Perception-Oriented Video Frame Interpolation via Asymmetric Blending

Guangyang Wu[1] Xin Tao[2] Changlin Li[3] Wenyi Wang[4] Xiaohong Liu[1†] Qingqing Zheng[5†]
[1]Shanghai Jiao Tong University [2]Kuaishou Technology [3]SeeKoo
[4]University of Electronic Science and Technology of China
[5]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences

Ground-Truth RIFE [17] EMA-VFI [59] STMFNet [8] PerVFI (ours)

Figure 1. We present challenging video frame interpolation examples, demonstrating our approach excels in handling large motion, outperforming alternatives prone to blurriness or ghosting.

Abstract

Previous methods for Video Frame Interpolation (VFI) have encountered challenges, notably the manifestation of blur and ghosting effects. These issues can be traced back to two pivotal factors: unavoidable motion errors and misalignment in supervision. In practice, motion estimates often prove to be error-prone, resulting in misaligned features. Furthermore, the reconstruction loss tends to bring blurry results, particularly in misaligned regions. To mitigate these challenges, we propose a new paradigm called PerVFI (Perception-oriented Video Frame Interpolation). Our approach incorporates an Asymmetric Synergistic Blending module (ASB) that utilizes features from both sides to synergistically blend intermediate features. One reference frame emphasizes primary content, while the other contributes complementary information. To impose a stringent constraint on the blending process, we introduce a self-learned sparse quasi-binary mask which effectively mitigates ghosting and blur artifacts in the output. Additionally, we employ a normalizing flow-based generator and utilize the negative log-likelihood loss to learn the conditional distribution of the output, which further facilitates the generation of clear and fine details. Experimental results validate the superiority of PerVFI, demonstrating significant improvements in perceptual quality compared to existing methods. Codes are available at https://github.com/mulns/PerVFI

1. Introduction

Video frame interpolation (VFI) is an important task in computer vision that focuses on synthesizing intermediate frames between consecutive frames in a video sequence. This technique plays a crucial role in various applications such as video enhancement [27, 56], slow-motion rendering [18, 52, 53], and frame rate conversion [33], especially for producing high-definition videos [17, 44].

Recently, the VFI task has also derived benefits from the application of deep neural networks [17–19, 24, 26, 29–32, 34, 36, 37, 39, 42–44, 48, 51, 55, 56, 58]. We broadly posit that recent methods typically consist of 3 main modules. The motion estimation module is used to estimate motion between consecutive frames using optical flow or deformable kernels. Subsequently, the alignment and fusion module aligns reference frames via warping operator or deformable convolution. Finally, the reconstruction module generates final results from extracted features. Despite the success of recent methods, blurry results and ghosting artifacts persist as inevitable problems. (shown in Figure 1). We attribute this primarily to two inherent challenges:

Inevitable Motion Errors. Ideally, with accurate motion estimates, the aforementioned procedure can yield satisfactory results. However, achieving error-free pixel-wise correspondence for real-world videos proves challenging, es-

† Corresponding authors.

pecially in the presence of large-scale motions. Unlike several prior methods [8, 19, 40, 44] that primarily focus on enhancing the quality of motion estimation, our objective is to fortify our network against alignment errors. Specifically, after a thorough investigation of existing methods, we conclude that in cases of inaccurate motion estimation, the network struggles to discern the correct frame. Consequently, preceding algorithms often produce blurred and ghosted results by averaging multiple frames. We contend that basing the output on a single frame while utilizing other frames to supplement specific details holds the potential to yield clearer and more plausible results.

Temporal Supervision Misalignment. Another crucial yet often overlooked issue in VFI is the temporal uncertainty. During the training phase, the ground truth (GT) intermediate frame only provides a reference at a specific time. However, in the case of a continuous natural video, scenes captured in the time interval between two frames can offer multiple potential solutions. Therefore, the learned intermediate features can vary across different training videos. We term this issue Temporal Supervision Misalignment, and this misalignment may cause the network to produce blurry results. To address this problem, conventional pixel-wise loss functions such as L1 and L2 are inadequate. Instead, we opt for generative models to reconstruct results sampled from a distribution.

In order to tackle the aforementioned challenges, we propose a novel perception-oriented video frame interpolation paradigm in this paper, referred to as PerVFI. Our approach introduces an Asymmetric Synergistic Blending (ASB) module and a self-learned sparse quasi-binary mask to fuse multi-frame features. In this process, one reference frame emphasizes primary content, while the other provides complementary information. Additionally, we employ a normalizing flow-based generator to decode the intermediate features. This generator models the conditional distribution of the output based on the reference inputs. Unlike GAN-based methods that struggle to converge [8] and diffusion-based methods [11] with numerous iterations, our normalizing flow-based approach demonstrates stability during training and low latency during inference. The proposed PerVFI paradigm consistently produces visually high-quality results, even in the presence of misalignment due to inaccurate motion estimates.

The contributions of this paper are as follows:

- We introduce a novel paradigm called PerVFI, specifically designed for the perception-oriented task of VFI. Our proposed method tackles the issue of misalignment by incorporating an asymmetric synergistic blending module (ASB) and a conditional normalizing flow-based generator.
- To effectively control the blending process in ASB, we propose a novel quasi-binary mask. This mask allows for sparse confidence values overall and adaptive values for partial areas, effectively addressing the occlusion and imposing a strict constraint on blending process.
- We have conducted extensive experiments to validate the efficacy of the proposed PerVFI. The experimental results demonstrate that the PerVFI paradigm is capable of generating visually plausible outputs even in the presence of inaccurate motion estimates, exhibiting state-of-the-art performance in terms of perceptual quality.

2. Related Works

2.1. Video frame interpolation.

Existing VFI approaches are mostly based on deep learning , and can be generally categorized as optical flow-based or kernel-based. Optical flow-based methods rely on optical flow estimation to generate interpolated frames[17–19, 24, 32, 36, 37, 39, 40, 44, 55, 56]. On the other hand, kernel-based methods argue that optical flows can be unreliable in dynamic texture scenes [4, 6, 13, 15, 25, 34, 38], so they predict locally adaptive convolution kernels to synthesize output pixels. Other than these two classes, there are also attempts to combine flows and kernels [2, 3, 8, 26] and to perform end-to-end frame synthesis [7, 21]. It is noted that the above methods use symmetric blending, which easily blends the features from two sides with equal contribution, even when they are misaligned. Although these results in reasonably good PSNR performance, it has been previously reported [9] that PSNR does not fully reflect the perceptual quality of interpolated videos, exhibiting poor correlation performance with subjective ground truth. To improve perceptual performance, some existing methods [36, 37] use the perceptual loss [45] in combination with the L1 loss. An alternative approach uses GANs [8, 25] to enhance perceptual quality of interpolated videos. However, due to the instability of GAN training, these models are pretrained using L1 loss before fine-tuned with adversarial loss, leading to limited improvement in perceptual quality.

2.2. Normalizing Flow-based model.

Recently, normalizing flow-based models have shown remarkable performance in synthesizing high-fidelity images and videos [14, 23, 35, 49]. These models leverage an invertible network to establish a mapping from a complex distribution to a simple distribution. Users can then decode a latent code sampled from the simple distribution to the target domain. Normalizing flow-based methods have been reported to outperform GANs in image generation tasks [23, 35, 49]. To the best of our knowledge, we are the first to adopt a normalizing flow-based model for VFI. In particular, we utilize conditional normalizing flow-based models [35], which have demonstrated a strong ability to

synthesize images with conditional information.

3. Proposed Method: PerVFI

Given two referenced frame images I_0 and $I_1 \in \mathbb{R}^{H\times W\times 3}$ with height H and width W, our goal is to reconstruct the intermediate frame I_t regarding the target time $t \in (0, 1)$. The overall framework of PerVFI is presented in Figure 2-(a), which includes an asymmetric synergistic blending (ASB) module illustrated in Figure 2-(b) and a conditional normalizing flow-based generator illustrated in Figure 2-(c).

Our first step is to estimate bidirectional optical flows, denoted by $\mathbf{F}_{0\to1}$ and $\mathbf{F}_{1\to0}$, using a motion estimator such as RAFT [47] or GMFlow [54]. Concurrently, we use a pyramidal architecture that extracts features at different scales to capture multiscale information. Specifically, we encode the two images into pyramid features with L levels using a feature encoder $\mathcal{E}_\theta$, denoted as $f_i = \mathcal{E}_\theta(I_i)$ for $i = 0, 1$. Once the bidirectional optical flow and feature pyramid have been obtained. We utilize a feature blending module, denoted as $\mathcal{B}_\theta$ and obtain intermediate pyramid features by blending, denoted as $f_t = \mathcal{B}_\theta(t, f_0, f_1, \mathbf{F}_{0\to1}, \mathbf{F}_{1\to0})$. Then, we decode f_t into the output frame I_t using a conditional normalizing flow-based generator $\mathcal{G}_\theta$ which is invertible, denoted as $I_t = \mathcal{G}_\theta^{-1}(z; f_t)$ where $z \sim \mathcal{N}(0, \tau) \in \mathbb{R}^{H\times W\times 3}$ is a variable sampled from a normal distribution with temperature τ. The feature pyramid, $f_t = \{f_t^l \mid l \in [0, 1, \ldots, L-1]\}$ represents L features with shapes of $\frac{H}{2^l} \times \frac{W}{2^l}$.

3.1. Asymmetric Synergistic Blending

To effectively learn f_t immune to bidirectional motion misalignment, an intuitive insight involves extracting primary information from one reference frame and compensating for occlusion information from the other frame, instead of simply averaging the information from both frames without any constraints. This can be achieved by obtaining a binary occlusion mask that describes whether certain regions are occluded or not, and blending the aligned features from both sides using this mask. However, due to the inaccurate motion estimates, obtaining an accurate binary occlusion mask is challenging, and aligning the features from both sides is non-trivial. Therefore, we propose a novel Asymmetric Synergistic Blending (ASB) module, which focuses on addressing these two problems. The ASB module consists of two major components: the Pyramid Alignment Module (PAM) and the Adaptive Dilation Module (ADM). The PAM is designed to achieve more accurate alignment of features from both sides, while the ADM aims to provide a quasi-binary mask that can serve as a weighting mask for better handling of occlusion.

Pyramid Alignment Module. Alignment is a crucial step in our framework, involving the warping of reference frames and aligning pyramid features. We employ different warping operators: backward warping ($\overleftarrow{\omega}$) and forward warping with different splatting methods. For multiple-to-one situations [37], we use the average splatting operator ($\overrightarrow{\omega_{avg}}$) and the softmax splatting operator ($\overrightarrow{\omega_Z}$) which subjects to an importance metric Z.

Following Niklaus and Liu [37], the f_0 is warped to time t using a Forward Warping module with a small neural network v_θ. The importance metric Z is computed as:

$$Z = v_\theta(f_0^0, -\left\|f_0^0 - \overleftarrow{\omega}(f_1^0, \mathbf{F}_{0\to1})\right\|), \tag{1}$$

and the warped pyramid features $f_{t,0}$ are obtained through:

$$f_{t,0}^l = \overrightarrow{\omega_Z}(f_0^l, t, \mathbf{F}_{0\to1}^l; Z^l), \quad l = 0, 1, \ldots, L-1 \tag{2}$$

where $\mathbf{F}_{0\to1}^l$ and Z^l are spatially downscaled optical flow and metric, respectively, by a factor of 2^l.

To handle occlusion regions in the warped features $f_{t,0}$, we introduce a Pyramid Alignment Module (PAM) to align f_1 to the target time t, producing $f_{t,1}$. The PAM utilizes a multiscale deformable convolution network u to perform alignment. Instead of warping f_1 directly using $\mathbf{F}_{1\to t}$, which may cause misalignment because of inaccurate bidirectional optical flows, the PAM converts $\mathbf{F}_{1\to t}$ to an initialized offset to guide the alignment process. This approach improves the handling of occlusion regions, resulting in more accurate alignment. The alignment process is formulated as:

$$\mathbf{F}_{1\to t} = (1-t)\cdot\mathbf{F}_{1\to0}, \tag{3}$$

$$f_{t,1} = u_\theta(-1\times\overrightarrow{\omega_{avg}}(\mathbf{F}_{1\to t}, \mathbf{F}_{1\to t}), f_1, f_{t,0}). \tag{4}$$

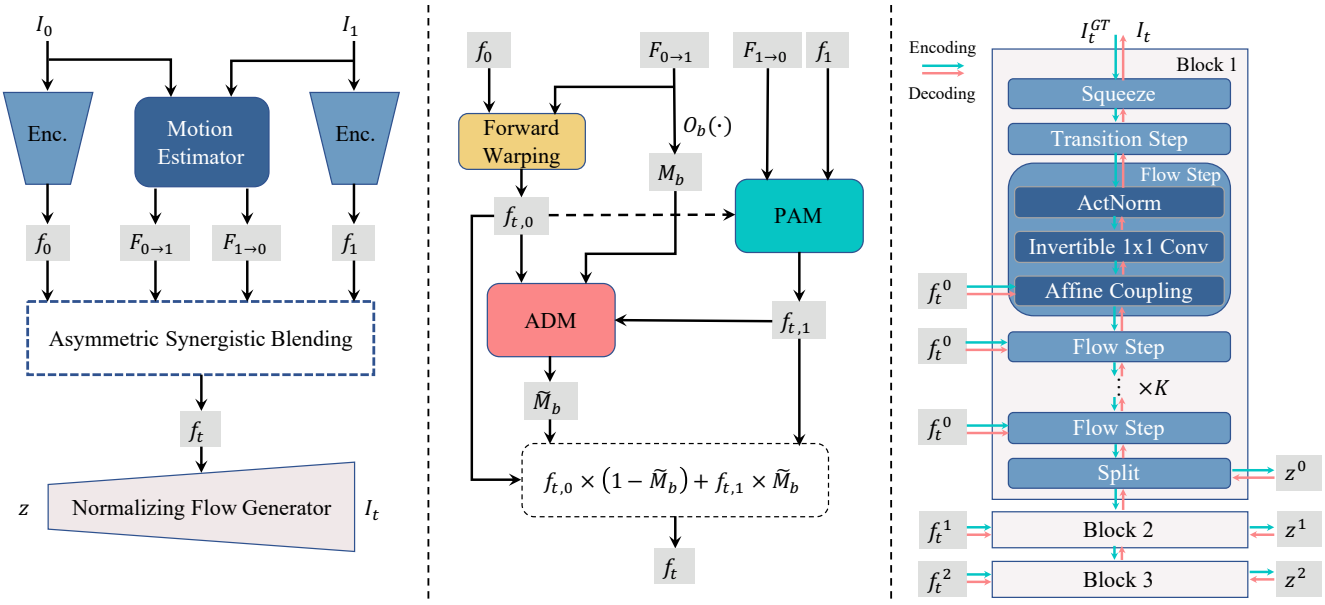
The network u_θ iteratively refines the transformation parameters in a coarse-to-fine manner, starting with aligning the features at the coarsest scale and propagating the alignment to finer scales. This helps to handle large pixel displacements and achieve more accurate alignment of features at different scales. A detailed network structure of u_θ and v_θ can be found in the appendix.

Adaptive Dilation Module. To mitigate errors caused by binary masks generated from motion estimates and improve occlusion compensation, we introduce a lightweight Adaptive Dilation module (ADM). This module generates a quasi-binary weighting mask to control the blending of features from two sides. Specifically, ADM adaptively dilates the binary mask with convolution layers and maintains the sparsity property. In ADM, we independently generate one weighting mask for each pyramid level. Before dilation, we obtain the binary occlusion mask $M_b^l \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}}$ at level l by applying a threshold method, denoted as $\mathcal{O}_b$, to $\mathbf{F}_{0\to1}^l$:

$$M_b^l = \mathcal{O}_b(\mathbf{F}_{0\to1}^l). \tag{5}$$

For each pixel $\mathbf{x} \in (1, \ldots, \frac{H}{2^l}) \times (1, \ldots, \frac{W}{2^l})$, we compute:

$$M_b^l(\mathbf{x}) = \begin{cases} 1, & \text{if } \overrightarrow{\omega_{avg}}(M_1^l, t\cdot\mathbf{F}_{0\to1}^l)(\mathbf{x}) < \epsilon; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$



(a) Overall Framework (b) Asymmetric Synergistic Blending (c) Normalizing Flow Generator

Figure 2. (a): Overview of the entire PerVFI framework. (b): Structure of the proposed Asymmetric Synergistic Blending (ASB) module. (c): Structure of the conditional normalizing flow-based generator.

Here, M_1^l is a constant metric with the same shape as M_b^l filled with value 1, and the threshold ϵ is a constant set to 0.5 by default.

ADM generates the quasi-binary mask $\widetilde{M}_b^l$ based on M_b^l and the two aligned feature sets $f_{t,0}$ and $f_{t,1}$, as illustrated in Figure 3. Firstly, we expand $\mathcal{M}_b$ to C feature maps using convolution layers denoted as $\mathcal{F}_{ex}$, consisting of three bias-free convolutional layers with kernel sizes $(7, 3, 1)$, respectively. The combination of these layers results in a 17×17 dilation field for each occluded pixel in M_b^l. We then obtain a sample-dynamic attention weight $\mathbf{a} \in \mathbb{R}^C$ derived from reference features using squeeze and excitation layers, denoted as $\mathcal{F}_{att}$. The expanded features are scaled with normalized attention weights, denoted as $\mathcal{F}_{scale}$. Afterwards, the scaled features are projected to metric $\widehat{M}^l$ through one bias-free convolution layer with kernel size 1, denoted as $\mathcal{F}_{proj}$. The procedure can be formulated as:

$$\widehat{M}^l = \mathcal{F}_{proj}(\mathcal{F}_{scale}(\mathcal{F}_{exp}(M_b^l), \mathcal{F}_{att}(f_{t,0}^l, f_{t,1}^l))). \quad (7)$$

Finally, the quasi-binary mask is obtained using the following equation:

$$\widetilde{M}_b^l = \tanh(\text{abs}(\widehat{M}^l + \alpha \cdot n) + \beta \cdot M_b^l). \quad (8)$$

Here, β controls the salience of occluded regions and set to 2 by default. $n \sim \mathcal{U}(-1, 1)$ is a random noise with the same shape as M_b^l, and α is set to $1e^{-3}$ during training and 0 during inference. We found it necessary to add the random noise during training to avoid gradient vanishing and lead to more robust occlusion compensation. By blending the two sides information through quasi-binary mask, we obtain the output pyramid feature f_t as follows:

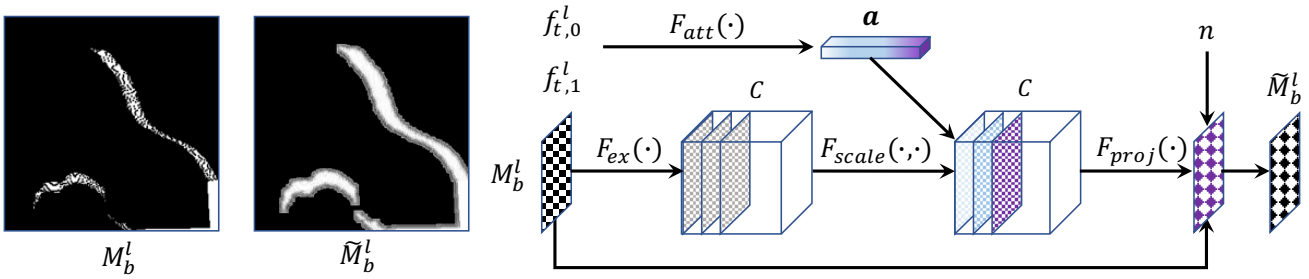
$$f_t^l = f_{t,0}^l \cdot (1 - \widetilde{M}_b^l) + f_{t,1}^l \cdot \widetilde{M}_b^l. \quad (9)$$

The adaptive dilation field of the quasi-binary mask maintains its sparsity and reduces the impact of errors in the binary mask, therefore providing more plausible primary content. Moreover, the sample-dynamic attention mechanism in the ADM module adapts to different pyramid levels and reference features, providing more robust occlusion compensation.

3.2. Normalizing Flow Generator

To parameterize the conditional distribution $p(I_t|f_t)$, we utilize an invertible neural network $\mathcal{G}_\theta$, which maps the intermediate feature f_t and the target image I_t to a latent variable $z = \mathcal{G}_\theta(I_t; f_t)$. By employing an invertible normalizing flow-based generator $\mathcal{G}_\theta$, we ensure that I_t can be accurately reconstructed from the latent encoding z as $y = \mathcal{G}_\theta^{-1}(z; f_t)$. By assuming a simple distribution $p_z(z)$ (*e.g.*, Gaussian) in the latent space z, the distribution $p(I_t|f_t, \theta)$ is implicitly defined through the mapping $y = \mathcal{G}_\theta^{-1}(z; f_t)$ of $z \sim p_z$. In a normalizing flow-based generator, the probability density p can be explicitly computed as:

$$p(I_t|f_t, \theta) = p_z(\mathcal{G}_\theta(I_t; f_t)) \left| \det \frac{\partial \mathcal{G}_\theta}{\partial I_t}(I_t; f_t) \right|, \quad (10)$$



(a) Visualization (b) Illustration of ADM

Figure 3. The Adaptive Dilation Module (ADM) produces the quasi-binary mask $\widetilde{M}_b^l$ by leveraging the binary occlusion mask M_b^l and the two aligned feature sets $f_{t,0}$ and $f_{t,1}$. Panel (a) provides a visualization of the input and output masks, while panel (b) presents the flowchart outlining the operations of ADM. Further information regarding the intricacies of the module is detailed in Equations 5 - 8.

and we can train the parameters θ by minimizing the negative log-likelihood (NLL). Let $\hat{p} = -\log p_z(\mathcal{G}_\theta(I_t; f_t))$, the NLL is formulated as:

$$\begin{aligned}\mathcal{L}_{nll}(\theta; f_t, y_t) &= -\log p(I_t|f_t, \theta) \\ &= \hat{p} - \log \left|\det \frac{\partial \mathcal{G}_\theta}{\partial I_t}(I_t; f_t)\right|. \end{aligned} \tag{11}$$

As shown in Figure 2-(c), we decompose $\mathcal{G}_\theta$ into a sequence of N invertible layers $h^{n+1} = \mathcal{G}_\theta^n(h^n; f_t)$, where we have $h^0 = I_t$ and $h^N = z$. The invertible layers consist of Squeeze Layer, Transition Layer, Invertible 1x1Conv Layer, Actnorm Layer, Affine Coupling Layer, and Split Layer. By applying the chain rule along with the multiplicative property of the determinant, the NLL can be expressed as:

$$\mathcal{L}_{nll}(\theta; f_t, I_t) = \hat{p} - \sum_{n=0}^{N-1} \log \left|\det \frac{\partial \mathcal{G}_\theta^n}{\partial h^n}(h^n; f_t)\right|. \tag{12}$$

We thus only need to compute the log-determinant of the Jacobian $\frac{\partial \mathcal{G}_\theta^n}{\partial h^n}$ for each individual flow-layer $\mathcal{G}_\theta^n$. Other than NLL loss, we also adopt perceptual loss as auxiliary loss implemented with VGG-network [20]. We find that the introduction of auxiliary loss can significantly improve the convergence speed of the network training, and the final generated images are less noisy and clearer. The auxiliary loss is calculated as follows:

$$z' \sim \mathcal{N}(mean(\mathcal{G}_\theta(I_t^{GT}; f_t)), var(\mathcal{G}_\theta(I_t^{GT}; f_t))), \tag{13}$$

$$\mathcal{L}_{per}(\theta; f_t, z') = ||\mathcal{F}_{vgg}(\mathcal{G}_\theta^{-1}(z'; f_t)) - \mathcal{F}_{vgg}(I_t^{GT})||_2. \tag{14}$$

Here, we first encode the I_t to latent space z, then randomly sample z' according to the mean and variation of z for decoding. The final bidirectional loss is formulated as:

$$\mathcal{L}(\theta; f_t, I_t) = \mathcal{L}_{nll} + \mu \cdot \mathcal{L}_{per}. \tag{15}$$

Affine Coupling. The affine coupling layer integrate the condition information and easily invertible. Unlike the conditional affine coupling layer introduced in [35], where the inverse operation is afflicted by numerical instability, we modify it into a more stable version as follows:

$$\begin{aligned} h_A^{n+1} &= h_A^n, \\ h_B^{n+1} &= \exp(\lambda^n \cdot \tanh(w_s^n(h_A^n; f_t)) + \eta^n) \cdot h_B^n \\ &\quad + w_b^n(h_A^n; f_t). \end{aligned} \tag{16}$$

Here, $h^n = (h_A^n, h_B^n)$ is a partition of the feature map in the channel dimension. w_s^n and w_b^n are neural networks generating the scaling and bias of h_B^n. λ^n and η^n are learnable scalars. For stability, we initialize parameters of λ^n and η^n to 1, and the last convolutional layer of w_s^n and w_b^n to 0. Since the Jacobian of (16) is triangular, its log-determinant is easily calculated as

$$\lambda^n \cdot \sum_{ijk}(\tanh(w_s^n(h_A^n; f_t))_{ijk} + \sum_{ijk} \eta^n. \tag{17}$$

Other Details. The Invertible 1×1 Conv layer, ActNorm Layer, Squeeze Layer and Transition Layer are following the basic settings in [35]. Specifically, we stack $L = 3$ blocks regarding three pyramid levels, each containing $K = 16$ flow-steps. During encoding, in each block, the Split Layer outputs a latent variable z_l, and the final latent variable $z = (z_l)_{l=1}^L$ models variations in the image at different resolutions. During decoding, all the components in z are independently sampled. More details can be found in the appendix.

4. Experiments

4.1. Experimental Settings

Training Settings. The PerVFI network is trained using the bidirectional loss in Equation (15), with μ set to 0.2 as the default value. It is worth noting that during training, we enhance the robustness of our network by randomly selecting motion estimates generated by RAFT [47] and GM-Flow [54]. For motion estimation modules, we utilize the

official pretrained models on the Sintel dataset and freeze their parameters. Our training dataset consists of frame-triples from the training portion of the publicly available Vimeo-90k [56] dataset. Throughout the training process, we employ a patch size of 256×256, with a batch size of 16 for each iteration. The ADAM optimizer with default hyperparameter settings in [22] is used, and the initial learning rate is set to $5e^{-4}$. The learning rate is halved every 20 epochs. The PerVFI model is trained until convergence, which typically occurs around 64 epochs, using two NVIDIA RTX 3090 GPUs.

State-of-the-art Methods. We compare our approach to several state-of-the-art video frame interpolation methods, including EDSC [5], RIFE [17], VFIFormer [34], EMA-VFI [59], AMT [28] and STMFNet [8], using their publicly available implementations. Additionally, we include LDMVFI [11] and refer to the data reported in paper [11]. Our PerVFI uses RAFT [47] as motion estimator, and we sample the latent code z using temperature $\tau = 0.3$.

Datasets. To evaluate the performance of the models, we employ commonly used VFI benchmarks: Vimeo-90K [56], DAVIS (2017) [41] and Xiph [37] datasets. We opted for video datasets to evaluate perceptual metrics like VFIPS [16] and FloLPIPS [10] (two bespoke VFI metrics). Evaluations for the DAVIS dataset are conducted at both 480P (640×480) and 1080P (1920×1080) resolutions. For the Xiph dataset, 8 video sequences, each containing 101 4K frames, are used. Following the approach in [37], we resize the 4K frames to 2K (2048×1080) or extract a 2K center crop. All video sequences interpolate even frames based on the corresponding odd frames. Results of comparisons on Middlebury [1] and UCF101 [46] datasets are included in the appendix.

Metrics. We mainly employ the following metrics for performance evaluation: LPIPS [60] and DISTS [12] (image quality metric); VFIPS [16] and FloLPIPS [10] (video quality metric). These metrics have demonstrated a stronger correlation with human judgments of frame interpolation quality. For completeness, we also present the performance of traditional metrics PSNR and SSIM [50]. However, it is important to note that they are not the primary focus of this paper. Higher values indicate better performance for PSNR, SSIM, and VFIPS, while lower values indicate better results for LPIPS and FloLPIPS. Results of comparisons using DISTS [12] are included in the appendix.

4.2. Quantitative Evaluation

We present a comparative analysis of PerVFI against state-of-the-art methods in Table 1 and 2. While STMFNet [8] achieves the highest PSNR and SSIM values, its performance falls short in perceptual quality according to the other three metrics. Notably, the utilization of symmetric blending in all methods, except for PerVFI, results in noticeable ghosting and blur artifacts. This is observed even in methods with advanced synthesis modules such as the GAN-based STMFNet [8] and the diffusion-based LDMVFI [11]. By leveraging the ASB module, our PerVFI surpasses other methods significantly in terms of perceptual quality. Moreover, when comparing high resolution videos, our PerVFI, trained exclusively on the Vimeo90K dataset, maintains superior visual quality even when compared to STMFNet [8] and LDMVFI [11], which are trained on high-resolution datasets. This further showcases the generalization capability of our proposed method.

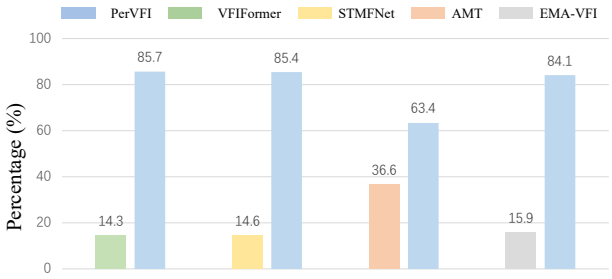


Figure 4. User study results.

4.3. Qualitative Evaluation

We also compare the visual quality of different methods in DAVIS dataset, as illustrated in Figure 5. In regions with significant pixel displacement, it is evident that other methods produce outputs with noticeable ghosting artifacts or blurriness, resulting in a significant degradation of visual quality. In contrast, PerVFI consistently generates outputs with sharp edges and intact content, leading to visually pleasing results. It is important to note that while the PerVFI results may not exhibit perfect pixel-wise alignment with the ground-truth image, the overall visual quality remains consistent with the reference images.

Additionally, to facilitate a more comprehensive examination of the visual quality, we conduct a user study involved 33 participants comparing 98 videos interpolated with 5 methods. We conduct A/B test on perceptual quality, where the ratio values indicate the percentages of participants preferring the corresponding model. Statistical results are presented in Figure 4, demonstrating our method consistently outperforms others.

4.4. Ablation Experiments

Symmetric vs. Asymmetric blending. To emphasize the importance of asymmetric synergistic blending, we have designed a symmetric blending module for a fair comparison. In this module, we introduce a learnable bias after each convolution layer in the Adaptive Blending Module (ADM), allowing the output mask to be fully adaptive without any constraints. As depicted in the first row of Figure 6-(a), the fully adaptive mask tends to merge features from both

Table 1. Performance comparison of VFI algorithms on DAVIS-2017 [41]. The scores for LDMVFI [11] are taken from their paper and indicated with the † symbol. 'OOM' means out of memory. The best values are highlighted in **red** and the second-best values are in blue.

	DAVIS (480P)					DAVIS (1080P)				
	PSNR↑	SSIM ↑	LPIPS↓	FloLPIPS↓	VFIPS↑	PSNR↑	SSIM↑	LPIPS↓	FloLPIPS↓	VFIPS↑
EDSC [5]	26.52	0.784	0.132	0.093	72.62	24.54	0.768	0.205	0.138	51.05
RIFE [17]	26.97	0.807	0.085	0.063	80.19	25.89	0.803	0.134	0.097	62.56
STMFNet [8]	28.55	0.850	0.121	0.086	77.38	27.43	0.844	0.178	0.119	60.25
LDMVFI [11]	25.54 †	-	0.107 †	0.153 †	75.78 †	-	-	-	-	-
VFIFormer [34]	27.33	0.814	0.124	0.090	77.32	OOM	OOM	OOM	OOM	OOM
EMA-VFI [59]	**28.83**	**0.856**	0.127	0.085	78.84	**27.61**	**0.846**	0.203	0.131	60.87
AMT [28]	27.42	0.818	0.101	0.073	80.57	25.72	0.806	0.177	0.122	60.39
PerVFI (ours)	26.83	0.804	**0.077**	**0.058**	**87.51**	26.23	0.808	**0.114**	**0.087**	**72.52**

Table 2. Performance comparison of VFI algorithms on Xiph4K [37] and Vimeo-90K [57]. The best values are highlighted in **red**, while the second-best values are in blue. 'OOM' means out of memory.

	Xiph - 2K			Xiph - "4K"			Vimeo-90K		
	LPIPS↓	FloLPIPS↓	VFIPS↑	LPIPS↓	FloLPIPS↓	VFIPS↑	PSNR↑	SSIM↑	LPIPS↓
EDSC [5]	0.085	0.072	64.73	0.177	0.120	51.24	34.86	0.961	0.027
RIFE [17]	0.041	0.050	65.26	0.099	0.067	54.31	34.16	0.955	0.020
STMFNet [8]	0.110	0.063	65.19	0.245	0.128	53.33	-	-	-
VFIFormer [34]	OOM	OOM	OOM	OOM	OOM	OOM	**36.38**	**0.971**	0.021
EMA-VFI [59]	0.110	0.081	65.12	0.241	0.114	53.57	36.34	0.967	0.026
AMT [28]	0.089	0.055	65.60	0.199	0.114	53.22	35.79	0.968	0.021
PerVFI (ours)	**0.038**	**0.032**	**68.67**	**0.086**	**0.062**	**57.47**	33.89	0.953	**0.018**

Table 3. Ablation Experiment Results. We show the PSNR and VFIPS on DAVIS (480P) dataset.

Mask	Noise	Prior	Loss	PSNR	VFIPS
Binary	✗	✓	Bi-D.	26.52	81.01
Quasi-B.	✗	✓	Bi-D.	26.71	81.24
Quasi-B.	✓	✗	Bi-D.	27.10	82.27
Quasi-B.	✓	✗	L1	26.81	80.34
Quasi-B.	✓	✓	L1	27.15	80.50
Quasi-B.	✓	✓	NLL	26.98	81.88
Binary	✓	✓	Bi-D.	26.69	81.25
Adaptive	✓	✓	Bi-D.	**27.61**	78.20
Quasi-B.	✓	✓	Bi-D.	27.16	**83.30**

sides equally, resulting in a blurry output, as shown in the second row. Furthermore, in Figure 6-(b), we provide a histogram illustrating the distribution of values for different types of masks. The quasi-binary mask (denoted as Quasi-B.) maintains overall sparsity while being partially adaptive. In Table 3, we use "Adaptive" to indicate the fully adaptive mask without any constraints, and "Quasi-B." to represent the quasi-binary mask. As observed, symmetric blending with a fully adaptive weighting mask achieves higher PSNR values but lower VFIPS scores compared to the asymmetric blending with the quasi-binary mask. This phenomenon demonstrates the significance of imposing strict constraints during the blending process to enhance the perceptual quality of the output.

Quasi-binary mask. In the ADM, we introduce a random uniform noise term n during training. This is done because multiplying with a sparse mask could potentially lead to gradient vanishing issues. To showcase the importance of this operation, we conduct experiments where we train the model without the adding the noise. Specifically, we evaluate the model using either the quasi-binary mask or a binary mask without dilation. As presented in the first and second rows of Table 3, the performance of the model experiences a noticeable degradation when trained without the inclusion of random noise. This emphasizes the necessity of incorporating the random noise term during training. We will provide further visual comparisons in the appendix to complement these findings.

PAM module. In the PAM, we utilize the optical flow $\mathbf{F}_{1\to 0}$ as prior information for alignment. We found it necessary to incorporate this prior in order to effectively handle occlusion compensation. As demonstrated in Table 3, removing prior information from the PAM leads to failures in occlusion compensation in the resulting frames. Additional visual comparisons are shown in the appendix, highlight-

Figure 5. Perceptual quality comparison between different methods. Our approach produces a high-quality result in spite of the fast-moving objects that is subject to large motion. Red arrows emphasize areas where PerVFI excels in visual quality compared to other methods.

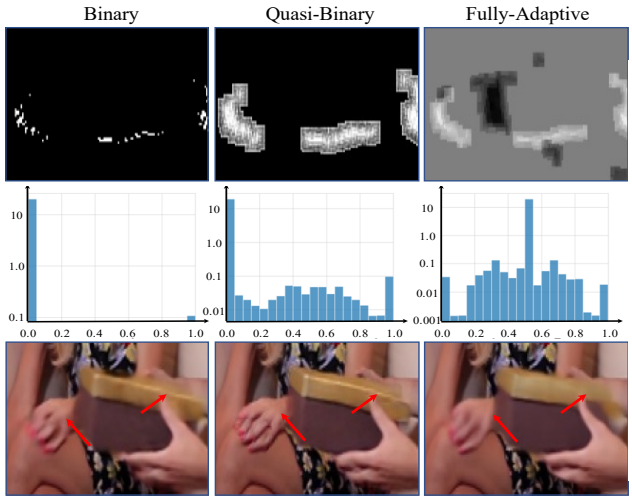


Figure 6. The first row visualize different masks, and the second row exhibits histograms for each. The fully-adaptive mask tends to center around 0.5, signifying an equal contribution from both sides features. The quasi-binary mask maintains sparsity while being partially adaptive, providing an effective blending mechanism. The third row presents results using these masks. Red arrows emphasize areas where the quasi-binary mask excels in visual quality

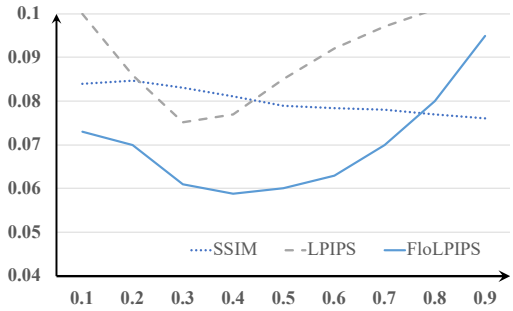


Figure 7. Different τ during inference. The SSIM is divided with 10 to enhance visibility.

ing the impact of the prior information on the quality of the interpolated frames.

Different loss functions. In Table 3, we also compare different loss functions in the training stage. Our PerVFI employs the bi-directional loss (denoted as Bi-D) as described in Equation (15). For comparison, we also train the network only using the negative log-likelihood (NLL) loss or the L1 loss. We have observed that training with only the NLL loss introduces certain noise in the output frames, particularly in misaligned regions. However, by introducing the perceptual loss, we are able to suppress this noise and generate better results. As shown in table 3, the L1 loss yields higher PSNR but lower VFIPS. To further highlight the superiority of our loss function, we provide additional visual samples in the appendix, demonstrating the effectiveness of our approach in producing visually superior results.

Different variances for latent codes. In Figure 7, we present results by sampling the latent codes with different variances τ during the inference stage. We can observe that the three metrics (SSIM, LPIPS and FloLPIPS) exhibit different optimal ranges. It is worth noting that the latent code can be flexibly customized to strike a balance between different metrics based on specific preferences.

5. Conclusion

We introduce PerVFI, a novel approach for video frame interpolation that decisively tackles issues of blur and ghosting artifacts, thereby significantly elevating perceptual quality. Our design incorporates an asymmetric blending module that strategically leverages features from two reference frames: one for primary content and the other for occlusion information. The model employs a normalizing flow-based generator with negative log-likelihood loss to capture the latent conditional distribution. The experiments validate its superiority in artifact reduction and high-quality generation.

6. Acknowledgment

The work was supported by the National Natural Science Foundation of China (62301310, U23A20391), the Shanghai Pujiang Program (22PJ1406800), and the Guang-Dong Basic and Applied Basic Research Foundation (2023A1515010644, 2021B151520011).

References

[1] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007. 6

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

[3] 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 Trans. Pattern Anal. Mach. Intell.*, 43(3):933–948, 2021. 2

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

[5] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10): 7029–7045, 2021. 6, 7

[6] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10): 7029–7045, 2022. 2

[7] 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

[8] Duolikun Danier, Fan Zhang, and David Bull. St-mfnet: A spatio-temporal multi-flow network for frame interpolation. In *CVPR*, 2022. 1, 2, 6, 7, 8

[9] Duolikun Danier, Fan Zhang, and David R. Bull. A subjective quality study for video frame interpolation. In *ICIP*, 2022. 2

[10] Duolikun Danier, Fan Zhang, and David R. Bull. Flolpips: A bespoke video quality metric for frame interpolation. In *PCS*, 2022. 6

[11] Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. *arXiv preprint arXiv:2303.09508*, 2023. 2, 6, 7

[12] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *arXiv preprint arXiv:2004.07728*, 2020. 6

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

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017. 2

[15] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *CVPR*, 2020. 2

[16] Qiqi Hou, Abhijay Ghildyal, and Feng Liu. A perceptual quality metric for video frame interpolation. In *ECCV*, 2022. 6

[17] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2020. 1, 2, 6, 7, 8

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

[19] Xin Jin, Longhai Wu, Guotao Shen, Youxin Chen, Jie Chen, Jayoon Koo, and Cheul-hee Hahm. Enhanced bi-directional motion estimation for video frame interpolation. In *WACV*, 2023. 1, 2

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5

[21] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv: 2012.08512*, 2021. 2

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[23] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018. 2

[24] 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

[25] 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

[26] Changlin Li, Guangyang Wu, Yanan Sun, Xin Tao, Chi-Keung Tang, and Yu-Wing Tai. H-VFI: hierarchical frame interpolation for videos with large motions. *arXiv preprint arXiv: 2211.11309*, 2022. 1, 2

[27] Wenhao Li, Guangyang Wu, Wenyi Wang, Peiran Ren, and Xiaohong Liu. Fastllve: Real-time low-light video enhancement with intensity-aware look-up table. In *ACMMM*, 2023. 1

[28] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, 2023. 6, 7

[29] Xiaohong Liu, Lei Chen, Wenyi Wang, and Jiying Zhao. Robust multi-frame super-resolution based on spatially weighted half-quadratic estimation and adaptive BTV regularization. *IEEE Trans. Image Process.*, 27(10):4971–4986, 2018. 1

[30] Xiaohong Liu, Lingshi Kong, Yang Zhou, Jiying Zhao, and Jun Chen. End-to-end trainable video super-resolution based on a new mechanism for implicit motion estimation and compensation. In *WACV*, 2020.

[31] Xiaohong Liu, Kangdi Shi, Zhe Wang, and Jun Chen. Exploit camera raw data for video super- resolution via hidden markov model inference. *IEEE Trans. Image Process.*, 30: 2127–2140, 2021.

[32] Yihao Liu, Liangbin Xie, Siyao Li, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *ECCVW*, 2020. 1, 2

[33] Guo Lu, Xiaoyun Zhang, Li Chen, and Zhiyong Gao. Novel integration of frame rate up conversion and HEVC coding based on rate-distortion optimization. *IEEE Trans. Image Process.*, 27(2):678–691, 2018. 1

[34] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *CVPR*, 2022. 1, 2, 6, 7

[35] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020. 2, 5

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

[37] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, 2020. 1, 2, 3, 6, 7

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

[39] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. BMBC: bilateral motion estimation with bilateral cost volume for video interpolation. In *ECCV*, 2020. 1, 2

[40] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *ICCV*, 2021. 2

[41] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 6, 7

[42] Zhihao Shi, Xiaohong Liu, Chengqi Li, Linhui Dai, Jun Chen, Timothy N. Davidson, and Jiying Zhao. Learning for unconstrained space-time video super-resolution. *IEEE Trans. Broadcast.*, 68(2):345–358, 2022. 1

[43] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video frame interpolation via generalized deformable convolution. *IEEE Trans. Multim.*, 24:426–439, 2022.

[44] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *ICCV*, 2021. 1, 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[46] 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*, 2012. 6

[47] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 3, 5, 6

[48] Wenyi Wang, Guangyang Wu, Weitong Cai, Liaoyuan Zeng, and Jianwen Chen. Robust prior-based single image super resolution under multiple gaussian degradations. *IEEE Access*, 8:74195–74204, 2020. 1

[49] Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex C. Kot. Low-light image enhancement with normalizing flow. In *AAAI*, 2022. 2

[50] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612, 2004. 6

[51] Guangyang Wu, Lili Zhao, Wenyi Wang, Liaoyuan Zeng, and Jianwen Chen. Pred: A parallel network for handling multiple degradations via single model in single image super-resolution. In *ICIP*, 2019. 1

[52] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. Accflow: Backward accumulation for long-range optical flow. In *ICCV*, 2023. 1

[53] 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

[54] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022. 3, 5

[55] Xiangyu Xu, Li Si-Yao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, 2019. 1, 2

[56] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.*, 127(8):1106–1125, 2019. 1, 2, 6

[57] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.*, 127(8):1106–1125, 2019. 7

[58] Guanghao Yin, Xinyang Jiang, Shan Jiang, Zhenhua Han, Ningxin Zheng, Xiaohong Liu, Huan Yang, Donglin Bai, Haisheng Tan, Shouqian Sun, Yuqing Yang, Dongsheng Li, and Lili Qiu. Online video streaming super-resolution with adaptive look-up table fusion. *arXiv preprint arXiv: 2303.00334*, 2023. 1

[59] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, 2023. 1, 6, 7, 8

[60] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6