AMT: All-Pairs Multi-Field Transforms for Efficient Frame Interpolation

Zhen Li* Zuo-Liang Zhu* Ling-Hao Han Qibin Hou Chun-Le Guo† Ming-Ming Cheng
VCIP, CS, Nankai University
{zhenli1031, nkuzhuzl}@gmail.com, lhhan@mail.nankai.edu.cn
{houqb, guochunle, cmm}@nankai.edu.cn

Abstract

*We present **A**ll-Pairs **M**ulti-Field **T**ransforms (**AMT**), a new network architecture for video frame interpolation. It is based on two essential designs. First, we build bidirectional correlation volumes for all pairs of pixels, and use the predicted bilateral flows to retrieve correlations for updating both flows and the interpolated content feature. Second, we derive multiple groups of fine-grained flow fields from one pair of updated coarse flows for performing backward warping on the input frames separately. Combining these two designs enables us to generate promising task-oriented flows and reduce the difficulties in modeling large motions and handling occluded areas during frame interpolation. These qualities promote our model to achieve state-of-the-art performance on various benchmarks with high efficiency. Moreover, our convolution-based model competes favorably compared to Transformer-based models in terms of accuracy and efficiency. Our code is available at https://github.com/MCG-NKU/AMT.*

1. Introduction

Video frame interpolation (VFI) is a long-standing video processing technology, aiming to increase the temporal resolution of the input video by synthesizing intermediate frames from the reference ones. It has been applied to various downstream tasks, including slow-motion generation [22, 61], novel view synthesis [11, 29, 71], video compression [60], text-to-video generation [52], etc.

Recently, flow-based VFI methods [17, 22, 26, 34, 53, 69] have been predominant in referenced research due to their effectiveness. A common flow-based technique estimates bilateral/bidirectional flows from the given frames and then propagates pixels/features to the target time step via backward [2, 17, 26] or forward [14, 37, 38] warping. Thus, the quality of a synthesized frame relies heavily on flow estimation results. In fact, it is cumbersome to approximate intermediate flows through pretrained optical flow models,

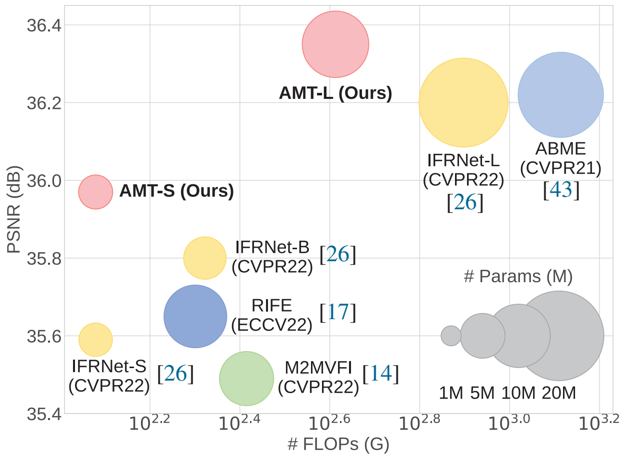

Figure 1. Performance vs. number of parameters and FLOPs. The PSNR values are obtained from the Vimeo90K dataset [65]. We use a 720p frame pair to calculate FLOPs. Our AMT outperforms the state-of-the-art methods and is with higher efficiency.

and these flows are unqualified for VFI usage [14, 17].

A feasible way to alleviate this issue is to estimate *task-oriented flows* in an end-to-end training manner [22, 26, 32, 65]. However, some major challenges, such as large motions and occlusions, are still pending to be resolved. These challenges mainly arise from the defective estimation of optical flows. Thus, a straightforward question should be: Why do previous methods have difficulties in predicting promising task-oriented flows when facing these challenges? Inspired by the recent studies [26, 65] that demonstrate *task-oriented flow is generally consistent with ground truth optical flow but diverse in local details*, we attempt to answer the above question from two perspectives:

(i) The flow fields predicted by existing VFI methods are *not consistent enough* with the true displacements, especially when encountering large motions (see Fig. 2). Existing methods mostly adopt the UNet-like architecture [48] with plain convolutions to build VFI models. However, this type of architecture is vulnerable to accumulating errors at early stages when modeling large motions [45, 56, 63, 70]. As a result, the predicted flow fields are not accurate.

(ii) Existing methods predict one pair of flow fields, re-

*Equal contribution
†C.L. Guo is the corresponding author.

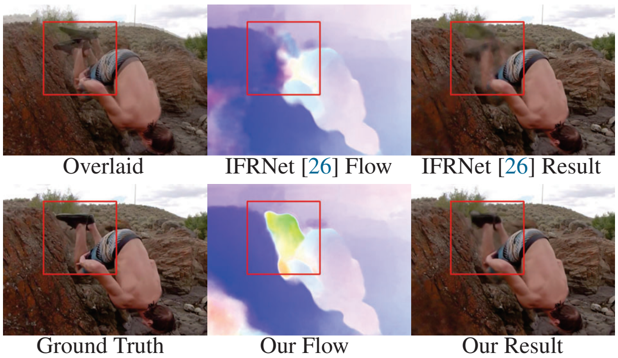

Figure 2. Qualitative comparisons of estimated flows and the interpolated frames. Our AMT guarantees the general consistency of intermediate flows and synthesizes fast-moving objects with occluded regions precisely, while the previous state-of-the-art IFRNet [26] fails to achieve them.

stricting the solution set in a tight space. This makes them struggle to handle occlusions and details around the motion boundaries, which consequently deteriorates the final results (see Fig. 2 and Fig. 5).

In this paper, we present a new network architecture, dubbed **A**ll-pairs **M**ulti-field **T**ransforms (**AMT**), for video frame interpolation. AMT explores two new designs to improve the fidelity and diversity of predicted flows regarding the above two main shortcomings of previous works.

Our first design is based on all-pairs correlation in RAFT [56], which adequately models the dense correspondence between frames, especially for large motions. We propose to build *bidirectional correlation volumes* instead of unidirectional one and introduce a *scaled* lookup strategy to solve the coordinate mismatch issue caused by the invisible frame. Besides, the retrieved correlations assist our model in *jointly* updating bilateral flows and the interpolated content feature in a *cross-scale* manner. Thus, the network guarantees the fidelity of flows across scales, laying the foundation for the following refinement.

Second, considering that predicting one pair of flow fields is hard to cope with the occlusions, we propose to derive multiple groups of fine-grained flow fields from one pair of updated coarse bilateral flows. The input frames can be separately backward warped to the target time step by these flows. Such diverse flow fields provide adequate potential solutions for each pixel to be interpolated, particularly alleviating the ambiguity issue in the occluded areas.

We examine the proposed AMT on several public benchmarks with different model scales, showing strong performance and high efficiency in contrast to the state-of-the-art (SOTA) methods (see Fig. 1). Our small model outperforms IFRNet-B, a SOTA lightweight model, by +0.17dB PSNR on Vimeo90K [65] with only 60% of its FLOPs and parameters. For large-scale setting, our AMT exceeds the previous SOTA (*i.e.,* IFRNet-L) by +0.15 dB PSNR on Vimeo90K [65] with 75% of its FLOPs and 65% of its parameters. Besides, we provide a huge model for comparison with the SOTA transformer-based method VFIFormer [34]. Our convolution-based AMT shows a comparable performance but only needs nearly 23× less computational cost compared to VFIFormer [34]. Considering its effectiveness, we hope our AMT could bring a new perspective for the architecture design in efficient frame interpolation.

2. Related Work

Video Frame Interpolation: The development of deep learning has spawned a large amount of VFI methods. These methods can be roughly divided into three categories: kernel-based [5,6,27,39,40,44], hallucination-based [8,23,25,50], and flow-based ones [1,2,17,26,32,37,38,64,65].

Kernel-based methods attempt to capture motion with dynamic kernel weights [39,40,44] or/and offsets [5,6,10,27]. With the help of off-the-shelf architectures [9,13,49,57], hallucination-based methods directly generate the interpolated frame from features of input pairs. Thanks to the robustness of optical flow, flow-based methods have become mainstream in VFI. Previous methods resort to a pretrained flow model [37,64] or a jointly trained estimation module [26,32,65] to obtain the flow estimation. For generating task-oriented flows, some methods [17,26] propose intermediate supervisions to distill motion knowledge from the pseudo ground truth. Subsequently, backward warping [2,17,26] and forward warping [37,38,41] are standard schemes in the usage of estimated flows. A UNet-like architecture is a common choice [1, 2, 37, 38] to obtain the final synthesized frame, and the transformer [31,58], as a prevailing architecture, is introduced [34,50,68] for a better synthesis. Recent works [26,46] discard an independent synthetic network in consideration of efficiency. However, these methods suffer from the inability in modeling large motions and in dealing with occlusions.

Task-Oriented Flow: Initially, flow-based video processing methods [1, 22, 42, 64] estimate flows and process images individually. However, this two-step pipeline ignores the gap between true optical flow with task-specific objectives, which could be suboptimal for a specific task. ToFlow [65] proposes the concept of task-oriented flow, facilitating the development of video processing methods [3,15,28,30,66] significantly. Typically, the VFI-oriented flow is generally consistent with the true flow while diverses in detail (*e.g.,* occluded regions). Super Slomo [22] introduces a mask to handle the occlusion explicitly and provides a standard formulation for synthesizing intermediate frames, which utilizes by following methods [7, 17, 26, 51] up to now. IFRNet [26] and RIFE [17] propose task-oriented flow distillation losses to provide a prior of intermediate flow in training. Different from them, we consider the estimation of task-oriented flows from *the perspective of architecture design*. We introduce all-pairs correlation to strengthen the

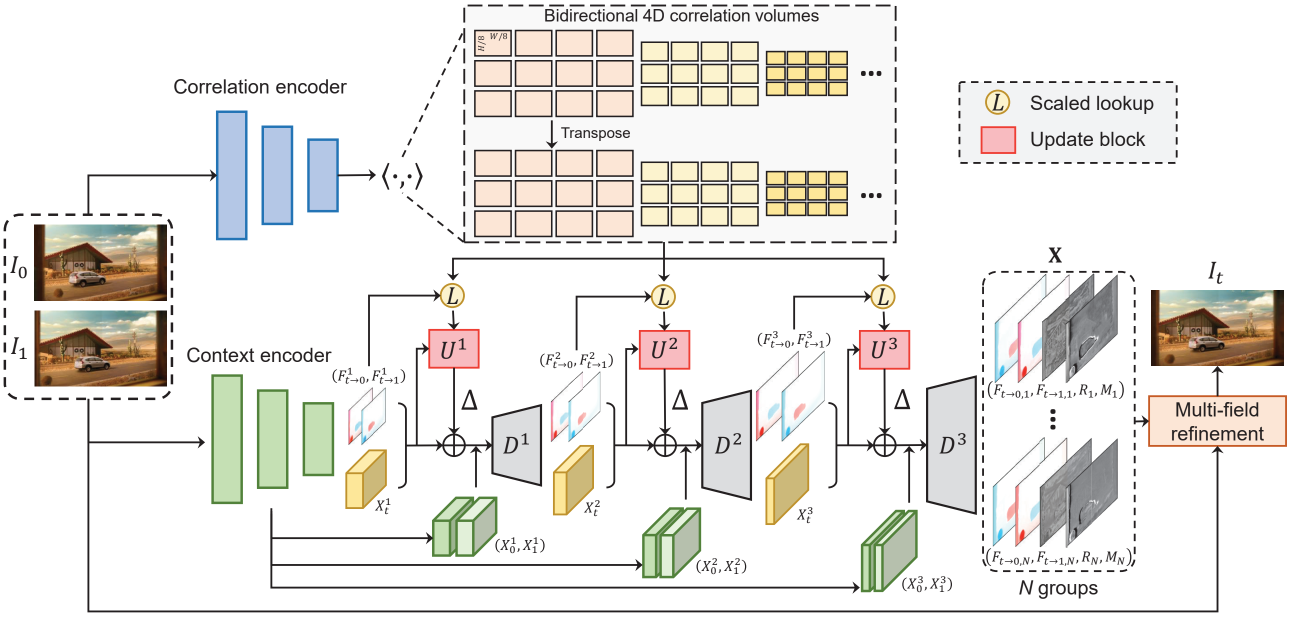

Figure 3. Architecture overview of the proposed AMT. Firstly, the input frames are sent to the correlation encoder to extract features, which are used to construct bidirectional correlation volumes. Then, the context encoder extracts pyramid features of visible frames and generates initial bilateral flows and interpolated intermediate feature. Next, we use bilateral flows to retrieve bidirectional correlations for jointly updating flow fields and the intermediate feature at each level. Finally, we generate multiple groups of flow fields, occlusion masks, and residuals based on the coarse estimate for interpolating the intermediate frame.

ability in motion modeling, which guarantees the consistency of flows on the coarse scale. At the finest scale, we employ multi-field refinement to ensure the diversity for the flow regions that need to be task-specific.

Cost Volume: Cost volume is introduced as a representation of matching costs in numerous vision tasks [12, 20, 24]. In the deep learning era, the concept of cost volume is also proved to be effective in optical flow estimation [16, 19, 55, 56, 67]. Among these works, the most influential ones are PWC-Net [55] and RAFT [56]. In VFI, the existing methods [21, 42, 43, 62] attempt to introduce the cost volume following the scheme of PWC-Net. However, those methods not only search the cost volume in a local region but also depend on inaccurate features warped from reference ones, resulting in a limited performance gain from the cost volume. Instead, the proposed AMT is based on RAFT, which enlarges the search space by iteratively updating the flow field with all-pairs correlation, and only constructs cost volumes between the visible frames. Besides, we involve many *novel* and *task-specific* designs beyond RAFT. The details are described in Sec. 3 and our supplement.

3. Method

Given a pair of input frames (I_0, I_1), we aim to synthesize an intermediate frame I_t at a target time step t, where $0 < t < 1$. Our AMT is a one-stage flow-based method, in which bilateral flows and the interpolated intermediate feature are updated and upsampled jointly. As shown in Fig. 3, it is composed of three main components: 1) an encoder for extracting features and initial bilateral flows simultaneously, 2) multi-scale bidirectional correlation volumes for jointly updating bilateral flows and intermediate features at coarse scales, and 3) a multi-field refinement operator for interpolating the target frame with multiple flow groups at the finest scale. Benefiting from such designs, the estimated motion vectors at coarser scales are generally consistent with the ground truth displacements. Meanwhile, they are diverse in fine-grained details at the finest scale, which meets the requirement of *task-oriented* flow. These designs also enable our AMT to capture large motions and successfully handle occlusion regions with high efficiency.

3.1. Initial Flow and Feature Extraction

We employ two separate feature extractors. They are applied to the input pair (I_0, I_1), but for different purposes. The first is the *correlation encoder*, which maps the input frames to a pair of dense features for constructing bidirectional correlation volumes. We can obtain the pair of features $\mathbf{g}_0, \mathbf{g}_1 \in \mathbb{R}^{H/8 \times W/8 \times D}$ at 1/8 the input image resolution with D channels.

The second is the *context encoder*, which outputs the initial interpolated intermediate feature X_t^1 and predicts the initial bilateral flows $F_{t\to 0}^1$ and $F_{t\to 1}^1$. Their spatial resolution is the same as the output of the correlation encoder. Besides, the pyramid features $\{X_0^l, X_1^l \mid l \in \{1, 2, 3\}\}$ for frames I_0, I_1 are extracted by context encoder for further progressive warping. The architectural details of them can be found in our supplement.

3.2. All-Pairs Correlation

Bidirectional Correlation Volumes: Similar to RAFT [56], we compute the dot-product similarities between all pairs of features vectors for constructing a 4D correlation volume. Given the pair of features $\mathbf{g}_0, \mathbf{g}_1$, we can obtain the correlation volume $\mathbf{C}$ through:

$$\mathbf{C}_{ijkl} = \sum_h \mathbf{g}_{0,ijh} \cdot \mathbf{g}_{1,klh}, \quad \mathbf{C} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times \frac{H}{8} \times \frac{W}{8}} \tag{1}$$

For further measuring similarities across scales, the last two dimensions of the correlation volume are downsampled by a repeated 2D average pooling layer with a kernel size of 2 and a stride of 2. We thus obtain a 4-level correlation pyramid $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\}$.

However, the correlation pyramid in RAFT is *unidirectional*. It only reflects multi-scale correspondences from I_0 to I_1. We thereby term it as the forward correlation pyramid. The unidirectional correspondence is insufficient for the VFI task, as the motions are usually asymmetric [43, 64]. Instead of recomputing the matrix multiplication, we directly transpose the correlation volume $\mathbf{C}$ to represent the correspondence in the opposite direction. After obtaining the transposed correlation volume $\mathbf{C}^T$, we perform the same pooling operation to form the backward correlation pyramid $\{\mathbf{C}_1^T, \mathbf{C}_2^T, \mathbf{C}_3^T, \mathbf{C}_4^T\}$. Note that the bidirectional correlation volumes only need to compute once. The compact global representations assist our network in being aware of large motions in an efficient way.

Correlation Scaled Lookup: After constructing the bidirectional correlation volumes, we intend to query correlation feature maps using estimated bilateral flows $F^l_{t\to0}$ and $F^l_{t\to1}$. In RAFT, the lookup operation can be directly performed since its estimated flow and the correlation volume share an *identical* coordinate system. For example, the motion $F_{0\to1}$ from frame 0 to frame 1 and the corresponding correlation volume are all based on the coordinate system of the frame 0. Thus, the correlation feature maps can be correctly sampled by the matched flow field. However, for frame interpolation, we can only build correlation volumes from visible reference frames (*i.e.,* I_0, I_1) but estimate the flows (*i.e.,* $F^l_{t\to0}$ and $F^l_{t\to1}$) of an invisible intermediate frame I_t. So there exists *a mismatch between coordinate systems*, which causes unfaithful correlation lookups and further influences the updating of the flows. A straightforward solution to this problem is transferring bilateral flows $F^l_{t\to0}$ and $F^l_{t\to1}$ to bidirectional flows $F^l_{0\to1}$ and $F^l_{1\to0}$.

To achieve this goal, we simply scale the estimated bilateral flows based on locally smooth motion assumption [22,42,43]. Specifically, we assume the moving objects are partially overlap within a small time interval. Thus, the bilateral flows and bidirectional flows at the same position are generally consistent in direction but different in magnitude. So the bidirectional flows $F^l_{0\to1}$ and $F^l_{1\to0}$ can be approximated by:

$$F^l_{0\to1} = \frac{1}{1-t} F^l_{t\to1}, \quad F^l_{1\to0} = \frac{1}{t} F^l_{t\to0}. \tag{2}$$

Subsequently, a lookup operation analogous to that in RAFT performs on bidirectional correlation volumes through approximated bidirectional flows. We construct two lookup windows centered by bidirectional flows with a predefined radius. The lookup operations in the windows are conducted on all levels of the bidirectional correlation pyramids. The retrieved bidirectional correlations are concatenated into one features map for further jointly updating bilateral flows and the interpolated intermediate feature.

Updating with Retrieved Correlations: While RAFT updates and maintains the flow prediction at a single resolution, we predict the bilateral flows in a coarse-to-fine manner following most flow-based VFI methods [17,26,37,38]. This is because that the features of the input pair need to be progressively warped based on the latest flow predictions for generating a faithful intermediate feature. Given the reciprocal relationship between bilateral flow fields and intermediate features in VFI task [17, 26, 32], we also update and upsample the intermediate feature along with the intermediate motions.

Specifically, during the update stage at each spatial level l, we employ an update block to jointly predict the residuals of the bilateral flow fields $F^l_{t\to0}, F^l_{t\to1}$ and the interpolated intermediate feature X^l_t based on the retrieved bidirectional correlations. In each update block, the bidirectional correlation features and bilateral flows are first passed through two convolutional layers. Then, they are concatenated with the interpolated intermediate feature and injected into two convolutional layers instead of a cumbersome GRU unit in RAFT. Finally, the output features are sent to two separate heads for predicting bilateral flow residuals $\Delta F^l_{t\to0}, \Delta F^l_{t\to1}$ and an interpolated feature residual ΔX^l_t. Each head is formed by two convolutional layers.

Note that the spatial dimension of the retrieved correlation features is the same as the first two dimensions of the correlation volume (*i.e.,* $\frac{H}{8} \times \frac{W}{8}$) but is different from that of the intermediate features and motions on upper levels. We thus need to downscale the flow fields and the intermediate feature accordingly before feeding them into the update block and upsample the predicted residuals for updating. Through downscaling, the update block works at a low-resolution space, leading to promising efficiency. The updated intermediate feature $\hat{X}^l_t$ can be formulated as: $\hat{X}^l_t = X^l_t + \Delta X^l_t$, where ΔX^l_t is the output content residual of the update block. The updated bilateral flows $\hat{F}^l_{t\to0}, \hat{F}^l_{t\to1}$ can be obtained following the same rule.

We employ the updated bilateral flows to warp the features X_0^l, X_1^l of the input frames. Let $\hat{X}_0^l, \hat{X}_1^l$ denote the warped features. The warped features, the updated bilateral flows, and the updated intermediate feature are concatenated together and then fed into the l-th decoder. The l-th decoder D^l predicts the upsampled bilateral flows $F_{t\to0}^{l+1}, F_{t\to1}^{l+1}$ and the intermediate feature X_t^{l+1} simultaneously as follows:

$$[F_{t\to0}^{l+1}, F_{t\to1}^{l+1}, X_t^{l+1}] = D^l([\hat{X}_0^l, \hat{X}_1^l, \hat{F}_{t\to0}^l, \hat{F}_{t\to1}^l, \hat{X}_t^l]). \tag{3}$$

Specially, the Eqn. (3) does not consider the last decoder D^3, which is responsible for generating multiple flow fields and occlusion masks for task-specific usage. The architecture details of each decoder are listed in our supplement.

3.3. Multi-Field Refinement

In flow-based VFI methods, the common formulation for interpolating the final intermediate frame is:

$$I_t = M \odot \mathcal{W}(I_0, F_{t\to0}) + (1-M) \odot \mathcal{W}(I_1, F_{t\to1}) + R, \tag{4}$$

where $\mathcal{W}$ denotes the backward warping operation, $\odot$ means the element-wise multiplication. M is an estimated occlusion mask which ranges from 0 to 1. $F_{t\to0}$ and $F_{t\to1}$ are final predictions of bilateral flows. R is the estimated residual. Such formulation considers temporal consistency and occlusion reasoning, synthesizing the intermediate frame efficiently. However, only predicting one pair of flow fields ignores that each location in the occlusion areas has many potential pixel candidates, restricting the solution set for interpolation in a tight space.

Based on previously predicted coarse flows, which are generally consistent with the ground truth displacements, we derive multiple fine-grained flow fields for task-specific usage. We also jointly estimate a residual content and an occlusion mask for each pair of optical flow. This process can be formulated as:

$$\begin{aligned} \mathbf{X} &= D^3([\hat{X}_0^3, \hat{X}_1^3, \hat{F}_{t\to0}^3, \hat{F}_{t\to1}^3, \hat{X}_t^3]), \\ \mathbf{X} &= \{F_{t\to0,n}, F_{t\to1,n}, M_n, R_n | n \in \{1, 2, ..., N\}\}, \end{aligned} \tag{5}$$

where N denotes the total number of output groups. $(F_{t\to0,n}, F_{t\to1,n})$, M_n, and R_n are the n-th estimated bilateral flows, occlusion mask, and residual content, respectively. Notably, Eqn. (5) can be easily implemented by enlarging the output channels of the last decoder according to the number of flow pairs, which ensures efficiency. The final intermediate frame can be obtained by:

$$I_t = \mathcal{C}([I_t^1, ..., I_t^N]), \tag{6}$$

where the n-th interpolated frame I_t^n can be obtained by Eqn. (4) with corresponding output group. We stack two convolutional layers (denoted as $\mathcal{C}$) for adaptively merging candidate frames and refining the final results. The analyses of multiple flow fields are detailed in Sec. 4.4.2.

3.4. Loss Functions

There are three losses involved in our AMT. To better predict task-oriented flows, we employ flow distillation loss $\mathcal{L}_{flow}$ in IFRNet [26], which concentrates more on the flow regions that are easy to be reconstructed, but slightly penalizes the regions that are difficult to recover. This loss is applied on updated multi-scale flow fields except for the finest flow predictions left for fully task-specific usage. The Charbonnier loss [4] $\mathcal{L}_{char}$ and the census loss [35] $\mathcal{L}_{css}$ are used to supervise the content generation of the interpolated frame. The former measures the pixel-wise errors between the ground truth intermediate frame I_t^{GT} and the generated one I_t, and the latter calculates the soft Hamming distance between census-transformed image patches of I_t^{GT} and I_t.

The full objective can be defined as:

$$\mathcal{L} = \lambda_{char}\mathcal{L}_{char} + \lambda_{css}\mathcal{L}_{css} + \lambda_{flow}\mathcal{L}_{flow}, \tag{7}$$

where λ_{char}, λ_{css}, and λ_{flow} are weights for each loss.

4. Experiments

4.1. Training Details

We train AMT on Vimeo90K [65] training set for 300 epochs with AdamW [33] optimizer on 2 NVIDIA RTX 3090 GPUs. The total batch size is 24, and the learning rate decay follows the cosine attenuation schedule from 2×10^{-4} to 2×10^{-5}. We follow the augmentation pipeline including random flipping, rotating, reversing sequence order, and random cropping patches with size 224×224 in IFRNet [26]. The flow predictions from the pre-trained LiteFlowNet [18] are served as the pseudo ground truth label for supervising the intermediate flows. λ_{char}, λ_{css}, and λ_{flow} are set as 1, 1, and 0.002, respectively. The code implemented by MindSpore framework is also provided.

4.2. Benchmarks

We evaluate our AMT on various benchmarks containing diverse motion scenes for a comprehensive comparison. PSNR and SSIM [59] as common evaluation metrics are utilized for comparison. The statistics of benchmarks used in the main paper are presented as follows.

Vimeo90K [65]: Vimeo90K is the most commonly used evaluation benchmark in recent VFI literature. There are 3,782 triplets of 448×256 resolution in the test part.

UCF101 [54]: UCF101 dataset contains videos with various human actions, and we adopt the test partition in DVF [32], which consists of 379 triplets of 256×256 size.

SNU-FILM [8]: SNU-FILM dataset contains 1,240 frame triplets, whose width ranges from 368 to 720 and height ranges from 384 to 1280. With respect to motion magnitude, it is partitioned into four exclusive parts, namely Easy, Medium, Hard, and Extreme.

Xiph [36]: Xiph dataset, consisting of eight video clips with a 4K resolution, was originally proposed by Niklaus

Method	Vimeo90K [65]	UCF101 [54]	SNU-FILM [8]				Xiph [36]		Latency	Params	FLOPs
			Easy	Medium	Hard	Extreme	2K	4K	(ms/f)	(M)	(T)
AdaCoF [27]	34.38/0.972	35.20/0.970	39.85/0.991	35.08/0.976	29.47/0.925	24.31/0.844	34.86/0.928	31.68/0.870	52	21.8	0.36
M2M-VFI [14]	35.49/0.978	35.32/0.970	39.66/0.991	35.74/0.980	30.32/0.936	25.07/0.860	**36.44/0.943**	33.92/0.899	40	7.6	0.26
RIFE [17]	35.65/0.978	35.28/0.969	**40.06**/0.991	35.75/0.979	30.10/0.933	24.84/0.853	36.19/0.938	33.76/0.894	29	9.8	0.20
IFRNet-S [26]	35.59/0.979	35.28/0.969	39.96/0.991	35.92/0.979	30.36/0.936	25.05/0.858	35.87/0.936	33.80/0.891	25	2.8	0.12
IFRNet-B [26]	35.80/0.979	35.29/0.969	40.03/0.991	35.94/0.979	30.41/0.936	25.05/0.859	36.00/0.936	33.99/0.893	30	5.0	0.21
AMT-S	**35.97/0.983**	**35.35/0.971**	39.95/**0.994**	**35.98/0.983**	**30.60/0.940**	**25.30/0.865**	36.11/0.940	**34.29/0.901**	51	3.0	0.12
ToFlow [65]	33.73/0.968	34.58/0.967	39.08/0.989	34.39/0.974	28.44/0.918	23.39/0.831	33.93/0.922	30.74/0.856	88	1.4	0.62
DAIN [1]	34.71/0.976	34.99/0.968	39.73/0.990	35.46/0.978	30.17/0.934	25.09/0.858	35.95/0.940	33.49/0.895	664	24.0	5.51
CAIN [8]	34.78/0.974	35.00/0.969	39.95/0.990	35.66/0.978	29.93/0.930	24.80/0.851	35.21/0.937	32.56/0.901	71	42.8	1.29
BMBC [42]	35.01/0.976	35.15/0.969	39.90/0.990	35.31/0.977	29.33/0.927	23.92/0.843	32.82/0.928	31.19/0.880	2234	11.0	2.50
ABME [43]	36.22/0.981	35.41/0.970	39.59/0.990	35.77/0.979	30.58/0.937	**25.42/0.864**	**36.53/0.944**	33.73/0.901	560	18.1	1.30
IFRNet-L [26]	36.20/0.981	35.42/0.970	**40.10**/0.991	**36.12**/0.980	30.63/0.937	25.27/0.861	36.21/0.937	34.25/0.895	80	19.7	0.79
AMT-L	**36.35/0.982**	**35.42/0.970**	39.95/**0.991**	36.09/**0.981**	**30.75/0.938**	25.41/0.864	36.27/0.940	**34.49/0.903**	116	12.9	0.58
VFIFormer [34]	36.50/0.982	35.43/0.970	**40.13**/0.991	36.09/0.980	30.67/0.938	25.43/0.864	OOM	OOM	1293	24.1	47.71
EMA-VFI† [68]	36.50/0.980	35.42/0.970	39.58/0.989	35.86/0.979	**30.80**/0.938	**25.59**/0.864	**36.74/0.944**	34.55/**0.906**	211	66.0	0.91
AMT-G	**36.53/0.982**	**35.45/0.970**	39.88/**0.991**	**36.12/0.981**	30.78/ **0.939**	25.43 /**0.865**	36.38/0.941	**34.63**/0.904	250	30.6	2.07

Table 1. Quantitative comparison with SOTA methods. We divide the existing methods into three groups, according to the computational complexity. For each group, the best result is shown in **bold**, and the second best is underlined. "OOM" denotes the out-of-memory issue when evaluating on an NVIDIA RTX 3090 GPU. † means we disable the test-time augmentation [17] for a fair comparison.

et al. [38]. Following their original evaluation setting, we reform this dataset to include "2K" version, obtained by downscaling original frames, and "4K" version, created by center-cropping 2K patches.

Except for these datasets, we provide the comparisons of multi-frame interpolation in the supplement.

4.3. Comparison with the SOTAs

We compare our AMT with the state-of-the-art (SOTA) methods, including ToFlow [65], DAIN [1], CAIN [8], AdaCoF [27], BMBC [42], RIFE [17], ABME [43], M2M-VFI [14], IFRNet [26], VFIFormer [34], and EMA-VFI [68]. We utilize the code provided by IFRNet [26] for benchmarks. The inference latency is the average running time of a method on 1280×720 resolution for 1000 iterations on an NVIDIA RTX 3090 GPU. To ensure a fair comparison, we group the SOTA methods into three categories based on their theoretical computational complexity. We then develop three models, called AMT-S, AMT-L, and AMT-G, for each group.

Quantitative Comparison. As shown in Tab. 1, our small model AMT-S achieves the best results among efficient VFI methods on almost all benchmarks, especially for challenging settings. Specifically, Our AMT-S outperforms the previous state-of-the-art method in effective VFI, IFRNet-B [26], by 0.17dB on Vimeo90K while using only about 60% of its parameters and FLOPs. This gap becomes more obvious on the Hard and Extreme partitions in SNU-FILM, revealing the strong ability of our AMT in modeling large motions. For the large scale setting, our AMT-L shows highly competitive results in contrast to the previous SOTA method IFRNet-L [26], with about 65% parameters and 75% FLOPs of it. In terms of inference speed, our method is comparable to IFRNet. Besides, our convolution-based model competes favorably compared to the SOTA Transformer-based models (*i.e.*, VFIFormer [34] and EMA-VFI [68]) in terms of accuracy and efficiency. Specifically, our AMT-G outperforms them in most casess, particularly when evaluated using the SSIM metric. Notably, our model achieves about 5× faster inference speed than VFIFormer and has only half the number of parameters of EMA-VFI. It is important to note that VFIFormer requires a two-stage training pipeline and 600 training epochs, while our model only requires 300 epochs. Additionally, EMA-VFI introduces a warm-up technique during training, which our method does not utilize. We observe that the performance of our method is saturated except for the Vimeo90K dataset after increasing the scale of the model to a huge version, which may indicate the overfitting problem.

Qualitative Comparison. In Fig. 4, we select the representative hallucination-based, kernel-based, and flow-based methods, including CAIN [8], AdaCoF [27], ABME [43], RIFE [17], and IFRNet(-B/-L) [26]. We compare them with our AMT on SNU-FILM [8] (Hard) dataset for visual comparison. It can be seen that previous VFI methods fail to provide sharp edges of moving objects, especially when the motion is complex. Due to our thorough consideration of VFI-oriented flows, our AMT synthesizes the content at motion boundaries faithfully and generates plausible textures with fewer artifacts. When the background objects are heavily occluded by the foreground unilaterally, our AMT can still obtain guidance from the reference frame in another direction, while other methods are unable to synthesize these occluded objects. We provide more comparisons in the supplement.

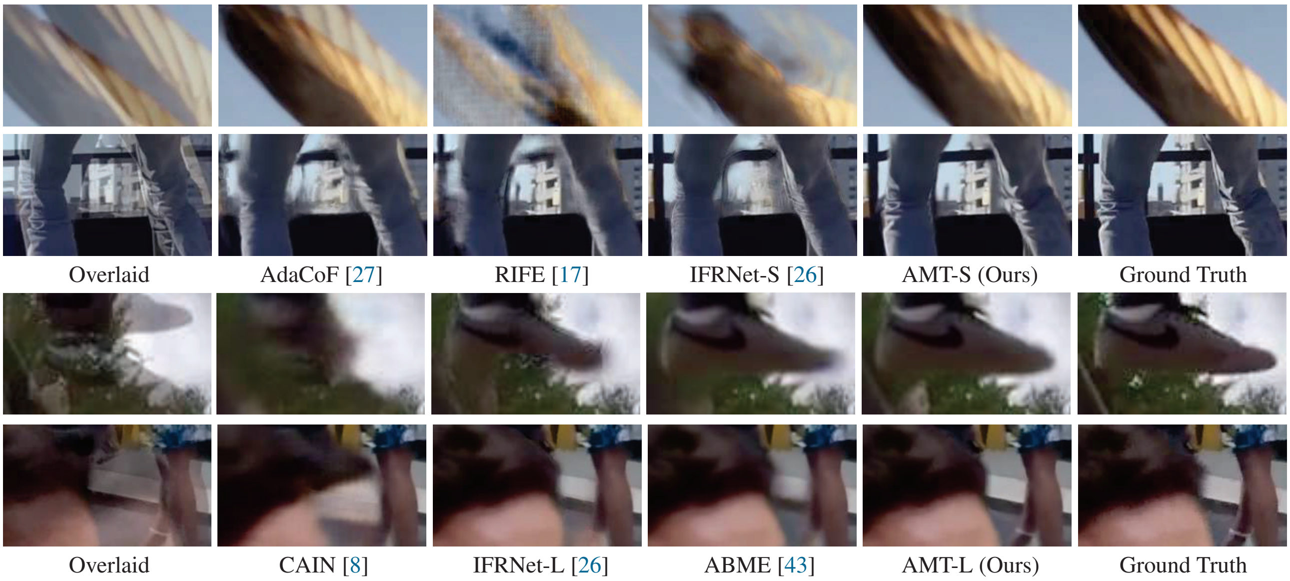

Figure 4. Qualitative results from different VFI methods. We divide these methods into two groups by computational cost. Our AMT-S and AMT-B synthesize precise boundaries of the objects with large motion and can reconstruct occluded regions with high fidelity.

Case	Vimeo	Hard	Extreme
w/o Corr. Enc.	35.76	30.49	25.22
Unidir. CV	35.93	30.34	25.18
PWC CV	35.61	30.48	25.16
Full Model	**35.97**	**30.60**	**25.30**

(a) **Correlation volume (CV) design.** We remove the correlation encoder ('w/o Corr. Enc.'), build a unidirectional CV ('Unidir. CV'), and build PWC-like [47] CV ('PWC CV') for ablations, respectively.

Lookup	Init	Vimeo	Hard	Extreme
Initial meshgrid		35.92	30.52	25.23
RAFT	Flow	35.93	30.34	25.18
Scaled	Zero	35.97	30.56	25.26
Scaled	Flow	**35.97**	**30.60**	**25.30**

(b) **Lookup strategy**. We investigate the initial meshgrid, RAFT-like [56] lookup ('RAFT'), and the proposed lookup ('Scaled') variants. We also investigate whether we use bilateral flows to perform an initial lookup.

Case	Vimeo	Hard	Extreme
Vanilla Guide	35.95	30.53	25.21
w/o Update	35.96	30.52	25.22
Full Model	**35.97**	**30.60**	**25.30**

(c) **Content update.** We investigate the content update by using features from visible frames as guidance ('Vanilla Guide') and discarding the content update ('w/o Update'), respectively.

1st	2nd	3rd	Vimeo	Hard	Extreme
			35.60	30.39	25.06
✔			35.84	30.55	25.19
✔	✔		35.92	30.58	25.28
✔	✔	✔	**35.97**	**30.60**	**25.30**
single-scale			35.95	30.50	25.22

(d) **Cross-scale update**. We investigate the impact of the update at different levels.

No.	Vimeo	Hard	Extreme	FLOPs (G)
1	35.84	30.52	25.25	116
3	35.97	30.60	25.30	121
5	36.00	**30.63**	**25.33**	127
7	**36.01**	30.57	25.25	135

(e) **Number of fields**. We investigate different numbers of flow pairs.

Case	Vimeo	Hard	Extreme
w/o Residual	35.87	30.57	25.27
w/o Refine	35.89	30.51	25.19
Full Model	**35.97**	**30.60**	**25.30**

(f) **Multi-field combination.** We investigate the residual component in Eqn. (4) and the refinement step in Eqn. (6).

Table 2. Ablation experiments of AMT on Vimeo90K [65] and SNU-FILM [8] (Hard, Extreme) dataset. We report the PSNR values of these variants, and the best result is shown in **bold**. The default setting is marked in gray.

4.4. Ablation Study

We conduct ablations to verify the effectiveness of two key components (*i.e.,* all-pairs correlation and multi-field refinement) in our AMT. All ablated versions are based on the AMT-S and evaluated on Vimeo90K [65] and the Hard and Extreme partitions of SNU-FILM [8].

4.4.1 All-Pairs Correlation

Volume Designs. As illustrated in Tab. 2a, our bidirectional correlation volumes thoroughly consider the correspondences between input frames for the VFI task, leading to a better performance than the unidirectional one. Besides, using an exclusive encoder (*i.e.,* correlation encoder) for building the correlation volumes is necessary. We can observe that the performance heavily drops when we utilize features from the context encoder to construct the correlation volumes. We also try to build the correlation volumes following PWC-Net [55]. This variant performs worse than any other one, for its partial correlation volume limits the ability in modeling motion information sufficiently.

Lookup Strategy. As shown in Tab. 2b, we can observe an obvious performance drop while utilizing the vanilla lookup strategy in RAFT [56]. For large motions, its performance is even worse than the one that directly uses the initial meshgrid, which indicates this strategy provides unfaithful correlation information for flow update. After we project the

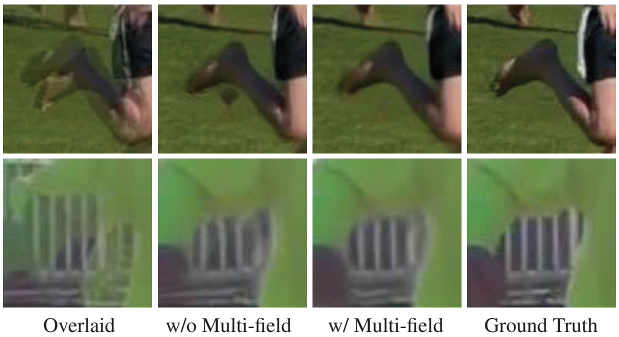

Figure 5. Effect of multi-field refinement. Multi-field refinement helps the network recover occluded regions better.

flows by scaling, the correlation volumes and flows share the identical coordinate system, and the network takes advantage of the correct lookup process. Besides, the initial flow pair from the context encoder gives a good initial point for further lookup, which brings a performance gain.

Content Update. In our AMT, each update block receives the intermediate content features as the context guidance and updates it along with bilateral flows. If we replace the context guidance with features from visible frames, the ambiguous information will be introduced, leading to a performance drop, as shown in Tab. 2c. Besides, we only keep one head in each update block for only updating the flow fields without updating the intermediate feature, resulting in the decrease of PSNR values on large motions. It demonstrates that all-pairs correlation is not only helpful for updating flows but also for updating content.

Update Strategy. As shown in Tab. 2d, all updates across levels are effective in our cross-scale update strategy. It is worth noticing that if we discard all updates, which is equivalent to a model without all-pairs correlation, the PSNR value will decrease dramatically. This demonstrates *the effectiveness of all-pairs correlation* in our AMT. Besides, only updating on the 1-st scale with $3\times$ iterations degrades the performance. The fact indicates that the cross-scale update strategy can take full advantage of progressively refined content features, leading to better motion modeling.

4.4.2 Multi-field Refinement

Number of Flow Fields. Tab. 2e illustrates the performance gain with respect to the number of flow fields. We observe that just using three pairs of flows bring a notable performance gain, which reveals that ensuring the diversity of flow fields is significant for VFI-oriented usage. The PSNR values rises in pace with the increase of field number until 7 pairs, which indicates saturation. We employ 3 pairs in our small model for efficiency (*i.e.,* AMT-S) and use 5 pairs in the larger models for better performance. In Fig. 5, we investigate the effect of multi-field refinement on occlusion handling. The results indicate that after employing multi-field refinement, our AMT can synthesize the background occluded by the foreground with more consistent textures.

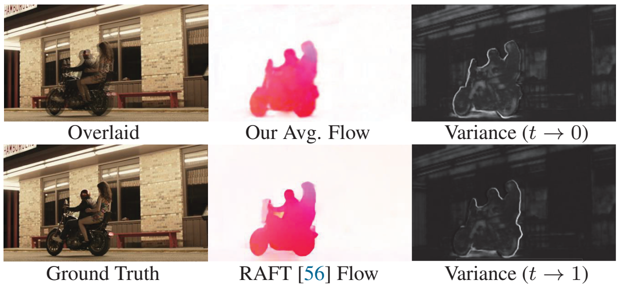

Figure 6. Visualizations of average and variance map of three flow pairs. We provide RAFT [56] flow for reference.

Multi-Field Combination. We investigate a variant that removes the residual component for each candidate frame in Eqn. (4) but estimates the residual part in the final interpolation result. As shown in Tab. 2f, the results of this variant underperform the original setting, which indicates we need to compensate details for each frame candidate separately. Besides, if we replace the convolution operators in Eqn. (6) with an average operation, the performance will be degraded (see Tab. 2f). This indicates that it is important for our AMT to perform an adaptive fusion and refinement.

Discussion. For further discussion, we visualize the mean and deviation of three estimated flow pairs. The results are shown in Fig. 6. On the one hand, our average flow is generally consistent with the flow estimated from RAFT [56], which approximates to the ground truth displacements. On the other hand, we observe that the major diversities of flows are at the motion boundaries and in the regions with rich textures. This indicates that these regions need to involve more potential pixel candidates for reconstruction. Through these visualizations, we see that our method generate promising task-oriented flows, *generally consistent with the ground truth optical flows but diverse in local details.*

5. Conclusion

Following the property of task-oriented flow, we have introduced All-pairs Multi-field Transforms (AMT) for efficient frame interpolation. It contains two essential designs, including all-pairs correlation and multi-field refinement. Through the two designs, our method could effectively handle large motions and occluded regions during frame interpolation and achieve state-of-the-art performance on various benchmarks with high efficiency.

Acknowledgment: This work is funded by the NSFC (NO. 62176130), Fundamental Research Funds for the Central Universities (Nankai University, NO. 63223050), China Postdoctoral Science Foundation (NO.2021M701780). This work is partially supported by Ascend AI Computing Platform and CANN (Compute Architecture for Neural Networks). We are also sponsored by CAAI-Huawei MindSpore Open Fund.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, 2019. 2, 6

[2] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE TPAMI*, 2018. 1, 2

[3] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 2022. 2

[4] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 5

[5] Xianhang Cheng and Zhenzhong Chen. Video frame interpolation via deformable separable convolution. In *AAAI*, 2020. 2

[6] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE TPAMI*, 2021. 2

[7] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Yuanhao Yu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. Error-aware spatial ensembles for video frame interpolation. *arXiv preprint arXiv:2207.12305*, 2022. 2

[8] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, 2020. 2, 5, 6, 7

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *CVPR*, 2017. 2

[10] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. Cdfi: Compression-driven network design for frame interpolation. In *CVPR*, 2021. 2

[11] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016. 1

[12] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE TPAMI*, 2012. 3

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2

[14] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *CVPR*, 2022. 1, 6

[15] Xiaotao Hu, Zhewei Huang, Ailin Huang, Jun Xu, and Shuchang Zhou. A dynamic multi-scale voxel flow network for video prediction. In *CVPR*, 2023. 2

[16] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *ECCV*, 2022. 3

[17] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 2022. 1, 2, 4, 6, 7

[18] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. In *CVPR*, 2018. 5

[19] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019. 3

[20] Sangryul Jeon, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn. Parn: Pyramidal affine regression networks for dense semantic correspondence. In *ECCV*, 2018. 3

[21] Zhaoyang Jia, Yan Lu, and Houqiang Li. Neighbor correspondence matching for flow-based video frame synthesis. In *ACM MM*, 2022. 3

[22] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018. 1, 2, 4

[23] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512*, 2020. 2

[24] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 3

[25] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In *AAAI*, 2020. 2

[26] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *CVPR*, 2022. 1, 2, 4, 5, 6, 7

[27] Hyeongmin Lee, Taeoh Kim, Tae young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, 2020. 2, 6, 7

[28] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *CVPR*, 2022. 2

[29] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 1

[30] Jing Lin, Yuanhao Cai, Xiaowan Hu, Haoqian Wang, Youliang Yan, Xueyi Zou, Henghui Ding, Yulun Zhang, Radu Timofte, and Luc Van Gool. Flow-guided sparse transformer for video deblurring. In *ICML*, 2022. 2

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[32] Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 1, 2, 4, 5

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 5

[34] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *CVPR*, 2022. 1, 2, 6

[35] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flowwith a bidirectional census loss. In *AAAI*, 2018. 5

[36] Christopher Montgomery. Xiph.org video test media (derf's collection). In *Online,https://media.xiph.org/video/derf/*, 1994. 5, 6

[37] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, 2018. 1, 2, 4

[38] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, 2020. 1, 2, 4, 6

[39] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017. 2

[40] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 2

[41] Simon Niklaus, Long Mai, and Oliver Wang. Revisiting adaptive convolutions for video frame interpolation. In *WACV*, 2021. 2

[42] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *ECCV*, 2020. 2, 3, 4, 6

[43] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *CVPR*, 2021. 1, 3, 4, 6, 7

[44] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *CVPR*, 2019. 2

[45] Feng Liu Qiqi Hou, Abhijay Ghildyal. A perceptual quality metric for video frame interpolation. In *ECCV*, 2022. 1

[46] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *ECCV*, 2022. 2

[47] Zhile Ren, Orazio Gallo, Deqing Sun, Ming-Hsuan Yang, Erik B Sudderth, and Jan Kautz. A fusion approach for multi-frame optical flow estimation. In *WACV*, 2019. 7

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[49] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 2

[50] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *CVPR*, 2022. 2

[51] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *ICCV*, 2021. 2

[52] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1

[53] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *CVPR*, 2021. 1

[54] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5, 6

[55] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3, 7

[56] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1, 2, 3, 4, 7, 8

[57] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 5

[60] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *ECCV*, 2018. 1

[61] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P Allebach, and Chenliang Xu. Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In *CVPR*, 2020. 1

[62] Jin Xin, Wu Longhai, Shen Guotao, Chen Youxin, Chen Jie, Koo Jayoon, and Hahm Cheul-hee. Enhanced bi-directional motion estimation for video frame interpolation. *arXiv preprint arXiv:2206.08572*, 2022. 3

[63] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022. 1

[64] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, 2019. 2, 4

[65] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019. 1, 2, 5, 6, 7

[66] Songhyun Yu, Bumjun Park, Junwoo Park, and Jechang Jeong. Joint learning of blind video denoising and optical flow estimation. In *CVPRW*, 2020. 2

[67] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Philip H.S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 3

[68] Guozhen Zhang, Yuhan Zhu, Hongya Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, 2023. 2, 6

[69] Haoxian Zhang, Yang Zhao, and Ronggang Wang. A flexible recurrent residual pyramid network for video frame interpolation. In *ECCV*, 2020. 1

[70] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022. 1

[71] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 1