frame at $t = 0.5$. In arbitrary-time interpolation [11, 14], the model is further given a user-specified timestep $t \in [0, 1]$, which is more flexible at predicting multiple frames in-between.

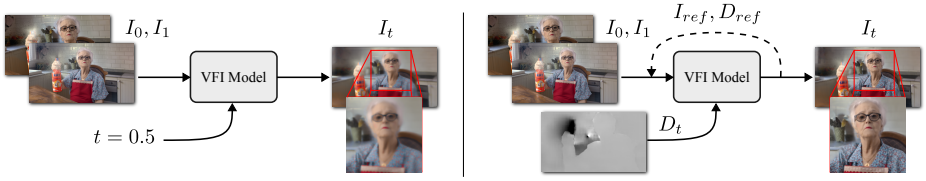
Yet, in both cases, the unsampled blank between the two frames, such as the motion between a ball's starting and ending points, presents infinite possibilities. The velocities of individual objects within these frames remain undefined, introducing a *velocity ambiguity*, a myriad of plausible time-to-location mappings during training. We observed that velocity ambiguity is a primary obstacle hindering the advancement of learning-based VFI: Models trained using aforementioned *time indexing* receive identical inputs with differing supervision signals during training. As a result, they tend to produce blurred and imprecise interpolations, as they average out the potential outcomes.

Could an alternative indexing method minimize such conflicts? One straightforward option is to provide the optical flow at the target timestep as an explicit hint on object motion. However, this information is unknown at inference time, which has to be approximated by the optical flow between I_0 and I_1, scaled by the timestep. This requires running optical flow estimation on top of VFI, which may increase the computational complexity and enforce the VFI algorithm to rely on the explicitly computed but approximate flow. Instead, we propose a more flexible *distance indexing* approach. In lieu of an optical flow map, we employ a *distance ratio* map D_t, where each pixel denotes *how far the object has traveled between start and end frames*, within a normalized range of $[0, 1]$,

$$\boxed{I_t = \mathcal{F}\left(I_0, I_1, \text{explicit motion hint}\right) \quad \Rightarrow \quad I_t = \mathcal{F}\left(I_0, I_1, D_t\right).} \tag{2}$$

During training, D_t is derived from optical flow ratios computed from ground-truth frames. During inference, it is sufficient to provide a uniform map as input, in the exactly same way as time indexing methods, *i.e.*, $D_t(x, y) = t$, $\forall x, y$. However, the semantics of this indexing map have shifted from an uncertain timestep map to a more deterministic motion hint. Through distance indexing, we effectively solve the one-to-many time-to-position mapping dilemma, fostering enhanced convergence and interpolation quality.

Although distance indexing addresses the scalar *speed ambiguity*, the *directional ambiguity* of motion remains a challenge. We observed that this directional uncertainty is most pronounced when situated equally far from the two input frames, *i.e.*, halfway between them. Drawing inspiration from countless computer vision algorithms that iteratively solve a difficult problem (*e.g.*, optical



(a) Training paradigm of time indexing (b) Training paradigm of distance indexing

Fig. 1: Comparison of time indexing and distance indexing training paradigms. (a) Time indexing uses the starting frame I_0, ending frame I_1, and a scalar variable t as inputs. (b) Distance indexing replaces the scalar with a distance map D_t and optionally incorporates iterative reference-based estimation (I_{ref}, D_{ref}) to address velocity ambiguity, resulting in a notably sharper prediction.

flow [38] and image generation [32]), we introduce an iterative reference-based estimation strategy. This strategy seeks to mitigate directional ambiguity by incrementally estimating distances, beginning with nearby points and advancing to farther ones, such that the uncertainty within each step is minimized and the image quality is further improved.

Our approach addresses challenges that are not bound to specific network architectures. Indeed, it can be applied as a plug-and-play strategy that requires only modifying the input channels for each model, as demonstrated in Fig. 1. We conducted extensive experiments on four existing VFI methods to validate the effectiveness of our approach, which produces frames of markedly improved perceptual quality. Moreover, instead of using a uniform map, it is also possible to use a spatially-varying 2D map as input to manipulate the motion of objects. Paired with state-of-the-art segmentation models such as Segment Anything Model (SAM) [17], this empowers users to freely control the interpolation of any object, *e.g.*, making certain objects backtrack in time.

In summary, our key contributions are: 1) Proposing distance indexing and iterative reference-based estimation to address the velocity ambiguity and enhance the capabilities of arbitrary time interpolation models; 2) Conducting comprehensive validation of the efficacy of our plug-and-play strategies across a range of state-of-the-art learning-based models. 3) Presenting an unprecedented manipulation method that allows for customized interpolation of any object.

2 Related Work

2.1 Video frame interpolation

General overview. Numerous VFI solutions rely on optical flows to predict latent frames. Typically, these methods warp input frames forward or backward using flow calculated by off-the-shelf networks like [6, 12, 37, 38] or self-contained flow estimators like [11, 21, 50]. Networks then refine the warped frame to improve visual quality. SuperSlomo [14] uses a linear combination of bi-directional flows

for intermediate flow estimation and backward warping. DAIN [1] introduces a depth-aware flow projection layer for advanced intermediate flow estimation. AdaCoF [19] estimates kernel weights and offset vectors for each target pixel, while BMBC [29] and ABME [30] refine optical flow estimation. Large motion interpolation is addressed by XVFI [34] through a recursive multi-scale structure. VFIFormer [24] employs Transformers to model long-range pixel correlations. IFRNet [18], RIFE [11], and UPR-Net [15] employ efficient pyramid network designs for high-quality, real-time interpolation, with IFRNet and RIFE using leakage distillation losses for flow estimation. Recently, more advanced network modules and operations are proposed to push the upper limit of VFI performance, such as the transformer-based bilateral motion estimator of BiFormer [28], a unifying operation of EMA-VFI [50] to explicitly disentangle motion and appearance information, and bi-directional correlation volumes for all pairs of pixels of AMT [21]. On the other hand, SoftSplat [26] and M2M [10] actively explore the forward warping operation for VFI.

Other contributions to VFI come from various perspectives. For instance, Xu *et al.* [9,46] leverage acceleration information from nearby frames, VideoINR [3] is the first to employ an implicit neural representation, and Lee *et al.* [20] explore and address discontinuity in video frame interpolation using figure-text mixing data augmentation and a discontinuity map. Flow-free approaches have also attracted interest. SepConv [27] integrates motion estimation and pixel synthesis, CAIN [5] employs the PixelShuffle operation with channel attention, and FLAVR [16] utilizes 3D space-time convolutions. Additionally, specialized interpolation methods for anime, which often exhibit minimal textures and exaggerated motion, are proposed by AnimeInterp [35] and Chen *et al.* [2]. On the other hand, motion induced blur [33, 52, 54], shutter mode [7, 13, 53], and event camera [22, 39] are also exploited to achieve VFI.

Learning paradigms. One major thread of VFI methods train networks on triplet of frames, always predicting the central frame. Iterative estimation is used for interpolation ratios higher than $\times 2$. This *fixed-time* method often accumulates errors and struggles with interpolating at arbitrary continuous timesteps. Hence, models like SuperSloMo [14], DAIN [1], BMBC [29], EDSC [4], RIFE [11], IFRNet [18], EMA-VFI [50], and AMT [21] have adopted an *arbitrary time* interpolation paradigm. While theoretically superior, the arbitrary approach faces challenges of more complicated time-to-position mappings due to the velocity ambiguity, resulting in blurred results. This study addresses velocity ambiguity in arbitrary time interpolation and offers solutions.

Prior work by Zhou *et al.* [55] identified motion ambiguity and proposed a texture consistency loss to implicitly ensure interpolated content resemblance to given frames. In contrast, we explicitly address velocity ambiguity and propose solutions. These innovations not only enhance the performance of arbitrary time VFI models but also offer advanced manipulation capabilities.

Segment anything The emergence of Segment Anything Model (SAM) [17] has marked a significant advancement in the realm of zero-shot segmentation, enabling numerous downstream applications including video tracking and segmen-

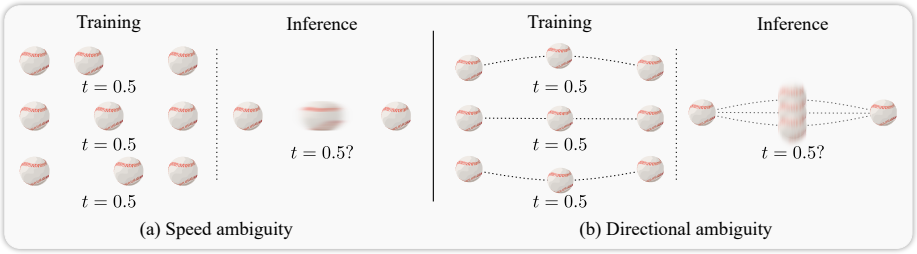


Fig. 2: Velocity ambiguity. (a) Speed ambiguity. (b) Directional ambiguity.

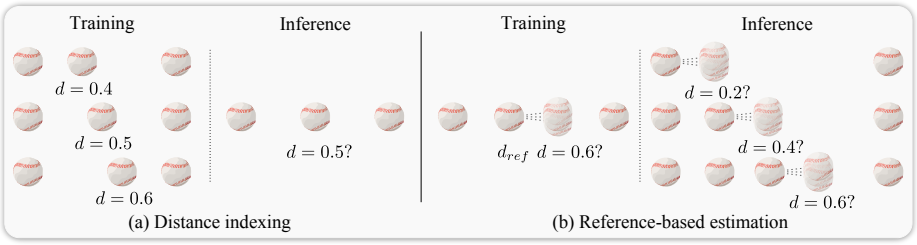


Fig. 3: Disambiguation strategies for velocity ambiguity. (a) Distance indexing. (b) Iterative reference-based estimation.

tation [48], breakthrough mask-free inpainting techniques [49], and interactive image description generation [40]. By specifying the distance indexing individually for each segment, this work introduces a pioneering application to this growing collection: Manipulated Interpolation of Anything.

3 Velocity Ambiguity

In this section, we begin by revisiting the time indexing paradigm. We then outline the associated velocity ambiguity, which encompasses both speed and directional ambiguities.

Fig. 2 (a) shows the example of a horizontally moving baseball. Given a starting frame and an ending frame, along with a time indexing variable $t = 0.5$, the goal of a VFI model is to predict a latent frame at this particular timestep, in accordance with Eq. (1).

Although the starting and ending positions of the baseball are given, its location at $t = 0.5$ remains ambiguous due to an unknown speed distribution: The ball can be accelerating or decelerating, resulting in different locations. This ambiguity introduces a challenge in model training as it leads to multiple valid supervision targets for the identical input. Contrary to the deterministic scenario illustrated in Eq. (1), the VFI function $\mathcal{F}$ is actually tasked with generating a

distribution of plausible frames from the same input frames and time indexing. This can be expressed as:

$$\left\{I_t^1, I_t^2, \ldots, I_t^n\right\} = \mathcal{F}(I_0, I_1, t), \tag{3}$$

where n is the number of plausible frames. Empirically, the model, when trained with this ambiguity, tends to produce a weighted average of possible frames during inference. While this minimizes the loss during training, it results in blurry frames that are perceptually unsatisfying to humans, as shown in Fig. 1 (a). This blurry prediction $\hat{I}_t$ can be considered as an average over all the possibilities if an L_2 loss is used:

$$\hat{I}_t = \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)}[I_t]. \qquad \text{(See details in supplementary materials)} \tag{4}$$

For other losses, Eq. (4) no longer holds, but we empirically observe that the model still learns an aggregated mixture of training frames which results in blur (RIFE [11] and EMA-VFI [50]: Laplacian loss, *i.e.*, L1 loss between the Laplacian pyramids of image pairs; IFRNet [18] and AMT [21]: Charbonnier loss).

Indeed, not only the speed but also the direction of motion remains indeterminate, leading to what we term as "directional ambiguity." This phenomenon is graphically depicted in Fig. 2 (b). This adds an additional layer of complexity in model training and inference. We collectively refer to speed ambiguity and directional ambiguity as velocity ambiguity.

So far, we have been discussing the ambiguity for the fixed time interpolation paradigm, in which t is set by default to 0.5. For arbitrary time interpolation, the ambiguity becomes more pronounced: Instead of predicting a single timestep, the network is expected to predict a continuum of timesteps between 0 and 1, each having a multitude of possibilities. This further complicates learning. Moreover, this ambiguity is sometimes referred to as *mode averaging*, which has been studied in other domains [41]. See supplemental materials for details.

4 Disambiguation Strategies

In this section, we introduce two innovative strategies, namely distance indexing and iterative reference-based estimation, aimed at addressing the challenges posed by the velocity ambiguity. Designed to be plug-and-play, these strategies can be seamlessly integrated into any existing VFI models without necessitating architectural modifications, as shown in Fig. 1 (b).

In traditional time indexing, models intrinsically deduce an uncertain time-to-location mapping, represented as $\mathcal{D}$:

$$I_t = \mathcal{F}(I_0, I_1, \mathcal{D}(t)). \tag{5}$$

This brings forth the question: Can we guide the network to interpolate more precisely without relying on the ambiguous mapping $\mathcal{D}(t)$ to decipher it independently? To address this, we introduce a strategy to diminish speed uncertainty

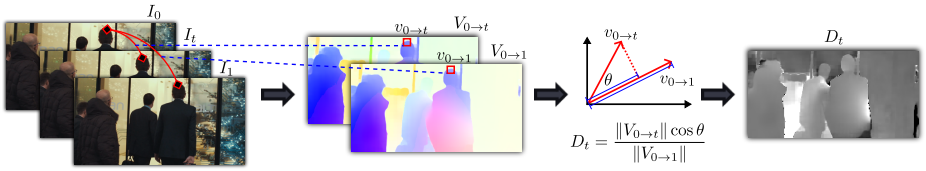


Fig. 4: Calculation of distance map for distance indexing. $V_{0\to t}$ is the estimated optical flow from I_0 to I_t by RAFT [38], and $V_{0\to 1}$ is the optical flow from I_0 to I_1.

by directly specifying a distance ratio map (D_t) instead of the uniform timestep map. This is termed as distance indexing. Consequently, the model sidesteps the intricate process of deducing the time-to-location mapping:

$$I_t = \mathcal{F}(I_0, I_1, D_t). \tag{6}$$

4.1 Distance indexing

We utilize an off-the-shelf optical flow estimator, RAFT [38], to determine the pixel-wise distance map, as shown in Fig. 4. Given an image triplet $\{I_0, I_1, I_t\}$, we first calculate the optical flow from I_0 to I_t, denoted as $\mathbf{V}_{0\to t}$, and from I_0 to I_1 as $\mathbf{V}_{0\to 1}$. At each pixel (x, y), we project the motion vector $\mathbf{V}_{0\to t}(x, y)$ onto $\mathbf{V}_{0\to 1}(x, y)$. The distance map is then defined as the ratio between the projected $\mathbf{V}_{0\to t}(x, y)$ and $\mathbf{V}_{0\to 1}(x, y)$:

$$D_t(x, y) = \frac{\|\mathbf{V}_{0\to t}(x, y)\| \cos\theta}{\|\mathbf{V}_{0\to 1}(x, y)\|}, \tag{7}$$

where θ denotes the angle between the two. By directly integrating D_t, the network achieves a clear comprehension of distance during its training phase, subsequently equipping it to yield sharper frames during inference, as showcased in Fig. 3 (a).

During inference, the algorithm does not have access to the exact distance map since I_t is unknown. In practice, we notice it is usually sufficient to provide a uniform map $D_t = t$, similar to time indexing. Physically this encourages the model to move each object at constant speeds along their trajectories. We observe that constant speed between frames is a valid approximation for many real-world situations. In Sec. 5, we show that even though this results in pixel-level misalignment with the ground-truth frames, it achieves significantly higher perceptual scores and is strongly preferred in the user study. Precise distance maps can be computed from multiple frames, which can potentially further boost the performance. See a detailed discussion in supplementary materials.

4.2 Iterative reference-based estimation

While distance indexing addresses speed ambiguity, it omits directional information, leaving directional ambiguity unresolved. Our observations indicate that,

even with distance indexing, frames predicted at greater distances from the starting and ending frames remain not clear enough due to this ambiguity. To address this, we propose an iterative reference-based estimation strategy, which divides the complex interpolation for long distances into shorter, easier steps. This strategy enhances the traditional VFI function, $\mathcal{F}$, by incorporating a reference image, I_{ref}, and its corresponding distance map, D_{ref}. Specifically, the network now takes the following channels as input:

$$I_t = \mathcal{F}(I_0, I_1, D_t, I_{\text{ref}}, D_{\text{ref}}). \tag{8}$$

In the general case of N steps, the process of iteration is as follows:

$$I_{(i+1)t/N} = \mathcal{F}(I_0, I_1, D_{(i+1)t/N}, I_{t/N}, D_{t/N}), \tag{9}$$

where $i \in \{0, 1, \ldots, N-1\}$. For example, if we break the estimation of of a remote step t into two steps:

$$I_{t/2} = \mathcal{F}(I_0, I_1, D_{t/2}, I_0, D_0), \tag{10}$$

$$I_t = \mathcal{F}(I_0, I_1, D_t, I_{t/2}, D_{t/2}). \tag{11}$$

Importantly, in every iteration, we consistently use the starting and ending frames as reliable appearance references, preventing divergence of uncertainty in each step. By dividing a long step into shorter steps, the uncertainty in each step is reduced, as shown in Fig. 3 (b). While fixed time models also employ an iterative method in a bisectioning way during inference, our strategy progresses from near to far, ensuring more deterministic trajectory interpolation. This reduces errors and uncertainties tied to a single, long-range prediction. **See more on the rationale for solving ambiguities in supplemental materials.**

5 Experiment

5.1 Implementation

We leveraged the plug-and-play nature of our distance indexing and iterative reference-based estimation strategies to seamlessly integrate them into influential arbitrary time VFI models such as RIFE [11] and IFRNet [18], and state-of-the-art models including AMT [21] and EMA-VFI [50]. We adhere to the original hyperparameters for each model for a fair comparison and implement them with PyTorch [31]. For training, we use the septuplet dataset from Vimeo90K [47]. The septuplet dataset comprises 91,701 seven-frame sequences at 448×256, extracted from 39,000 video clips. For evaluation, we use both pixel-centric metrics like PSNR and SSIM [42], and perceptual metrics such as reference-based LPIPS [51] and non-reference NIQE [25]. Concerning the iterative reference-based estimation strategy, D_{ref} during training is calculated from the optical flow derived from ground-truth data at a time point corresponding to a randomly selected reference frame, like $t/2$. In the inference phase, we similarly employ a uniform map for reference, for example, setting $D_{ref} = t/2$. See our source code for details.

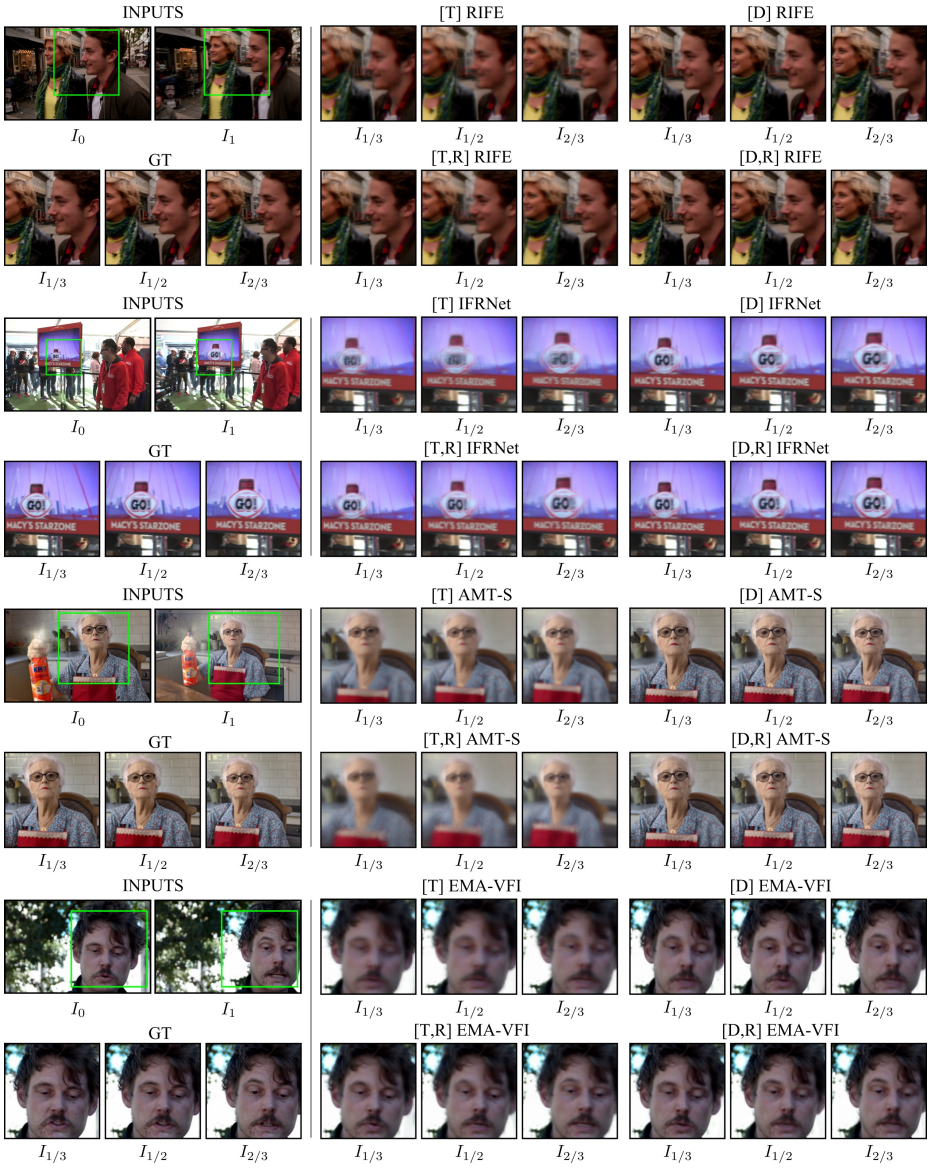


Fig. 5: Comparison of qualitative results. **[T]**: original arbitrary time VFI models using time indexing. **[D]**: models using our distance indexing. **[T,R]**: models using time indexing with iterative reference-based estimation. **[D,R]**: models using both strategies. **All models use uniform maps. Zoom in for a closer look**.

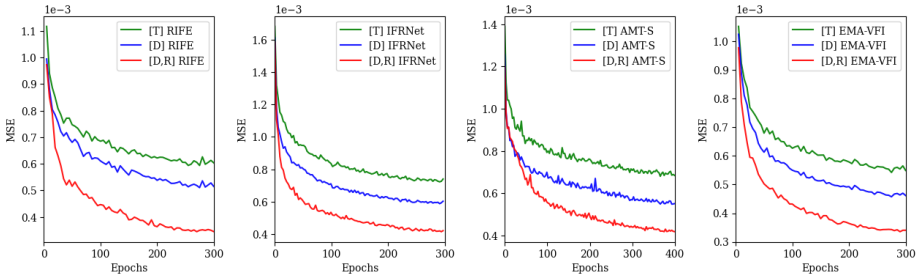


Fig. 6: Convergence curves. $[T]$ denotes traditional time indexing. $[D]$ denotes the proposed distance indexing. $[R]$ denotes iterative reference-based estimation.

5.2 Qualitative comparison

We conducted a qualitative analysis on different variants of each arbitrary time VFI model. We evaluate the base model, labeled as $[T]$, against its enhanced versions, which incorporate distance indexing ($[D]$), iterative reference-based estimation ($[T, R]$), or a combination of both ($[D, R]$), as shown in Fig. 5. We observe that the $[T]$ model yields blurry results with details difficult to distinguish. Models with the distance indexing ($[D]$) mark a noticeable enhancement in perceptual quality, presenting clearer interpolations than $[T]$. In most cases, iterative reference ($[T, R]$) also enhances model performance, with the exception of AMT-S. As expected, the combined approach $[D, R]$ offers the best quality for all base models including AMT-S. This highlights the synergistic potential of distance indexing when paired with iterative reference-based estimation. Overall, our findings underscore the effectiveness of both techniques as plug-and-play strategies, capable of significantly elevating the qualitative performance of cutting-edge arbitrary time VFI models.

5.3 Quantitative comparison

Convergence curves. To further substantiate the efficacy of our proposed strategies, we also conducted a quantitative analysis. Fig. 6 shows the convergence curves for different model variants, *i.e.*, $[T]$, $[D]$, and $[D, R]$. The observed trends are consistent with our theoretical analysis from Sec. 4, supporting the premise that by addressing velocity ambiguity, both distance indexing and iterative reference-based estimation can enhance convergence limits.

Comparison on Vimeo90K septuplet dataset. In Tab. 1, we provide a performance breakdown for each model variant. The models $[D]$ and $[D, R]$ in the upper half utilize ground-truth distance guidance, which is not available at inference in practice. The goal here is just to show the achievable upper-bound performance. On both pixel-centric metrics such as PSNR and SSIM, and perceptual measures like LPIPS and NIQE, the improved versions $[D]$ and $[D, R]$

Table 1: Comparison on Vimeo90K septuplet dataset. $[T]$ denotes the method trained with traditional arbitrary time indexing paradigm. $[D]$ and $[R]$ denote the distance indexing paradigm and iterative reference-based estimation strategy, respectively. $[R]$ uses 2 iterations by default. $[\cdot]_u$ denotes inference with uniform map as time indexes. We utilize the first and last frames as inputs to predict the rest five frames. The **bold font** denotes the best performance in cases where comparison is possible. While the gray font indicates that the scores for pixel-centric metrics, PSNR and SSIM, are not calculated using strictly aligned ground-truth and predicted frames.

	RIFE [11]			IFRNet [18]			AMT-S [21]			EMA-VFI [50]		
	$[T]$	$[D]$	$[D,R]$	$[T]$	$[D]$	$[D,R]$	$[T]$	$[D]$	$[D,R]$	$[T]$	$[D]$	$[D,R]$
PSNR↑	28.22	**29.20**	28.84	28.26	**29.25**	28.55	28.52	**29.61**	28.91	29.41	**30.29**	25.10
SSIM↑	0.912	**0.929**	0.926	0.915	**0.931**	0.925	0.920	**0.937**	0.931	0.928	**0.942**	0.858
LPIPS↓	0.105	0.092	**0.081**	0.088	0.080	**0.072**	0.101	0.086	**0.077**	0.086	**0.078**	0.079
NIQE↓	6.663	6.475	**6.286**	6.422	6.342	**6.241**	6.866	6.656	**6.464**	6.736	6.545	**6.241**
	$[T]$	$[D]_u$	$[D,R]_u$	$[T]$	$[D]_u$	$[D,R]_u$	$[T]$	$[D]_u$	$[D,R]_u$	$[T]$	$[D]_u$	$[D,R]_u$
PSNR↑	28.22	27.55	27.41	28.26	27.40	27.13	28.52	27.33	27.17	29.41	28.24	24.73
SSIM↑	0.912	0.902	0.901	0.915	0.902	0.899	0.920	0.902	0.902	0.928	0.912	0.851
LPIPS↓	0.105	0.092	**0.086**	0.088	0.083	**0.078**	0.101	0.090	**0.081**	0.086	**0.079**	0.081
NIQE↓	6.663	6.344	**6.220**	6.422	6.196	**6.167**	6.866	6.452	**6.326**	6.736	6.457	**6.227**

Table 2: Ablation study of the number of iterations on Vimeo90K septuplet dataset. $[\cdot]^{\#}$ denotes the number of iterations used for inference.

	RIFE [11]			IFRNet [18]			AMT-S [21]			EMA-VFI [50]		
$[D,R]_u$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$
LPIPS↓	0.093	0.086	**0.085**	0.085	**0.078**	0.078	0.086	**0.081**	0.081	0.084	0.081	**0.080**
NIQE↓	6.331	6.220	**6.186**	6.205	6.167	**6.167**	6.402	**6.326**	6.327	6.303	6.227	**6.211**
$[T,R]$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$	$[\cdot]^1$	$[\cdot]^2$	$[\cdot]^3$
LPIPS↓	0.103	0.087	**0.087**	0.091	0.084	**0.084**	**0.106**	0.135	0.157	0.088	**0.083**	0.085
NIQE↓	6.551	6.300	**6.206**	6.424	6.347	**6.314**	**6.929**	7.246	7.502	6.404	6.280	**6.246**

outperform the base model $[T]$. Notably, the combined model $[D,R]$ using both distance indexing and iterative reference-based estimation strategies performs superior in perceptual metrics, particularly NIQE. The superior pixel-centric scores of model $[D]$ compared to model $[D,R]$ can be attributed to the indirect estimation (2 iterations) in the latter, causing slight misalignment with the ground-truth, albeit with enhanced details.

Table 3: Comparison on Adobe240 [36] for ×8 interpolation with RIFE [11].

	$[T]$	$[D]_u$	$[D,R]_u$
PSNR ↑	30.24	**30.47**	30.30
SSIM ↑	**0.939**	0.938	0.937
LPIPS ↓	0.073	0.057	**0.054**
NIQE ↓	5.206	4.974	**4.907**

Table 4: Comparison on X4K1000FPS [34] for ×8 interpolation with RIFE [11].

	$[T]$	$[D]_u$	$[D,R]_u$
PSNR ↑	36.36	**36.80**	36.26
SSIM ↑	**0.967**	0.964	0.964
LPIPS ↓	0.040	0.032	**0.032**
NIQE ↓	7.130	6.936	**6.924**

Table 5: Comparison on X4K1000FPS [34] for ×16 interpolation with RIFE [11].

	$[T]$	$[D]_u$	$[D,R]_u$
PSNR ↑	31.04	**31.60**	31.52
SSIM ↑	0.910	0.914	**0.922**
LPIPS ↓	0.104	0.094	**0.079**
NIQE ↓	7.215	6.953	**6.927**

Table 6: Comparison on Vimeo90K [47] using GMFlow [45] for distance map calculation with RIFE [11].

	$[T]$	$[D]_u$	$[D,R]_u$
PSNR ↑	**28.22**	27.29	26.96
SSIM ↑	**0.912**	0.898	0.895
LPIPS ↓	0.105	0.101	**0.092**
NIQE ↓	6.663	6.449	**6.280**

In realistic scenarios where the precise distance map is inaccessible at inference, one could resort to a uniform map akin to time indexing. The bottom segment of Tab. 1 shows the performance of the enhanced models $[D]$ and $[D,R]$, utilizing identical inputs as model $[T]$. Given the misalignment between predicted frames using a uniform distance map and the ground-truth, the enhanced models do not outperform the base model on pixel-centric metrics. However, we argue that in most applications, the goal of VFI is not to predict pixel-wise aligned frames, but to generate plausible frames with high perceptual quality. Furthermore, pixel-centric metrics are less sensitive to blur [51], the major artifact introduced by velocity ambiguity. The pixel-centric metrics are thus less informative and denoted in gray. On perceptual metrics (especially NIQE), the enhanced models significantly outperforms the base model. This consistency with our qualitative observations further validates the effectiveness of distance indexing and iterative reference-based estimation.

Ablation study of the number of iterations. Tab. 2 offers an ablation study on the number of iterations and the efficacy of a pure iterative reference-based estimation strategy. The upper section suggests that setting iterations at two strikes a good trade-off between computational efficiency and performance. The lower segment illustrates that while iterative reference-based estimation generally works for time indexing, there are exceptions, as observed with AMT-S. However, when combined with distance indexing, iterative reference-based estimation exhibits more stable improvement, as evidenced by the results for $[D,R]_u$. This is consistent with qualitative comparison.

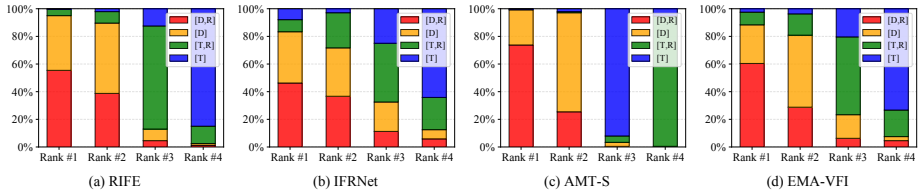


Fig. 7: User study. The horizontal axis represents user rankings, where #1 is the best and #4 is the worst. The vertical axis indicates the percentage of times each model variant received a specific ranking. Each model variant was ranked an equal number of times. The model $[D, R]$ emerged as the top performer. All models use uniform maps.

Comparison on other benchmarks. The septuplet set of Vimeo90K [47] is large enough to train a practical video frame interpolation model, and it represents the situations where the temporal distance between input frames is large. Thus, Vimeo90K (septuplet) can well demonstrate the velocity ambiguity problem that our work aims to highlight. Nonetheless, we report more results on other benchmarks. Tab. 3 and Tab. 4 show the results of RIFE [11] on Adobe240 [36] and X4K1000FPS [34] for $\times 8$ interpolation respectively, using uniform maps. Distance indexing $[D]_u$ and iterative reference-based estimation $[R]_u$ strategies can consistently help improve the perceptual quality. In addition, it is noteworthy that $[D]_u$ is better than $[T]$ in terms of the pixel-centric metrics like PSNR, showing that the constant speed assumption (uniform distance maps) holds well on these two easier benchmarks. We further show $\times 16$ interpolation on X4K1000FPS with larger temporal distance in Tab. 5. The results highlight that the benefits of our strategies are more pronounced with increased temporal distances.

Other optical flow estimator. We also employ GMFlow [45] for the precomputation of distance maps, enabling an analysis of model performance when integrated with alternative optical flow estimations. The results are as shown in Tab. 6. Our strategies still lead to consistent improvement on perceptual metrics. However, this more recent and performant optical flow estimator does not introduce improvement compared to RAFT [38]. A likely explanation is that since we quantify the optical flow to $[0, 1]$ scalar values for better generalization, our training strategies are less sensitive to the precision of the optical flow estimator.

5.4 User study

To validate the effectiveness of our proposed strategies, we further conducted a user study with 30 anonymous participants. Participants were tasked with ranking the interpolation quality of frames produced by four model variants: [T], [D], [T,R], and [D,R]. See details of user study UI in supplementary materials. The results, presented in Fig. 7, align with our qualitative and quantitative

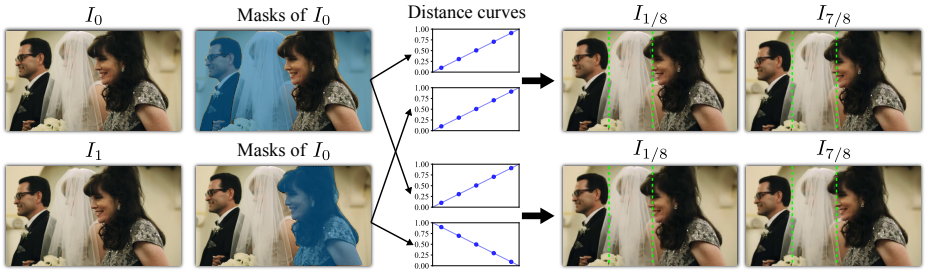


Fig. 8: Manipulated interpolation of anything. Leveraging Segment-Anything [17], users can tailor distance curves for selected masks. Distinct masks combined with varying distance curves generate unique distance map sequences, leading to diverse interpolation outcomes.

findings. The [D,R] model variant emerged as the top-rated, underscoring the effectiveness of our strategies.

5.5 2D manipulation of frame interpolation

Beyond simply enhancing the performance of VFI models, distance indexing equips them with a novel capability: tailoring the interpolation patterns for each individual object, termed as "manipulated interpolation of anything". Fig. 8 demonstrates the workflow. The first stage employs SAM [17] to produce object masks for the starting frame. Users can then customize the distance curve for each object delineated by the mask, effectively controlling its interpolation pattern, *e.g.*, having one person moving backward in time. The backend of the application subsequently generates a sequence of distance maps based on these specified curves for interpolation. One of the primary applications is re-timing specific objects **(See the supplementary video)**.

6 Conclusion and Future Work

We challenge the traditional time indexing paradigm and address its inherent uncertainties related to velocity distribution. Through the introduction of distance indexing and iterative reference-based estimation strategies, we offer a transformative paradigm to VFI. Our innovative plug-and-play strategies not only improves the performance in video interpolation but also empowers users with granular control over interpolation patterns across varied objects. Estimating accurate distance ratio maps from multiple frames represents a direction for our future research. Furthermore, the insights gleaned from our strategies have potential applications across a range of tasks that employ time indexing, such as space-time super-resolution, future predictions, blur interpolation and more.

Acknowledgement

This work was partially supported by the Shanghai Artificial Intelligence Laboratory. We thank Dorian Chan, Zhirong Wu, and Stephen Lin for their insightful feedback and advice. Our thanks also go to Vu An Tran for developing the web application, and to Wei Wang for coordinating the user study.

References

1. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3703–3712 (2019)
2. Chen, S., Zwicker, M.: Improving the perceptual quality of 2d animation interpolation. In: European Conference on Computer Vision. pp. 271–287. Springer (2022)
3. Chen, Z., Chen, Y., Liu, J., Xu, X., Goel, V., Wang, Z., Shi, H., Wang, X.: 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. pp. 2047–2057 (2022)
4. Cheng, X., Chen, Z.: Multiple video frame interpolation via enhanced deformable separable convolution. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 7029–7045 (2021)
5. Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M.: Channel attention is all you need for video frame interpolation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10663–10671 (2020)
6. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
7. Fan, B., Dai, Y.: Inverting a rolling shutter camera: bring rolling shutter images to high framerate global shutter video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4228–4237 (2021)
8. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
9. Hu, M., Jiang, K., Zhong, Z., Wang, Z., Zheng, Y.: Iq-vfi: Implicit quadratic motion estimation for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6410–6419 (2024)
10. Hu, P., Niklaus, S., Sclaroff, S., Saenko, K.: Many-to-many splatting for efficient video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3553–3562 (2022)
11. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV. pp. 624–642. Springer (2022)
12. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)

13. Ji, X., Wang, Z., Zhong, Z., Zheng, Y.: Rethinking video frame interpolation from shutter mode induced degradation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12259–12268 (2023)
14. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9000–9008 (2018)
15. Jin, X., Wu, L., Chen, J., Chen, Y., Koo, J., Hahm, C.h.: A unified pyramid recurrent network for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1578–1587 (2023)
16. Kalluri, T., Pathak, D., Chandraker, M., Tran, D.: Flavr: Flow-agnostic video representations for fast frame interpolation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2071–2082 (2023)
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
18. Kong, L., Jiang, B., Luo, D., Chu, W., Huang, X., Tai, Y., Wang, C., Yang, J.: Ifrnet: Intermediate feature refine network for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1969–1978 (2022)
19. Lee, H., Kim, T., Chung, T.y., Pak, D., Ban, Y., Lee, S.: Adacof: Adaptive collaboration of flows for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5316–5325 (2020)
20. Lee, S., Lee, H., Shin, C., Son, H., Lee, S.: Exploring discontinuity for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9791–9800 (2023)
21. Li, Z., Zhu, Z.L., Han, L.H., Hou, Q., Guo, C.L., Cheng, M.M.: Amt: All-pairs multi-field transforms for efficient frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9801–9810 (2023)
22. Lin, G., Han, J., Cao, M., Zhong, Z., Zheng, Y.: Event-guided frame interpolation and dynamic range expansion of single rolling shutter image. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 3078–3088 (2023)
23. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: Proceedings of the IEEE international conference on computer vision. pp. 4463–4471 (2017)
24. Lu, L., Wu, R., Lin, H., Lu, J., Jia, J.: Video frame interpolation with transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3532–3542 (2022)
25. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal processing letters **20**(3), 209–212 (2012)
26. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5437–5446 (2020)
27. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 670–679 (2017)
28. Park, J., Kim, J., Kim, C.S.: Biformer: Learning bilateral motion estimation via bilateral transformer for 4k video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1568–1577 (2023)

29. Park, J., Ko, K., Lee, C., Kim, C.S.: Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 109–125. Springer (2020)
30. Park, J., Lee, C., Kim, C.S.: Asymmetric bilateral motion estimation for video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14539–14548 (2021)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
33. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Blurry video frame interpolation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5114–5123 (2020)
34. Sim, H., Oh, J., Kim, M.: Xvfi: Extreme video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14489–14498 (2021)
35. Siyao, L., Zhao, S., Yu, W., Sun, W., Metaxas, D., Loy, C.C., Liu, Z.: Deep animation video interpolation in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6587–6595 (2021)
36. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1279–1288 (2017)
37. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
38. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
39. Tulyakov, S., Gehrig, D., Georgoulis, S., Erbach, J., Gehrig, M., Li, Y., Scaramuzza, D.: Time lens: Event-based video frame interpolation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16155–16164 (2021)
40. Wang, T., Zhang, J., Fei, J., Ge, Y., Zheng, H., Tang, Y., Li, Z., Gao, M., Zhao, S., Shan, Y., et al.: Caption anything: Interactive image description with diverse multimodal controls. arXiv preprint arXiv:2305.02677 (2023)
41. Wang, Y., Stanton, D., Zhang, Y., Ryan, R.S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., Saurous, R.A.: Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning. pp. 5180–5189. PMLR (2018)
42. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
43. Wu, C.Y., Singhal, N., Krahenbuhl, P.: Video compression through image interpolation. In: Proceedings of the European conference on computer vision (ECCV). pp. 416–431 (2018)
44. Wu, Y., Wen, Q., Chen, Q.: Optimizing video prediction via video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17814–17823 (2022)

45. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Tao, D.: Gmflow: Learning optical flow via global matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8121–8130 (2022)
46. Xu, X., Siyao, L., Sun, W., Yin, Q., Yang, M.H.: Quadratic video interpolation. Advances in Neural Information Processing Systems **32** (2019)
47. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision **127**, 1106–1125 (2019)
48. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)
49. Yu, T., Feng, R., Feng, R., Liu, J., Jin, X., Zeng, W., Chen, Z.: Inpaint anything: Segment anything meets image inpainting. arXiv preprint arXiv:2304.06790 (2023)
50. Zhang, G., Zhu, Y., Wang, H., Chen, Y., Wu, G., Wang, L.: 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. pp. 5682–5692 (2023)
51. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
52. Zhong, Z., Cao, M., Ji, X., Zheng, Y., Sato, I.: Blur interpolation transformer for real-world motion from blur. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5713–5723 (2023)
53. Zhong, Z., Cao, M., Sun, X., Wu, Z., Zhou, Z., Zheng, Y., Lin, S., Sato, I.: Bringing rolling shutter images alive with dual reversed distortion. In: European Conference on Computer Vision. pp. 233–249. Springer (2022)
54. Zhong, Z., Sun, X., Wu, Z., Zheng, Y., Lin, S., Sato, I.: Animation from blur: Multi-modal blur decomposition with motion guidance. In: European Conference on Computer Vision. pp. 599–615. Springer (2022)
55. Zhou, K., Li, W., Han, X., Lu, J.: Exploring motion ambiguity and alignment for high-quality video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22169–22179 (2023)