Aligning Anything: Hierarchical Motion Estimation for Video Frame Interpolation

Mengshun Hu[1], Zhihang Zhong[4*], Yansheng Qiu[2,3], Zheng Wang[2,3*], and Xiao Sun[5]

[1] INSAIT, Sofia University "St. Kliment Ohridski"
[2] National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University
[3] Hubei Key Laboratory of Multimedia and Network Communication Engineering
[4] Shanghai Jiao Tong University
[5] Shanghai Artificial Intelligence Laboratory
mengshun.hu@insait.ai, zhongzhihang@sjtu.edu.cn, {qiuyansheng,wangzwhu}@whu.edu.cn, sunxiaojay123@gmail.com

Abstract. Existing advanced video frame interpolation (VFI) methods struggle to achieve accurate per-pixel motion or object-level motion estimation. The reasons lie in that pixel-level motion estimation allows for infinite possibilities, making it challenging to guarantee global motion consistency. Conversely, local pixel-level motion tends to be discontinuous whereas object-level motion estimation is typically optimized for global motion. Therefore, a hierarchical motion learning scheme is imperative to enhance the consistency and continuity of motion prediction in VFI. To this end, we marry the object-level motion to the pixel-level motion to construct a hierarchical motion estimation framework. This approach elaborately incorporates object priors derived from open-world knowledge models, such as Segment Anything Model (SAM), to facilitate latent object-level motion learning. In particular, a hybrid contextual feature extraction module (HCE) is employed to aggregate both pixel-wise and semantic representations. This is further reinforced by hierarchical motion loss (HML) and hierarchical feature loss (HFL), which respectively simulate the consistent and continuous motion patterns for individual objects. Our framework supports two configurations: a full variant with HCE, HML, and HFL for higher accuracy, and an efficient variant with only HML and HFL, which introduces no additional inference overhead. Extensive experiments demonstrate consistent improvements over state-of-the-art VFI methods across multiple benchmarks. https://github.com/hhhhhumengshun/SAM-VFI

Keywords: Video frame interpolation · Motion estimation · Segment Anything Model

1 Introduction

Video frame interpolation (VFI) aims to increase the frame rate of videos by synthesizing intermediate frames between two consecutive input frames. As a classical problem

* Corresponding authors.

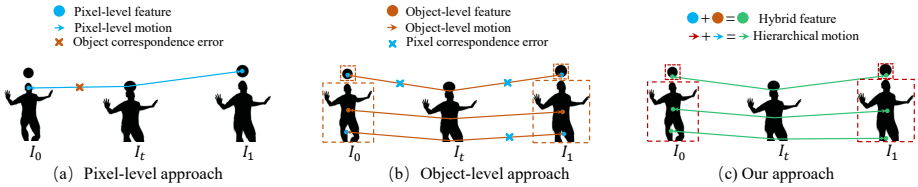

Fig. 1: Different schemes for VFI. (a) Pixel-level approach: they extract pixel-level feature to predict per-pixel motion between two input frames I_0 and I_1 for frame interpolation I_t. (b) Object-level approach: they extract object-level feature to predict the entire object motion for VFI. (c) Ours: **we aggregate pixel-level and object-level features to derive hybrid discriminative feature, enabling hierarchical motion estimation to simulate the current motion patterns.**

in video processing, this task has contributed to various applications, including slow-motion generation [15, 26], movie production [45], video compression [47], *etc*.

Based on the granularity of motion learning, existing VFI methods can be roughly divided into pixel-level and object-level approaches. The former typically predicts pixel-level motion between two consecutive input frames (**See Fig. 1 (a)**), and then synthesize intermediate frames by warping the input frames [26]. However, pixel-level motion estimation presents infinite possible correspondences, making great difficult to ensure global motion consistency. Though advanced techniques like global attention representation [29, 54], have been employed to refine motion estimation, the correspondence ambiguity cannot be eradicated.

Object-level approaches leverage object priors to facilitate effective motion estimation [17, 41]. Traditional methods typically segment the scene into distinct semantic categories and learn motion representations for individual objects. However, these methods are primarily optimized for global object motion, while local pixel-level motion within an object often remain discontinuous(**See deformed ball in Fig. 1 (b)**). In addition, their reliance on predefined semantic categories limits their ability to recognize arbitrary objects in open-world scenes.

To accurately simulate the motion of anything in complex scenes, we explicitly introduce object priors and propose a hierarchical motion learning strategy for VFI. Our approach seamlessly marries the object-level motion to the pixel-level motion, thereby improving the consistency and continuity of motion estimation. Specifically, we first preprocess the outputs of the Segment Anything Model [21, 40] to obtain precise object masks, which are then used as object priors to enhance both contextual extraction and motion optimization through three adaptations. 1) A hybrid contextual feature extraction module (HCE) that aggregates pixel-wise and semantic representations to produce hybrid discriminative features for motion estimation. 2) A hierarchical motion loss (HML) that adaptively distills consistent motion patterns from pseudo labels at both pixel and object levels. 3) A hierarchical feature loss (HFL) that further evaluates feature-correlation differences within individual objects to alleviate intra-object motion discontinuity. These adaptations can be seamlessly integrated into SOTA VFI methods. For accuracy-oriented applications, the complete configuration combines HCE, HML, and HFL to achieve the strongest interpolation performance. For efficiency-oriented

applications, HML and HFL can be used without HCE, improving motion consistency and continuity without introducing additional computational overhead during inference. Extensive experiments demonstrate that both configurations improve motion estimation and interpolation quality across various benchmark datasets. Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to explicitly leverage object information to improve motion estimation for VFI using deep learning.
- We propose a hybrid contextual feature extraction module (HCE) to aggregate pixel-wise and semantic representations, together with a hierarchical motion loss (HML) and a hierarchical feature loss (HFL) to improve motion consistency and continuity, respectively.
- We provide both accuracy-oriented and efficiency-oriented configurations, allowing the proposed framework to improve VFI performance with or without additional inference overhead on multiple benchmark datasets.

2 Related Work

2.1 Video Frame Interpolation

Existing VFI methods can be broadly categorized into motion-free [20] and motion-based [13] methods, depending on whether they explicitly incorporate motion cues such as optical flow.

Motion-free methods: These methods rely on phase prediction [31,32], adaptive kernel generation [5,6,24,35,36,42,43] or spatio-temporal encoder-decoder [9] architectures to directly synthesize intermediate frames. However, the lack of explicit motion modeling constraints often leads to undesirable artifacts, particularly in scenes involving large motions and occlusions.

Motion-based methods: These methods typically estimate intermediate optical flows between two consecutive frames, and use the predicted flows to propagate pixels/features for intermediate frame synthesis [18, 19, 26, 27, 37]. To improve robustness in complex scenarios, Niklaus *et al.* extract per-pixel contextual information from the input frames to compensate for inaccuracies in optical flow estimation [33]. Bao *et al.* introduce depth information to explicitly detect occlusions, reasoning that closer pixels should be preferably synthesized in the intermediate frame [2]. Nevertheless, most existing motion-based methods mainly focus on pixel-level motion estimation. Due to the lack of semantic information, they often struggle to establish reliable correspondences between input frames in complex scenes. Recent work has introduced SAM prior [22] to identify corresponding regions in adjacent frames for motion estimation [11]. However, such methods do not fully and explicitly exploit object-level information for VFI. In contrast, we preprocess SAM outputs to obtain precise object masks and use them as object priors to enhance both contextual feature extraction and motion optimization. This is achieved through three novel adaptations: hybrid contextual feature extraction module (HCE), hierarchical motion loss (HML) and hierarchical feature loss (HFL).

2.2 Segment Anything Model

Segment Anything Model (SAM) [22] and Segment Anything Model 2 (SAM2) [40] are two recently introduced foundation models for general image/video segmentation. Trained on large-scale datasets with high-quality annotations, they can generate object masks from different forms of prompts, such as points and bounding boxes. Owing to their strong zero-shot generalization ability, these pretrained models can adapt effectively to unseen data distributions. As a result, they are been widely applied to various computer vision tasks, including low-level vision [51, 52, 55], object tracking [7, 50], video segmentation [4, 10] and medical imaging [1, 8, 30]. In this work, we explicitly utilize SAM priors to perform hierarchical motion estimation, thereby improving the accuracy, consistency, and continuity of motion estimation for individual objects.

3 Methodology

3.1 Preliminaries

Given two consecutive frames I_0 and I_1, pixel-level VFI extracts their features $\hat{F}_0$ and $\hat{F}_1$, and predicts bidirectional pixel-level motions $\hat{f}_{t0}$ and $\hat{f}_{t1}$ via a shared feature extraction module (FE) and motion estimation module (ME), these motions are then used to synthesize the intermediate frame $\hat{I}_t$ via synthesis module (Syn). The whole process is defined as follows:

$$\begin{aligned} \hat{F}_0 = FE(I_0),\ \hat{F}_1 = FE(I_1), \\ \hat{f}_{t0}, \hat{f}_{t1} = ME(W(\hat{F}_0, \hat{f}_{t0}), W(\hat{F}_1, \hat{f}_{t1}), t), \\ \hat{I}_t = Syn(W(I_0, \hat{f}_{t0}), W(I_1, \hat{f}_{t1})), \end{aligned} \tag{1}$$

where $W(,)$ denotes backward warping [27]. By observing Eq. (1), motion estimation [13] is the most critical step in the well-established paradigms of VFI (Note that synthesis module serves merely as a post-processing module and is not a key focus of this paper). To thoroughly analyze motion estimation from a probabilistic perspective, taking motion estimation in one direction as an example:

$$\hat{f}_{t0} = \underset{f_{t0}}{\operatorname{argmax}}\, p(f_{t0}|I_0, I_1, t) = \frac{p(t)p(I_0|t)p(f_{t0}|I_0, t)p(I_1|I_0, f_{t0}, t)}{p(I_0, I_1, t)}, \tag{2}$$

where $\hat{f}_{t0}$ is the most likely estimated motion, and $p(f_{t0}|I_0, I_1, t)$ is the posterior distribution of the motion. we omit unrelated terms and take the logarithm to simplify the multiplication terms (Note that motion estimation from another direction is similar):

$$\begin{aligned} \hat{f}_{t0} = \underset{f_{t0}}{\operatorname{argmax}} \big\{ \underbrace{\log p(f_{t0}|I_0, t)}_{\text{context extraction}} + \underbrace{\log p(I_1|I_0, f_{t0}, t)}_{\text{motion optimization}} \big\}, \\ \hat{f}_{t1} = \underset{f_{t1}}{\operatorname{argmax}} \big\{ \underbrace{\log p(f_{t1}|I_1, 1-t)}_{\text{context extraction}} + \underbrace{\log p(I_0|I_1, f_{t1}, 1-t)}_{\text{motion optimization}} \big\}. \end{aligned} \tag{3}$$

3.2 Motivation

The context extraction terms from Eq. (3) provide an alternative information source (I_0 and I_1) for motion estimation ($\hat{f}_{t0}$ and $\hat{f}_{t1}$), respectively. However, pixel-wise contextual features $\hat{F}_0$ and $\hat{F}_1$ extracted by existing methods [2, 33] are largely limited to local spatial cues, making them insufficient for establishing reliable correspondences in challenging scenarios. In contrast, object-level information provides a more global representation of individual moving entities and can serve as a strong prior for motion estimation. To exploit such information, we introduce SAM priors to capture object-level global contextual features $\hat{O}_0$ and $\hat{O}_1$ via hybrid contextual feature extraction module (HCE), thereby forming more discriminative hybrid features $\hat{H}_0$ and $\hat{H}_1$ that are essential for understanding spatial relationships and interactions between objects:

$$\hat{H}_0 = HCE(\hat{F}_0, \hat{O}_0),\ \hat{H}_1 = HCE(\hat{F}_1, \hat{O}_1). \tag{4}$$

Since the intermediate frame I_t is unavailable, the motion optimization terms in Eq (3) emphasize the complementary relationship between the warped intermediate features ($W(\hat{F}_0, \hat{f}_{t0}), W(\hat{F}_1, \hat{f}_{t1})$) and optical flows ($\hat{f}_{t0}$, $\hat{f}_{t1}$), maintaining motion consistency through a joint optimization framework [23, 25]. But existing methods rely solely on the supervision from the final intermediate frame $\hat{I}_t$ via Charbonnier loss L_{cl} [3]:

$$L_{cl} = \rho(\hat{I}_t - I_t), \tag{5}$$

where $\rho(x) = (x^2 + \epsilon^2)^\alpha$ with $\alpha = 0.5$, $\epsilon = 10^{-3}$. These methods struggle to simultaneously achieve motion consistency and continuity for individual objects, primarily due to their neglect of object-level information. To alleviate this limitation, in addition to L_{cl}, we integrate SAM priors M_t extracted from the ground-truth intermediate frame I_t with pixel-level motions $\hat{f}_{t0}$ and $\hat{f}_{t1}$, enabling the proposed hierarchical motion loss (HML) (L_{hml}), to improve motion consistency through adaptive distillation:

$$L_{hml} = HML(\hat{f}_{t0}, f_{t0}, M_t) + HML(\hat{f}_{t1}, f_{t1}.M_t). \tag{6}$$

Inspired by the theory that optical flow field and the image itself share high structure similarity for individual object [48], we propose hierarchical feature loss (HFL) L_{hfl}, which evaluates feature-correlation differences within each objects to promote intra-object motion continuity:

$$L_{hfl} = HFL(\hat{I}_t, I_t, M_t). \tag{7}$$

3.3 Overview

Building upon AMT [25], a representative pixel-level baseline for VFI, we introduce object-level cues to enhance its contextual feature extraction and motion estimation capabilities through three adaptations: hybrid contextual feature extraction module (HCE), hierarchical motion loss (HML), and hierarchical feature loss (HFL). As shown in Figure 2, the baseline network takes two consecutive frames I_0 and I_1 as input. A series of feature extractors E_i first produce multi-scale features $\hat{F}_0^i$ $\hat{F}_1^i$ while the motion estimation modules D_i predict the corresponding multi-scale optical flows $\hat{f}_{t0}^i$ and $\hat{f}_{t1}^i$. The

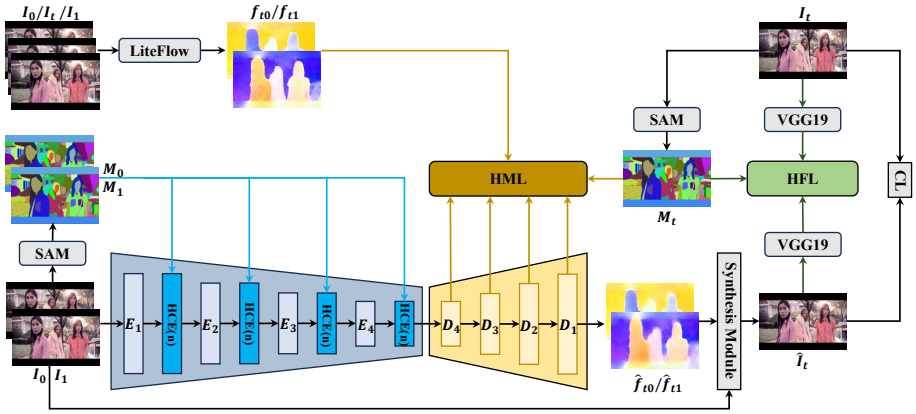

Fig. 2: The overall framework. Our framework consists of pixel-level baseline network (*i.e., feature extraction module, motion estimation module and synthesis module*, pre-trained models (*i.e., SAM [22,40] and LiteFlow [16]*), and our three adaptations (*i.e., HCE, HML and HFL*).Take AMT as an example, we omit the all-pairs correlation processing for brevity, as it does not require improvement. **Note: HML and HFL are used only during training and do not involve any additional computation during inference. Using HCE introduces computation cost from itself and SAM during inference.**

synthesis module then generates the intermediate frame $\hat{I}_t$ through warping and fusing. To enhance this framework, we employ the Segment Anything Model (SAM) to extract object masks M_0, M_1 and M_t, which serve as object priors for three key adaptations: 1) HCE uses M_0 and M_1 to aggregate pixel-wise and object-level semantic representations, yielding hybrid multi-scale contextual features H_0^i and H_1^i. 2) HML incorporates M_t to adaptively distill reliable motion patterns from pseudo labels at both pixel and object levels, thereby improving motion consistency. 3) HFL uses M_t to measure feature-correlation differences within individual object regions, promoting intra-motion continuity. HML and HFL are applied only during training and introduce no additional inference overhead, whereas HCE incurs extra computation from both the module itself and SAM during inference.

Semantic Masks Generation. The remarkable capabilities of the Segment Anything Model (SAM) have demonstrated its versatility across a wide range of computer vision (CV) tasks [20, 21, 52]. In VFI, a core challenge is to accurately establish correspondences between regions in consecutive input frames for reliable motion estimation. Intuitively, incorporating semantic information can enhance conventional pixel-level VFI methods by providing higher object-level representations. However, unlike conventional semantic segmentation outputs, a single pixel in SAM's output may belong to multiple object masks, while some pixels may remain unassigned to any mask. This ambiguity complicates the indexing of corresponding object-level information, limiting its effectiveness in enhancing contextual feature extraction and motion optimization.

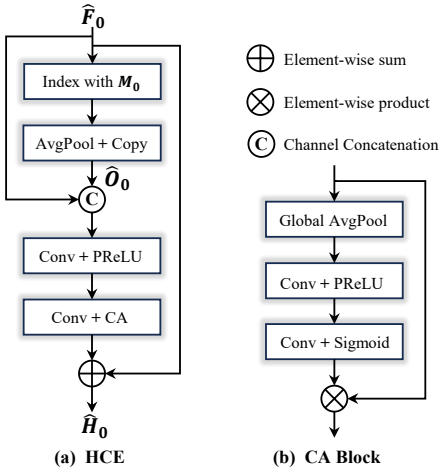

Fig. 3: Architecture of the proposed HCE.

To address this issue, we preprocess the SAM outputs to ensure that each pixel is assigned to exactly one mask. Specifically, The masks are processed in descending order of area. Pixels not covered by any object mask are assigned to a newly created background mask.

Hybrid Contextual Feature Extraction Module. Hybrid contextual feature extraction module (HCE) is designed to extract discriminative features as contextual information for motion estimation. Previous methods [23, 25] extract pixel-level features $\hat{F}_0$ and $\hat{F}_1$ using a shared network, but these features are limited in capturing global spatial context. In this paper, as illustrated in Figure 3, HCE uses the generated object masks to index object-level features in a layered manner. These features are then combined with pixel-level features to model discriminative representation for individual objects (See Eq.(4)). Specifically, taking $HCE(,)$ to obtain $\hat{H}_0$ as an example, the input feature $\hat{F}_0$ from the encoder is first used to index the corresponding object-level global features $\hat{O}_t$ using the generated object masks M_0 ($M_0 = \{M_0^j \mid j = 1, 2, \ldots, m\}$ in a layered manner, where m is the number of objects). This is followed by global pooling and copying operations for each object:

$$\hat{O}_0^j = Copy(AvgPool(Id(\hat{F}_0, M_0^j))), \tag{8}$$

where $Id(\cdot)$ denotes the indexing operation, while $AvgPool(\cdot)$ and $Copy(\cdot)$ denote average pooling and feature copying within the indexed regions, respectively. To further enhance feature aggregation, we employ a channel attention (CA) mechanism [12] to estimate the relative importance of pixel-level feature $\hat{F}_0$ and object-level feature $\hat{O}_0$. The CA block selectively emphasizes informative channels and refines the pixel-level representation to produce the final hybrid contextual feature $\hat{H}_0$:

$$\hat{H}_0 = CA(C(CP([\hat{F}_0, \hat{O}_0]))) + \hat{F}_0, \tag{9}$$

where $C(\cdot)$ and $CP(\cdot)$ denote a convolutional layer and a convolutional layer followed by PReLU activation, respectively.

Fig. 4: Visualization of the predicted pixel-level flows $\hat{f}_{t0}$ and $\hat{f}_{t1}$, pseudo object-level flows $\hat{o}_{t0}$ and $\hat{o}_{t1}$ and pseudo flows f_{t0} and f_{t1}. It shows that object-level optical flows provide useful prior knowledge for suppressing erroneous motion patterns from pseudo-labels, particularly in regions involving large motion, occlusions, and motion boundaries (*See the moving head and hands*).

Hierarchical Motion Loss. Existing methods [15] use the ground-truth intermediate frame I_t to generate pseudo labels f_{t0} and f_{t1} via an off-the-shelf motion estimator for distillation. However, since pseudo optical flow is a sub-optimal representation for VFI [49], such an indiscriminate distillation strategy may introduce undesirable knowledge from the pseudo labels. To mitigate this issue, IFRNet [23] introduces a task-oriented flow distillation loss that suppress adverse effects while retaining beneficial knowledge. Nevertheless, as shown in Fig. 4, the difficulty of distillation varies across different objects, and placing stronger emphasis on objects with complex motion can further suppress negative influences from pseudo labels. Accordingly, we incorporate object priors and propose hierarchical motion loss (HML) to adaptively distill reliable motion patterns from pseudo labels at both pixel and object levels. Specifically, we use pre-trained LiteFlow [16] to estimate bidirectional optical flow f_{t0} and f_{t1} as pseudo labels. From these, we derive pixel-level weight P_l, object-level weight O_l, and hierarchical weight H_l $(l \in 0, 1)$ as follows:

$$\begin{aligned} P_l &= \|f_{tl} - \hat{f}_{tl}\|_{\text{epe}}, \\ \hat{o}_{tl}^j &= Copy(AvgPool(Id(f_{tl}, M_t^j))), \\ O_l &= \|f_{tl} - \hat{o}_{tl}\|_{\text{epe}}, \\ H_l &= exp(-\beta(O_l + P_l)), \end{aligned} \tag{10}$$

where $\|\cdot\|_{\text{epe}}$ calculates per-pixel end-point-error. Following prior work [23], HML employs the Charbonnier loss $\rho(x) = (x^2 + \epsilon^2)^\alpha$ for robust flow distillation, with ϵ and α

controlling its robustness ($\epsilon = 10^{(H_l-1)/3}$ and $\alpha = H_l/2$). The HML is defined as:

$$\mathcal{L}_{hml} = \sum_{i=1}^{3}\sum_{l=0}^{1}\left\|\mathcal{U}_{2^i}(\hat{f}^i_{t\to l}) - f_{t\to l}\right\|, \tag{11}$$

where U_s denotes bilinear upsampling by scale factor s. In this formulation [23], the adaptive weight H_l decreases when the predicted flow at a pixel or object deviates from the pseudo label. This mechanism enhances the robustness of the loss by controlling ϵ and α, and effectively suppresses harmful flow knowledge in the lower-level decoders through gradient descent.

Hierarchical Feature Loss. HML effectively mitigates the negative impact of pseudo labels while distilling useful knowledge. However, the generated intermediate frames still exhibit noticeable deviations from a visually realistic appearance. This limitation stems primarily from the fact that HML still cannot faithfully model continuous motion for individual object from pseudo labels, particularly in challenging scenarios with large motions and occlusions. Inspired by the theory that optical flow field and the image itself share high structure similarity for individual object [48], we introduce hierarchical feature loss (HFL) designed to enforce intra-object feature correlations to alleviate motion discontinuity. Specifically, we employ transformer to model global correlations G_p among pixel-level features via self-attention mechanism:

$$G_p(I) = \mathrm{Softmax}\left(\frac{\Phi(I)\Phi(I)^\top}{\sqrt{d_k}}\right), \tag{12}$$

where $\Phi(\cdot)$ denotes VGG19 feature extractor [44], and d_k is the number of attention heads. To prevent interference from other objects, we further incorporate object masks to constrain the attention regions in the global correlations, thereby focusing more on the correlations within individual object. This process is formulated as follows:

$$\begin{aligned} M_{\mathrm{att}} &= \lambda_1 \sum_{i=1}^{L}\left(I - M_t^j {M_t^j}^\top\right), \\ G_o(I) &= \mathrm{Softmax}\left(M_{\mathrm{att}} + \frac{\Phi(I)\Phi(I)^\top}{\sqrt{d_k}}\right)\Phi(I), \end{aligned} \tag{13}$$

where λ_1 is negative infinite value. Accordingly, the proposed hierarchical feature loss (HFL) measures the correlation differences of individual objects between features extracted from predicted intermediate frame $\hat{I}_t$ and ground-truth intermediate frame I_t:

$$L_{hfl} = \left\|G_o(I_t) - G_o(\hat{I}_t)\right\|_2. \tag{14}$$

4 Experiments

4.1 Benchmarks.

We evaluate our framework on a range of benchmarks containing diverse motion scenarios for a comprehensive comparison. Structural Similarity Index (SSIM) ([46]),

Methods	Vimeo90K [49]	SNU-FILM [9]				Xiph [34]	
		Easy	Medium	Hard	Extreme	2K	4K
AdaCoF [24]	34.59/0.9733/0.0254	39.96/0.9908/0.0217	35.17/0.9768/0.0369	29.54/0.9257/0.0718	24.48/0.8455/0.1361	35.12/0.9603/0.0544	32.19/0.9341/0.1134
ABME [38]	36.22/0.9808/0.0189	39.64/0.9904/0.0233	35.80/0.9791/0.0381	30.59/0.9367/0.0652	25.43/0.8642/0.1193	36.53/0.9641/0.0679	33.74/0.9435/0.1530
RIFE [15]	35.64/0.9779/0.0201	40.06/0.9907/0.0215	35.75/0.9790/0.0354	30.10/0.9330/0.0683	24.84/0.8533/0.1362	36.18/0.9647/0.0616	33.28/0.9418/0.1380
M2M-VFI [14]	35.47/0.9778/0.0220	39.61/0.9902/0.0230	35.71/0.9791/0.0370	30.29/0.9357/0.0649	25.07/0.8601/0.1203	36.44/0.9664/0.0656	33.92/0.9448/0.1419
VFIFormer [29]	36.55/0.9819/0.0184	40.19/0.9910/0.0201	36.12/0.9802/0.0361	30.68/0.9381/0.0646	25.44/0.8645/0.1198	OOM	OOM
IFRNet-M [23]	35.80/0.9793/0.0174	40.03/0.9909/0.0215	35.94/0.9797/0.0328	30.40/0.9361/0.0567	25.05/0.8589/0.1120	36.02/0.9627/0.0470	34.01/0.9414/0.0756
IFRNet-L [23]	36.20/0.9810/0.0168	40.10/0.9910/0.0216	36.12/0.9801/0.0329	30.63/0.9371/0.0558	25.26/0.8611/0.1082	36.24/0.9636/0.0477	34.28/0.9428/0.0762
EMA-VFI [54]	36.50/0.9800/0.0179	39.58/0.9893/0.0218	35.85/0.9786/0.0366	30.81/0.9377/0.0627	25.59/0.8639/0.1127	36.78/0.9678/0.0684	34.57/0.9489/0.1509
VFIMamba* [53]	36.13/0.9806/0.0182	39.90/0.9910/0.0191	35.59/0.9791/0.0342	30.09/0.9330/0.0688	25.01/0.8571/0.1308	36.37/0.9664/0.0644	33.40/0.9445/0.1434
AMT-L [25]	36.35/0.9816/0.0179	39.95/0.9913/0.0256	36.09/0.9805/0.0388	30.75/0.9384/0.0632	25.41/0.8639/0.1115	36.31/0.9649/0.0718	34.51/0.9472/0.1364
AMT-G [25]	36.53/0.9817/0.0171	39.88/0.9913/0.0252	36.12/0.9805/0.0384	30.78/0.9385/0.0622	25.43/0.8645/0.1098	36.42/0.9653/0.0702	34.65/0.9478/0.1334
IFRNet-L‡ [23]	36.06/0.9802/0.0168	40.02/0.9906/**0.0213**	35.98/0.9794/**0.0327**	30.55/0.9363/**0.0555**	25.25/0.8609/**0.1057**	36.16/**0.9652**/0.0471	34.16/**0.9436**/0.0762
Ours (IFRNet-L‡)	**36.28/0.9809/0.0163**	**40.15/0.9907**/0.0214	**36.11/0.9799**/0.0328	**30.61/0.9373**/0.0556	**25.33/0.8618**/0.1072	**36.31**/0.9640/**0.0458**	**34.31**/0.9429/**0.0751**
AMT-G‡ [25]	36.42/0.9816/0.0174	39.91/0.9904/0.0243	36.07/0.9798/0.0376	30.75/0.9377/0.0614	25.38/0.8637/0.1080	36.48/0.9659/0.0700	34.73/0.9489/0.1339
Ours (AMT-G‡)	**36.62/0.9820/0.0168**	**40.30/0.9909/0.0212**	**36.33/0.9802/0.0341**	**30.97/0.9384/0.0583**	**25.61/0.8647/0.1036**	**36.79/0.9664/0.0668**	**34.87/0.9490/0.1295**

Table 1: Quantitative comparisons (PSNR(↑)/SSIM(↑)/LPIPS(↓)) of SOTA methods with our proposed method on Vimeo90K [49], SNU-FILM [9] and Xiph [34] datasets. The numbers in **bold** represents the best score. ∗ indicates that VFIMamba [53] was retrained using its official source code exclusively on the Vimeo90K dataset for fair comparisons.‡ indicates that plugged-in models were retrained under our setting to demonstrate the effectiveness of our adaptations.

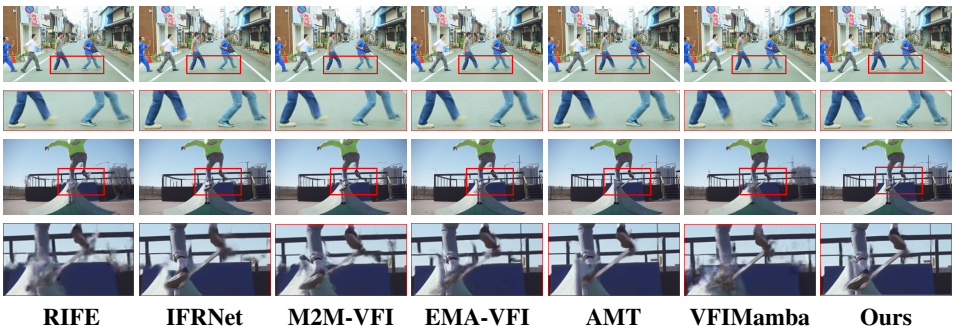

Fig. 5: Visual comparisons of different video frame interpolation methods.

Peak Signal-to-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) [56] are used as evaluation metrics. Higher values of PSNR and SSIM indicate better performance, whereas a lower value of LPIPS signifies better performance. The benchmarks are summarized below:

Vimeo90K [49]. This dataset includes over 60,000 triplets with a resolution of 448×256. A total of 51,312 triplets are cropped into 224×224 patches for training, while 3,782 triplets are reserved for testing.

SNU-FILM [9]. This testset includes 1,240 video triplets with resolutions up to 1280×720, presenting significant challenges in scenarios with large motions and occlusions. This dataset is divided into four difficulty levels: Easy, Medium, Hard, and Extreme.

Xiph [34]. This testset comprises eight 4K videos. Following [2], we downsample and center-crop the original image to 2K resolution to get "Xiph-2K" and "Xiph-4K".

4.2 Implementation Details

We optimize the full models using the proposed loss functions. The number of HCE is set to 6 for large VFI models and 1 for small VFI models. Specifically, the models are trained end-to-end with the AdamW optimizer [28] on 8 RTX 4090 GPUs. Training is performed on the Vimeo90K dataset for 300 epochs, with a batch size of 48 and image patches of size 224×224. The initial learning rate is set to 1×10^{-4} and is gradually reduced to 1×10^{-5} following a cosine annealing schedule.

4.3 Comparisons with the SOTAs

We integrate object masks into the representative SOTA method AMT [25], and compare their performance with nine methods: AdaCoF [24], ABME [39], RIFE [15], M2M-VFI [14], VFIFormer [29], IFRNet [23], EMA-VFI [54], AMT [25] and VFIMamba [53]. Note that we retrain VFIMamba and plugged-in model using its official code exclusively on Vimeo90K dataset for fair comparisons.

Quantitative Comparison. As shown in Table 1, when faced with simple scenarios on Vimeo90K and SNU-FILM datasets, SOTA pixel-level methods achieve results comparable to ours. However, the model with our adaptations significantly outperforms these methods in challenging scenarios through hierarchical motion estimation. Specifically, our model surpasses advanced CNN-based pixel-level methods, including RIFE [15], IFRNet [23] and AMT [25], by clear margins of 0.77 dB, 0.35 dB and 0.18 dB, respectively, on the Extreme subsets of SNU-FILM dataset. This superiority arises from the inherent limitations of CNN-based pixel-level methods, which struggle with infinite possibilities in motion estimation, making accurate motion simulation and interpolation highly challenging. Moreover, their limited receptive fields hinder the ability to capture large motions effectively. Additional, Transformer-based pixel-level method EMA [54] employs window-based attention for motion estimation in VFI. However, due to the lack of object boundary constraints, it is susceptible to interference from irrelevant information, leading to a performance gap of 0.30 dB on 4K dataset compared to our method. All these results highlight the effectiveness of our object prior in VFI.

Qualitative Comparison. The qualitative results of SOTA methods and our method are shown in Figure 5. It is apparent that previous pixel-level VFI methods struggle to produce sharp edges of moving objects, particularly in scenarios involving large and complex motions **(See the moving legs and skateboard)**. Even transformer-based method EMA [54] encounters similar challenges. The underlying issue is their inability to distinguish, match and refine object motion between input frames. In contrast, our approach comprehensively incorporates object prior, allowing for motion estimation and interpolation specific to corresponding regions. So our model accurately synthesizes content at motion boundaries and generates credible textures with fewer artifacts.

4.4 Ablation Study

This section presents comprehensive ablation studies to evaluate the impact of each component in a generalized manner, like IFRNet [23], using IFRNet-M integrated with our three adaptations. For a fair comparison, all models are trained on the Vimeo90K

dataset. The inference time is measured for processing 1280×720 resolution image on an NVIDIA RTX 3090 GPU.

Individual Components. We conducted additional experiments to validate the effectiveness of individual components across various model configurations. Quantitative results are summarized in Table 2, the baseline model, which relies solely on pixel-level contexts from the encoder to estimate pixel-level motion estimation using the loss function L_{cl}, achieves the worst performance for video frame interpolation. Upon incrementally incorporating hybrid contextual feature extraction module (HCE) for discriminative feature extraction and hierarchical motion loss (HML) for adaptive distillation, performance improves by 0.15 dB and 0.17 dB in PSNR, respectively. Integrating both components further enhances video frame interpolation, yielding an additional gains of 0.26dB. Moreover, with the inclusion of the hierarchical feature loss (HFL), our model enjoys strong capability to produce results that are perceptually closer to the ground truth, achieving the best performance in LPIPS.

Effects of HCE. To evaluate the importance of our HCE in context extraction, we conduct an ablation study comparing pixel-level and hybrid feature extraction. As shown in Table 3, applying HCE model solely on pixel-level features without incorporating object-level information yields a PSNR improvement of 0.06dB, confirming the effectiveness of HCE. Furthermore, we introduce object-level and obtain an additional gain of 0.09dB, which can be largely attributed to the model's enhanced capability to extract more discriminative contextual features.

Effects of HML. To verify the important of our HIR in motion estimation, we perform an ablation study on pixel-level, object-level and hierarchical motion loss. As evidenced by Table 4, the baseline performance is significantly improved by the inclusion of pixel-level (+0.09 dB) and object-level (+0.1 dB) motion losses. The hierarchical combination of these losses, formulated in Eq. 10, achieves the most substantial gain of 0.17 dB. These findings demonstrate that our HML offers superior adaptability in distilling the teacher model's knowledge from pseudo optical flows.

Effects of HFL. To verify the important of our HFL in continuous motion estimation, we perform an ablation study on pixel-level and hierarchical feature losses. As shown in Table 5, compared to the baseline, introducing pixel-level motion feature loss leads to a significant improvement in LPIPS performance. This improvement stems from our model's ability to leverage feature correlations to enforce flow continuity. With the

Case			Vimeo90K	Medium	Time	Params	FLOPs
HCE	HML	HFL	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS	(ms)	(M)	(T)
✗	✗	✗	35.70/0.9789/0.0176	35.84/0.9796/0.0327	30	5.0	0.21
✓	✗	✗	35.85/0.9793/0.0171	36.01/0.9798/0.0325	38	5.5	0.24
✗	✓	✗	35.87/0.9795/0.0175	35.96/0.9798/0.0330	30	5.0	0.21
✓	✓	✗	35.96/0.9800/0.0172	**36.03**/0.9799/0.0325	38	5.5	0.24
✗	✓	✓	35.84/0.9793/0.0172	35.92/0.9797/0.0327	30	5.0	0.21
✓	✓	✓	**35.98/0.9802/0.0169**	36.01/**0.9800/0.0320**	38	5.5	0.24

Table 2: Effects of individual components (PSNR/SSIM/LPIPS). **For real-time application of the plug-in model, we recommend avoiding the introduction of the HCE module to prevent the additional computational overhead introduced by both the module itself and SAM.**

HCE		Vimeo90K	Medium
Pixel-level	Object-level	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS
✓	✗	35.76/0.9790/0.0172	35.90/0.9793/0.0326
✓	✓	**35.85/0.9793/0.0171**	**36.01/0.9798/0.0325**

Table 3: Effects of HCE (PSNR/SSIM/LPIPS).

HML		Vimeo90K	Medium
Pixel-level	Object-level	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS
✓	✗	35.79/0.9793/0.0177	35.88/0.9796/0.0332
✗	✓	35.80/0.97930.0176	35.90/0.9797/0.0331
✓	✓	**35.87/0.9795/0.0175**	**35.96/0.9798/0.0330**

Table 4: Effects of HML (PSNR/SSIM/LPIPS).

guidance of the object priors, we can further constrain HFL to focus on intra-object correlations with clear boundaries, resulting in even better LPIPS.

4.5 Visualization Analysis

The Visualization of Intermediate Motion. To further validate the effectiveness of integrating object masks for motion estimation, we visualize the final bidirectional optical flows generated by our method and the baseline IFRNet [15]. As shown in Figure 6, IFRNet fails to accurately capture the large motion of the wings in the absence of motion supervision. Although pixel-level supervision allows the model to distill useful knowledge from pseudo flows, it struggle to effectively filter out detrimental information, leading to inaccurate motion estimation with artifact and blurry boundaries. In contrast, our method achieves more accurate and complete motion estimations through hierarchical motion supervision. This improvement can be attributed to the use of object masks, which help suppress the negative influences of pseudo flows while effectively distilling useful knowledge in a region-aware manner.

The Visualization of Intermediate Frame. As revealed by the visualization results in Figure 7, the framework is able to capture global correlations through pixel-level feature loss. However, it encounters difficulties in interpolating consistent content within each object undergoing large motion (**See the leg of the bear**). With the assistance of HFL, our framework effectively constrains the attention mechanism to capture the correlations within the leg region, thereby gaining a stronger ability to generate visually and semantically coherent images that closely match the ground truth.

HFL		Vimeo90K	Medium
Pixel-level	Object-level	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS
✓	✗	35.72/0.9790/0.0175	35.88/0.9796/0.0326
✓	✓	**35.74/0.9791/0.0172**	**35.93/0.9797/0.0321**

Table 5: Effects of HFL (PSNR/SSIM/LPIPS).

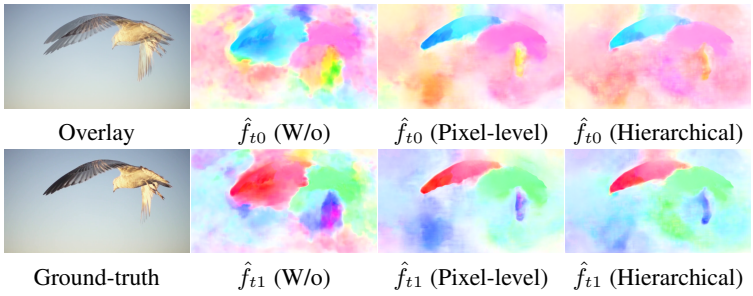

Fig. 6: Visualization of pixel-level flows $\hat{f}_{t0}$ and $\hat{f}_{t1}$ under three settings: without motion supervision, with pixel-level motion supervision and with hierarchical motion supervision. The results demonstrate that the hierarchical motion loss enables the model to produce complete motion with **clear boundary**. **(See the wing)**.

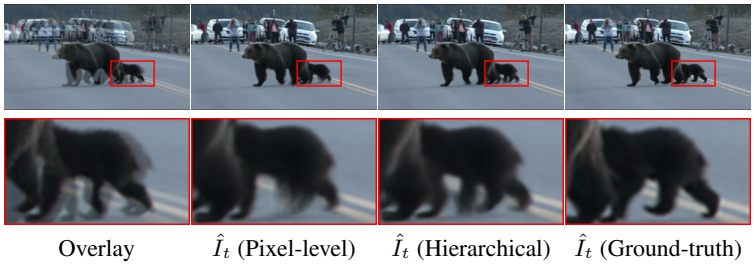

Fig. 7: Visualization of intermediate frame $\hat{I}_t$ under two settings: with pixel-level feature supervision and with hierarchical feature supervision. The results demonstrate that the hierarchical feature loss enables the model to produce complete details. **(See the leg of the bear)**.

5 Discussion and Future Work

Our framework supports two configurations to accommodate different accuracy–efficiency trade-offs. The full configuration combines HCE, HML, and HFL to achieve higher interpolation accuracy, at the cost of additional computation introduced by HCE and the segmentation model. In contrast, HML and HFL are used exclusively during training, providing object-aware motion supervision without introducing any additional inference overhead.

We further present representative failure cases in Fig. 8 that reveal two main limitations of the current framework. First, its performance depends on the quality of the object masks. In scenes involving severe occlusion or ambiguous object boundaries, inaccurate segmentation may introduce incorrect object-level priors and consequently degrade the interpolation results. This limitation could be alleviated by adopting more advanced segmentation models with stronger boundary localization and occlusion reasoning capabilities. Second, even when the object masks are accurate, the current model may still struggle with extremely large or complex motions. Such cases require stronger

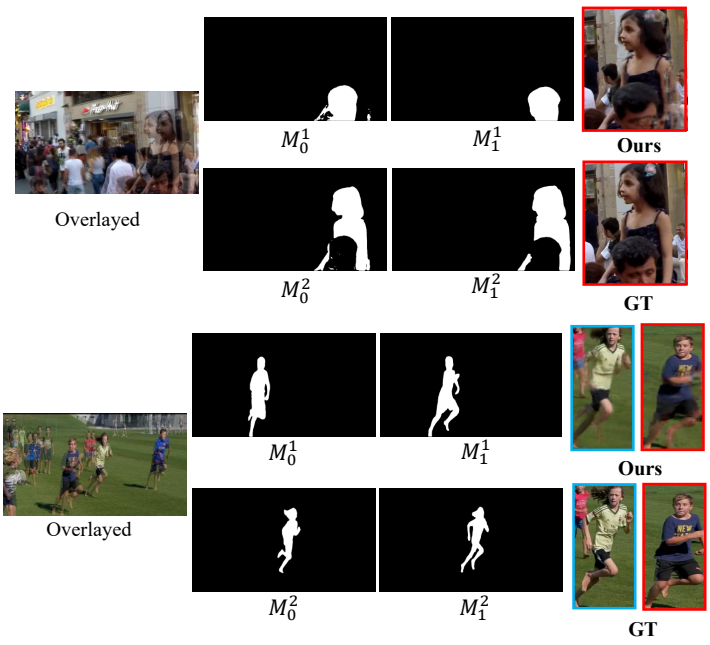

Fig. 8: Visualization of our failure cases.

motion modeling and a larger-capacity interpolation model to better capture long-range correspondences and substantial non-rigid displacements.

Our work is also complementary to Velocity Disambiguation [57, 58], which enables object-wise retiming through manually specified distance maps. In future work, we plan to extend explicit SAM priors from passive motion regularization to active object-level motion control, allowing users to specify spatial trajectories or non-rigid motion patterns for individual objects during frame interpolation. We will also investigate stronger segmentation models and more scalable motion-estimation backbones to improve robustness under inaccurate masks and extreme motion.

6 Conclusion

This paper introduces explicit object priors for VFI, effectively bringing object-level motion to pixel-level motion to enhance the accuracy and continuity of motion prediction through hierarchical learning. Specifically, we leverage Segment Anything Model to generate object masks. Additionally, we propose hybrid contextual feature extraction module to aggregate both pixel-wise and semantic representation, along with hierarchical motion loss and hierarchical feature loss to simulate current motion patterns. Extensive experiments demonstrate that integrating these three components into our framework outperforms SOTA method across multiple benchmark datasets.

References

1. Aleem, S., Wang, F., Maniparambil, M., Arazo, E., Dietlmeier, J., Curran, K., Connor, N.E., Little, S.: Test-time adaptation with salip: A cascade of sam and clip for zero-shot medical image segmentation. In: CVPR. pp. 5184–5193 (2024)
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: CVPR. pp. 3703–3712 (2019)
3. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: ICIP. vol. 2, pp. 168–172. IEEE (1994)
4. Cheng, H.K., Oh, S.W., Price, B., Schwing, A., Lee, J.Y.: Tracking anything with decoupled video segmentation. In: ICCV. pp. 1316–1326 (2023)
5. Cheng, X., Chen, Z.: Video frame interpolation via deformable separable convolution. In: AAAI. vol. 34, pp. 10607–10614 (2020)
6. Cheng, X., Chen, Z.: Multiple video frame interpolation via enhanced deformable separable convolution. TPAMI (2021)
7. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint arXiv:2305.06558 (2023)
8. Cheng, Z., Wei, Q., Zhu, H., Wang, Y., Qu, L., Shao, W., Zhou, Y.: Unleashing the potential of sam for medical adaptation via hierarchical decoding. In: CVPR. pp. 3511–3522 (2024)
9. Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M.: Channel attention is all you need for video frame interpolation. In: AAAI. pp. 10663–10671 (2020)
10. Deng, X., Wu, H., Zeng, R., Qin, J.: Memsam: Taming segment anything model for echocardiography video segmentation. In: CVPR. pp. 9622–9631 (2024)
11. Han, Y., Xu, X., Lin, Y., Wu, J., Liu, Z.: Video frame interpolation with region-distinguishable priors from sam. arXiv preprint arXiv:2312.15868 (2023)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. pp. 7132–7141 (2018)
13. Hu, M., Jiang, K., Zhong, Z., Wang, Z., Zheng, Y.: Iq-vfi: Implicit quadratic motion estimation for video frame interpolation. In: CVPR. pp. 6410–6419 (2024)
14. Hu, P., Niklaus, S., Sclaroff, S., Saenko, K.: Many-to-many splatting for efficient video frame interpolation. In: CVPR. pp. 3553–3562 (2022)
15. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: ECCV. pp. 624–642. Springer (2022)
16. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: CVPR. pp. 8981–8989 (2018)
17. Hur, J., Roth, S.: Joint optical flow and temporally consistent semantic segmentation. In: ECCV. pp. 163–177. Springer (2016)
18. 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: CVPR. pp. 9000–9008 (2018)
19. Jin, X., Wu, L., Chen, J., Chen, Y., Koo, J., Hahm, C.h.: A unified pyramid recurrent network for video frame interpolation. In: CVPR. pp. 1578–1587 (2023)
20. Kalluri, T., Pathak, D., Chandraker, M., Tran, D.: Flavr: Flow-agnostic video representations for fast frame interpolation. In: WACV. pp. 2071–2082 (2023)
21. Ke, L., Ye, M., Danelljan, M., Tai, Y.W., Tang, C.K., Yu, F., et al.: Segment anything in high quality. NeurIPS **36** (2024)
22. 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. In: ICCV. pp. 4015–4026 (2023)
23. 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: CVPR. pp. 1969–1978 (2022)

24. Lee, H., Kim, T., Chung, T.y., Pak, D., Ban, Y., Lee, S.: Adacof: Adaptive collaboration of flows for video frame interpolation. In: CVPR. pp. 5316–5325 (2020)
25. 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: CVPR. pp. 9801–9810 (2023)
26. Liu, C., Zhang, G., Zhao, R., Wang, L.: Sparse global matching for video frame interpolation with large motion. In: CVPR. pp. 19125–19134 (2024)
27. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: ICCV. pp. 4463–4471 (2017)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. ICLR (2017)
29. Lu, L., Wu, R., Lin, H., Lu, J., Jia, J.: Video frame interpolation with transformer. In: CVPR. pp. 3532–3542 (2022)
30. Ma, J., He, Y., Li, F., Han, L., You, C., Wang, B.: Segment anything in medical images. Nature Communications **15**(1), 654 (2024)
31. Meyer, S., Djelouah, A., McWilliams, B., Sorkine-Hornung, A., Gross, M., Schroers, C.: Phasenet for video frame interpolation. In: CVPR. pp. 498–507 (2018)
32. Meyer, S., Wang, O., Zimmer, H., Grosse, M., Sorkine-Hornung, A.: Phase-based frame interpolation for video. In: CVPR. pp. 1410–1418 (2015)
33. Niklaus, S., Liu, F.: Context-aware synthesis for video frame interpolation. In: CVPR. pp. 1701–1710 (2018)
34. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: CVPR. pp. 5437–5446 (2020)
35. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: CVPR. pp. 670–679 (2017)
36. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. In: ICCV. pp. 261–270 (2017)
37. Park, J., Kim, J., Kim, C.S.: Biformer: Learning bilateral motion estimation via bilateral transformer for 4k video frame interpolation. In: CVPR. pp. 1568–1577 (2023)
38. Park, J., Ko, K., Lee, C., Kim, C.S.: Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. ECCV (2020)
39. Park, J., Lee, C., Kim, C.S.: Asymmetric bilateral motion estimation for video frame interpolation. ICCV (2021)
40. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
41. Sevilla-Lara, L., Sun, D., Jampani, V., Black, M.J.: Optical flow with semantic segmentation and localized layers. In: CVPR. pp. 3889–3898 (2016)
42. Shi, Z., Liu, X., Shi, K., Dai, L., Chen, J.: Video frame interpolation via generalized deformable convolution. TMM (2021)
43. Shi, Z., Xu, X., Liu, X., Chen, J., Yang, M.H.: Video frame interpolation transformer. In: CVPR. pp. 17482–17491 (2022)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
45. 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)
46. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4), 600–612 (2004)
47. Wu, C.Y., Singhal, N., Krahenbuhl, P.: Video compression through image interpolation. In: ECCV. pp. 416–431 (2018)
48. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Tao, D.: Gmflow: Learning optical flow via global matching. In: CVPR. pp. 8121–8130 (2022)

49. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV **127**(8), 1106–1125 (2019)
50. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)
51. Yang, Z., Chen, H., Qian, Z., Zhou, Y., Zhang, H., Zhao, D., Wei, B., Xu, Y.: Region attention transformer for medical image restoration. In: MICCAI. pp. 603–613. Springer (2024)
52. Yuan, S., Luo, L., Hui, Z., Pu, C., Xiang, X., Ranjan, R., Demandolx, D.: Unsamflow: Unsupervised optical flow guided by segment anything model. In: CVPR. pp. 19027–19037 (2024)
53. Zhang, G., Liu, C., Cui, Y., Zhao, X., Ma, K., Wang, L.: Vfimamba: Video frame interpolation with state space models. arXiv preprint arXiv:2407.02315 (2024)
54. 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: CVPR. pp. 5682–5692 (2023)
55. Zhang, Q., Liu, X., Li, W., Chen, H., Liu, J., Hu, J., Xiong, Z., Yuan, C., Wang, Y.: Distilling semantic priors from sam to efficient image restoration models. In: CVPR. pp. 25409–25419 (2024)
56. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
57. Zhong, Z., Krishnan, G., Sun, X., Qiao, Y., Ma, S., Wang, J.: Clearer frames, anytime: Resolving velocity ambiguity in video frame interpolation. In: ECCV. pp. 346–363. Springer (2024)
58. Zhong, Z., Zhang, Y., Wang, W., Sun, X., Qiao, Y., Krishnan, G., Ma, S., Wang, J.: Velocity disambiguation for video frame interpolation. TPAMI (2026)