Towards Robust Video Frame Interpolation with Long-Term Propagation

Ziqi Huang[1][0000−0001−8008−5873], Kelvin C.K. Chan[1][0000−0002−5456−8991], Bihan Wen[2][0000−0002−6874−6453], and Ziwei Liu[1][0000−0002−4220−5958]

[1] S-Lab, Nanyang Technological University
[2] School of EEE, Nanyang Technological University
{ziqi002, chan0899, bihan.wen, ziwei.liu}@ntu.edu.sg

Abstract. Video frame interpolation aims to synthesize non-existent frames between two consecutive frames in a video, and its importance can be seen from its wide applications in computer vision. The key to video frame interpolation is predicting the intermediate motions between two given frames, so that the synthesized frames are coherent with the input video. However, the existing approaches of imposing assumptions on motions, such as linear and quadratic trajectories, are not generalizable to complex motions in real-world videos. In this work, we propose **long-term propagation** for **robust and general motion prediction** in video frame interpolation. To more thoroughly understand the motion trajectories, we propose to implicitly track the motion paths through a long sequence of video frames. The motion features are then used to refine the motion predicted from the primitive motion assumptions. Our proposed long-term propagation of motion can be ***easily integrated into existing video frame interpolation approaches***. Quantitative and qualitative results demonstrate that long-term propagation effectively ***improves interpolation performance*** when incorporated into various state-of-the-art baselines. In addition, we present a series of analytical experiments to study the mechanism, advantages, and limitations of long-term propagation in video frame interpolation to inspire future works.

Keywords: Video frame interpolation · Motion modeling

1 Introduction

Given a low-frame-rate video, we can apply ***video frame interpolation*** as a post-processing step to synthesize non-existent frames between two consecutive video frames. Video frame interpolation is mainly applied to increase the frame rate and produce videos with smoother transitions. It is also useful in video compression [4] and super-slow-motion generation [9, 12].

Video frame interpolation has attracted considerable attention in the computer vision community. There have been three major classes of solutions: flow-based [2,3,7,8,12,16,22,23,26,30,31], kernel-based [6,24,25], and phase-based [19,

20] methods. In recent years, ***flow-based methods*** have seen notable advancement and achieved relatively impressive results. This class of methods usually consists of three steps: ***1)*** estimating the optical flows from the input frames to the non-existent frames, ***2)*** warping the input frames according to the reversed flow estimations, and ***3)*** fusing the warped frames. Since the hidden frame is unavailable at inference time, the intermediate flows (*i.e.*, the optical flows from the input frames to the non-existent hidden frame) are not directly accessible, and can only be estimated under certain assumptions.

Most flow-based methods [2, 12, 22, 23, 26] assume a linear motion model and approximate the intermediate flows using the two nearest given frames. A more recent work QVI [30] relaxes the linearity constraint and uses the four nearest input frames to model quadratic motions. However, *these motion models assuming constant velocities or accelerations severely simplified real-world situations*, and does not reflect the complexity of real-world motions, potentially resulting in misalignment and unnatural movements in the outputs. Recognizing the complexity of real-life object motions, which cannot be fully captured by simple polynomial models (*e.g.*, linear, quadratic, or even higher-order), we are interested in developing *a more robust and general motion estimation module*.

To this end, we propose ***long-term propagation*** for intermediate motion estimation in video frame interpolation, where we *understand the object motion through implicitly tracking it*. Specifically, we propagate motion features along a sequence of video frames. We adopt a recurrent neural network (RNN) to implicitly learn the motion trajectory in order to accurately estimate the intermediate optical flows. Our approach models motion without relying on prior assumptions about its characteristics, offering a more general method of motion modeling. By leveraging propagated motion information from the entire video sequence rather than just two input frames, we capture the true motion more accurately, and offer improved interpolation results.

Our work begins with a preliminary study of several existing motion models to better understand how motion models affect interpolation results, setting the stage for an in-depth exploration of long-term propagation. We then evaluate long-term propagation both qualitatively and quantitatively to show its effectiveness in improving video frame interpolation performance. We also investigate different factors that might influence the effectiveness of long-term propagation, and discuss its underlying mechanism, advantages, and limitations. Recommendations for future research are also provided.

Our contributions are summarized as follows:

- We propose ***long-term propagation*** for more robust video frame interpolation, moving beyond existing assumptions of linear or quadratic motions. This approach allows for more accurate intermediate motion estimation, leading to improved interpolation outcomes.
- Through comprehensive experiments, we demonstrate the superiority of long-term propagation in accurate motion estimation and high-quality frame interpolation.

– Long-term propagation is a plug-and-play module that can be easily integrated into any flow-based video frame interpolation methods for more accurate motion estimation and better interpolation quality.

2 Related Work

2.1 Video Frame Interpolation

Video frame interpolation is a challenging problem in computer vision, whose objective is to generate frames that do not originally exist between two consecutive input frames. This field has seen diverse approaches, broadly categorized into flow-based, kernel-based, and phase-based methods.

Flow-Based Methods. Flow-based methods [2,3,7,8,12,15,16,22,23,26,30,31] first estimate the optical flows from the input frames to the non-existent frames, then synthesize preliminary frames via spatial warping, and finally fuse the preliminary frames guided by occlusion masks [3, 12, 31] or depth information [2]. Several techniques are shown effective in improving interpolation performance, including softmax splatting [23], contextual information [22], and cycle consistency [26]. Compared to backward warping [2, 8, 12, 16, 30], forward warping [1, 22, 23] is less adopted due to various challenges, such as multiple pixels being mapped to the same pixel in the non-existent frame. The performance of flow-based methods is limited by the accuracy of flow estimation to the non-existent frames. Most methods [2, 12, 22, 23, 26] assume uniform linear motion between two consecutive frames, which is far from ideal for real-world motions. Recent works relax this assumption by proposing quadratic [16,30] and cubic [7] motion models. In contrast, our *long-term propagation* does not pose polynomial assumptions on intermediate motions, and implicitly track the motion trajectories by propagating motion features through a long sequence of video frames, achieving more accurate motion estimation and interpolation quality.

Kernel-Based Methods. Kernel-based methods [6, 24, 25] use spatially adaptive filters to sample from neighboring pixels in input frames to synthesize a non-existent frame. Early attempts, such as DVF [17], first estimate a 3D voxel flow across space and time, followed by a trilinear sampling operation on the voxel volume. This approach implicitly assumes linear motion. More recent works like FLAVR [13] use 3D space-time convolutions to learn motion and occlusion patterns, facilitating direct interpolation.

Phase-Based Methods. Phase-based methods [19, 20] consider each frame as a linear combination of wavelets, and interpolate the phase and magnitude of individual components. Since phase is modeled as a linear-time function [20], motions are implicitly assumed to be linear.

2.2 Propagation

Propagation is an important aspect of video restoration because it allows leveraging information in other frames to enrich the understanding of video content.

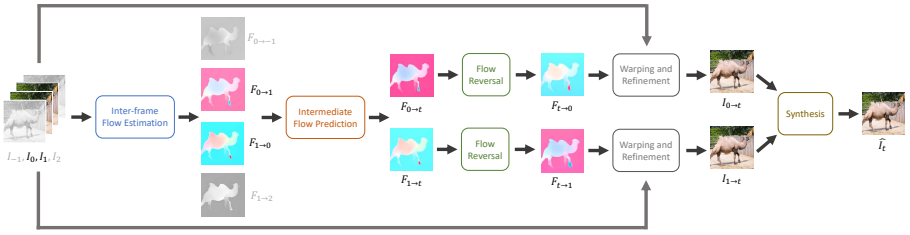


Fig. 1: Flow-Based Frame Interpolation Pipeline. Flow-based methods usually perform interpolation in four steps: (1) intermediate flow estimation, (2) flow reversal (optional), (3) warping, and (4) synthesis. Since most existing methods only use two input frames I_0 and I_1, we present I_{-1} and I_2 in gray color. In this work, to better estimate the intermediate flows, we work on: *(1) using more than two input frames to gather long-term motion information, and (2) better alternatives to the "Intermediate Flow Prediction" module.* (Images and optical flows in the figure are obtained from the original paper [30] for illustration purposes.)

Existing video frame interpolation methods [2,3,6–8,12,16,17,19,20,22–26,30,31] typically reply on local information propagation, using only the nearest two to four frames. This approach results in a limited temporal receptive fields and confines the amount of motion information that can be utilized. In contrast, we propose to propagate information across a long temporal range to better solve the problem of video frame interpolation. Long-term propagation has been applied in other tasks like video super-resolution [10, 11] and has been shown effective. In this work, we would like to initiate the study on long-term propagation in video frame interpolation.

3 Method

3.1 Preliminary Study on Motion Models

To deepen our understanding of how motion models influence video frame interpolation outcomes, we initiated a preliminary investigation into the impact of selecting various motion models on interpolation quality. Specifically, we introduce and study several ***polynomial motion models***, including *Linear*, *Quadratic*, *Quadratic Least Squares*, and *Cubic*. Detailed theoretical formulation of these model models, and their implementation details are provided in the supplementary file.

Problem Formulation. Given two consecutive input frames I_0 and I_1 in a video, the objective of video frame interpolation is to synthesize a non-existent frame I_t, where $t \in (0, 1)$. In this work, we focus on the flow-based approach. An example of flow-based approaches is shown in Fig. 1. In general, flow-based methods [2, 3, 7, 8, 12, 16, 22, 23, 26, 30, 31] first estimate the optical flows, $F_{0\to t}$ and $F_{1\to t}$, from the input frames (*i.e.*, I_0 and I_1) to the non-existent frame (*i.e.*, the frame to be interpolated, I_t). The estimated flows are used to warp the given

Table 1: Interpretation of Motion Models. (1) **Order**: the polynomial degree that models the motion trajectory over time, (2) **Formula**: the mathematical representation for calculating an object's location $x(t)$ at a given time t, where a, b, c, and d are constant coefficients that define the motion's characteristics, (3) **Physical Meaning**: the nature of the motion that each model aims to capture, (4) **#Frames**: the number of timestamps t (*i.e.*, frames) needed to solve the constant coefficients in the motion formula.

Motion Model	Order	Formula	Physical Meaning	#Frames
Linear	1	$x(t) = at + b$	constant velocity	2
Quadratic	2	$x(t) = at^2 + bt + c$	constant acceleration	3
Quadratic Least Squares	2	$x(t) = at^2 + bt + c$	constant acceleration	4
Cubic	3	$x(t) = at^3 + bt^2 + ct + d$	linear-time acceleration	4
Long-Term Propagation (Ours)	N.A.	N.A. (not applicable)	no constraint	any

frames to produce two preliminary source frames $I_{0\to t}$ and $I_{1\to t}$, which are then fused to produce the final output $\hat{I}_t$.

In this work, we aim to explore better approaches (*i.e.*, long-term propagation) to estimate the intermediate optical flows $F_{0\to t}$ and $F_{1\to t}$. Particularly, we will explore different alternatives to the "Intermediate Flow Prediction" module in Fig. 1.

Polynomial Motion Models. We explore the effects of different polynomial motion models on the final interpolation results. We conduct experiments using four models: *Linear*, *Quadratic*, *Quadratic Least Squares*, and *Cubic*. Table 1 presents an overview of these four models, and Fig. 2 illustrates the process of computing the intermediate optical flows according to each motion model. The supplementary file offers detailed formulas for calculating intermediate optical flows for each model and their comprehensive analysis.

Analysis of Preliminary Study. We evaluate the performance of video frame interpolation when using four different polynomial motion models, on the Adobe240 dataset [28]. We report the PSNR and SSIM of the synthesized frames with respect to ground-truth frames in Table 2. For each data sample, given I_{-1}, I_0, I_1, I_2, we use the trained baselines to synthesize seven frames at $t = 0.125$, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875 respectively, where "avg" denotes the mean of PSNR or SSIM at all seven possible t values, while "center" denotes PSNR and SSIM at $t = 0.5$ only. We provide more experimental implementations and detailed result analysis in the supplementary file. Through the preliminary study, we observe that ***the choice of motion models can largely affect interpolation results***. For instance, the *Quadratic* intermediate flow estimation method (PSNR: 33.02) outperforms the *Linear* method (PSNR: 30.99) by 1.03 in terms of PSNR. This result encourages us to seek better motion models and intermediate flow estimation methods to further improve interpolation performance. In subsequent sections, we introduce our long-term propagation approach for estimating intermediate flows, and demonstrate its effectiveness to achieve more robust motion estimation and video frame interpolation.

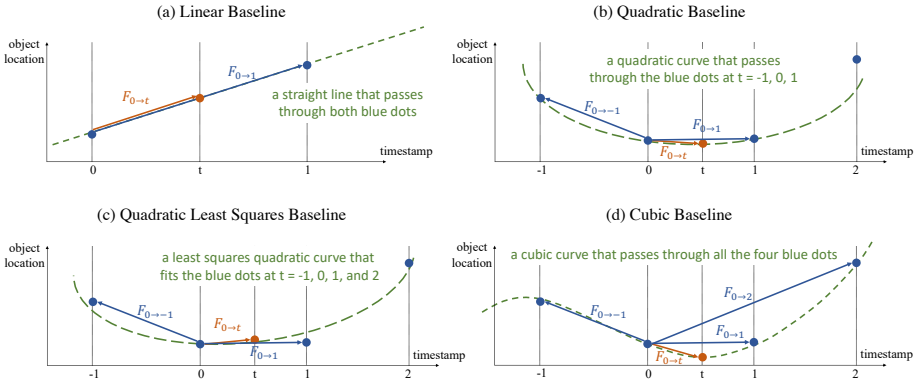


Fig. 2: Illustrations of Intermediate Flow Prediction. In each baseline, we first predict optical flows between input frames (*i.e.*, the blue arrows) using off-the-shelf optical flow estimators (*e.g.*, [27]). We then fit the object locations (*i.e.*, blue dots) using a motion trajectory curve (*i.e.*, the green dashed line) according to the motion model. After that, we compute the object location at time t (*i.e.*, the orange dot) on the motion trajectory curve. Finally, we take the object displacement from time 0 to t as the intermediate optical flow $F_{0\to t}$ (*i.e.*, the orange arrow).

Table 2: Quantitative Comparisons of Polynomial Motion Models (Adobe240 Dataset [28]). We conduct a preliminary study on the video frame interpolation performance with different polynomial motion models. We find that the choice of motion models can largely affect interpolation results.

Baseline	PSNR (avg) ↑	PSNR (center) ↑	SSIM (avg) ↑	SSIM (center) ↑
Linear	30.99	29.46	0.9329	0.9113
Quadratic	**33.02**	**32.32**	**0.9558**	**0.9504**
Quadratic Least Squares	32.76	31.97	0.9537	0.9473
Cubic	32.70	31.84	0.9537	0.9467

3.2 Long-Term Propagation

Recurrent Network Overview. Our goal is to obtain an accurate estimation of the intermediate optical flows. We use a recurrent network, as illustrated in Fig. 3, to propagate motion information through a sequence of frames, and use this knowledge to guide the intermediate flow estimation step shown in Fig. 1.

Given a sequence of input frames $\mathbf{I} = [I_0, I_1, I_2, ..., I_n]$, we propagate the motion information forward, from I_0 to I_n. Specifically, the operation of the i-th recurrent step is as follows:

$$r_i^f, h_{i+1}^f = R_f(h_i^f, t, F_{i\to i+1}, F_{i\to i+t}), \tag{1}$$

where R_f is the forward propagation branch, h_i^f is the hidden state that carries the motion information forward, $t \in (0, 1)$ is the timestamp of the unknown

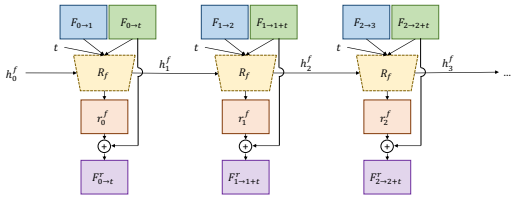


Fig. 3: Long-Term Propagation. We propose long-term propagation to achieve robust interpolation. Specifically, we propagate long-term motion information using a recurrent network. At each recurrent step, the network R_f takes the inter-frame flow (*e.g.*, $F_{0\to1}$), the preliminary intermediate flow (*e.g.*, $F_{0\to t}$), the hidden state (*e.g.*, h_0^f), and the timestamp t as inputs, and predicts a residue r that refines the preliminary intermediate flow. The hidden state propagates the motion information through the video sequence.

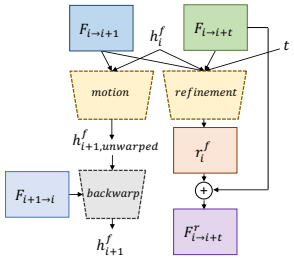


Fig. 4: Recurrent Step. Each recurrent step consists of three modules: *motion*, *refinement*, and *backwarp*. (1) The *motion* module takes motion information as input, and update the hidden state to propagate long-term motion information forward. (2) The *refinement* module predicts the residue that refines the intermediate flow. (3) The *backwarp* module aligns the updated hidden state to the next temporal interval for future information propagation.

frame I_{i+t} to be synthesized, $F_{i\to i+1}$ is the inter-frame optical flow between the input frames I_i and I_{i+1}, which is predicted by an off-the-shelf flow estimation model, and $F_{i\to i+t}$ is the preliminary intermediate flow to be refined by long-term propagation. Each recurrent step outputs the residue r_i^f, and the updated hidden state h_{i+1}^f. The residue r_i^f is used to refine the preliminary intermediate flow $F_{i\to i+t}$:

$$F_{i\to i+t}^r = F_{i\to i+t} + r_i^f, \tag{2}$$

where $F_{i\to i+t}^r$ is the refined intermediate flow. We choose to refine a preliminary intermediate flow instead of directly predicting the refined intermediate flow because of two reasons: (1) A coarse intermediate flow obtained from existing baselines such as *Linear* and *Quadratic* provides rough motion information of the object, serving as a good starting point, and (2) predicting only the residue to coarse motions eases training and accelerates convergence.

Bidirectional Propagation. To warp both source frames I_i and I_{i+1} to the target timestamp t, we need two intermediate optical flows $F_{i\to i+t}^r$ and $F_{i+1\to i+t}^r$. Therefore, other than the forward propagation branch which estimates the forward intermediate flow $F_{i\to i+t}^r$, we also need a backward branch to estimate the backward intermediate flow $F_{i+1\to i+t}^r$ similarly:

$$r_i^b, h_i^b = R_f(h_{i+1}^b, t, F_{i+1\to i}, F_{i+1\to i+t}), \tag{3}$$

$$F_{i+1\to i+t}^r = F_{i+1\to i+t} + r_i^b. \tag{4}$$

Both $F^r_{i\to i+t}$ and $F^r_{i+1\to i+t}$ will be used in the subsequent stages of the flow-based framework (refer to Fig. 1). Both forward and backward branches use the same recurrent step R_f.

Recurrent Step. For clarity, we detail the recurrent steps using the forward branch as an example, and the backward branch operates in a similar fashion.

As illustrated in Fig. 4, the recurrent step consists of three key modules: *motion*, *refinement*, and *backwarp*:

$$h^f_{i+1,unwarped} = motion(h^f_i, F_{i\to i+1}), \tag{5}$$

$$r^f_i = refinement(h^f_i, t, F_{i\to i+1}, F_{i\to i+t}), \tag{6}$$

$$h^f_{i+1} = backwarp(h^f_{i+1,unwarped}, F_{i+1\to i}). \tag{7}$$

The *motion* module. As shown in Eq. 5, the *motion* module aims to update the hidden state h^f_i to $h^f_{i+1,unwarped}$ in order to pass the motion information along. It does not take the timestamp t or the intermediate flow $F_{i\to i+t}$ as input, ensuring that the hidden state's update is not influenced by specific timestamps.
The *refinement* module. The *refinement* module (Eq. 6) predicts a residue r^f_i to refine the preliminary intermediate flow $F_{i\to i+t}$. Since the value of $F^r_{i\to i+t}$ varies with the timestamp t, the residue r^f_i should also depend on t. Therefore, the *refinement* module takes both the timestamp t and preliminary intermediate flow $F_{i\to i+t}$ as inputs. To guide intermediate flow refinement, the hidden state h^f_i provides long-term motion information, while the inter-frame flow $F_{i\to i+1}$ provides local motion information.
The *backwarp* module. Since the hidden state $h^f_{i+1,unwarped}$ is aligned to I_i rather than I_{i+1}, we need to warp it to align with I_{i+1} for the next recurrent step. As shown in Eq. 7, we use $F_{i+1\to i}$ to *backwarp* (*i.e.*, the backward spatial warping operation) the hidden state $h^f_{i+1,unwarped}$ to obtain h^f_{i+1}, which is aligned to I_{i+1}.

3.3 Training Scheme

We train the network in three stages:

1. **Pretrain**: We train a preliminary model (*e.g.*, *Quadratic+PWCNet*, which will be discussed in detail in Sec. 4), which estimates intermediate flows using one of the motion assumptions discussed in Sec. 3.1.
2. **Plug in**: We integrate the long-term propagation module (*i.e.*, the recurrent network in Sec. 3.2) into the preliminary model for intermediate flow refinement. We keep the weights of the preliminary model frozen, and only train the long-term propagation module.
3. **Finetune**: We perform finetuning on the entire network to optimize its performance.

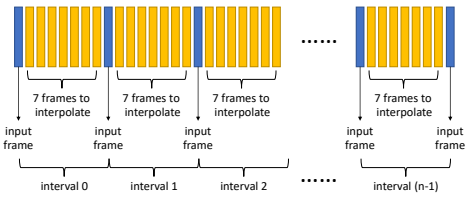


Fig. 5: Interval Definition. The blue frames are the given input frames. The yellow frames are the ones to be synthesized. Each segment has n intervals, where in each interval we synthesize seven in-between frames. When $n = 10$, the total number of frames per segment (including input frames) is 81.

4 Experiments

4.1 Implementation Details

Dataset. Due to the long training time[3] experienced during the preliminary study, we use a subset of the REDS dataset [21] for faster training and evaluation in our subsequent experiments. We use 100 video clips (*i.e.*, clip 000 to 099) as the training dataset, with spatial random crops for data augmentation. We use four video clips (*i.e.*, clip 240, 241, 246, and 257) as the evaluation dataset. We follow BasicVSR [5]'s choice on the four evaluation clips. These four clips have diverse video content, serving as a good benchmark for model performance.

Interpolation Intervals. For each interpolation interval (refer to Fig. 5 for its definition), we synthesize seven in-between frames (*i.e.*, $t \in \{0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875\}$) during both training and evaluation. We use ten intervals as one segment, resulting in 81 frames per segment. Information only propagates within a segment, and does not propagate across two different segments. During training, we use all possible 81-frame segments in each 500-frame video clip, and randomly flip the segments temporally for data augmentation. At test time, we only use the first 81-frame segment in each video clip for evaluation.

Hyperparameters. The hyperparameters for the three training stages are detailed below:

1. **Pretrain**: We adopt the Adam optimizer [14], initialize the learning rate as 10^{-4} and further decrease it by a factor of 0.1 at the end of the 50^{th} and 75^{th} epoch. We freeze the weights of the pretrained optical flow estimation model for 100 epochs before finetuning.
2. **Plug in**: We choose 5×10^{-5} as the initial learning rate, and use the Cosine Annealing scheme [18] for learning rate decay. For both the *motion* and *refinement* modules, we use five residual blocks.
3. **Finetune**: We choose 3×10^{-5} as the initial learning rate, and use the Cosine Annealing scheme [18] for learning rate decay. We apply a learning rate multiplier of 0.25 to the off-shelf optical flow estimation model.

[3] For instance, training the *Quadratic Least Squares* model takes 17 GPU days on 32-GB V100 GPUs.

The design choices (*e.g.*, hyperparameters and optimizer schemes) are carefully determined through extensive comparative experiments. We provide these experimental set-up and results in the supplementary file.

4.2 Comparison Methods

For the first stage (refer to Sec. 3.3) of our training scheme, we experiment on different preliminary models to assess (1) the ability of long-term propagation to enhance interpolation results across different preliminary models, and (2) the impact of these models' characteristics on the effectiveness of long-term propagation. We adopt (1) *Quadratic+PWCNet*, (2) *Quadratic+RAFT*, and (3) *Linear+PWCNet* as the preliminary models in the subsequent experiments. The naming convention used is *"Motion Model + Optical Flow Estimator"*.

Quadratic+PWCNet. We follow the QVI [30] baseline, employing the *Quadratic* motion assumption, and using the pretrained PWCNet [27] as the optical flow estimator.

Quadratic+RAFT. RAFT [29] is one of the state-of-the-art optical flow estimation methods. We use *Quadratic+RAFT* alongside *Quadratic+PWCNet* to evaluate the influence of different pretrained flow estimators on long-term propagation's performance.

Linear+PWCNet. Since *Linear* and *Quadratic* are two commonly adopted motion models, we use *Linear+PWCNet* alongside *Quadratic+PWCNet* to analyze how different preliminary motion models can affect the PSNR improvement introduced by long-term propagation.

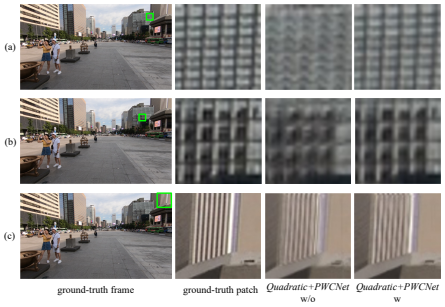
4.3 Quantitative Comparisons

In Table 3, we present a comparison of PSNR for preliminary models before and after applying long-term propagation. "PSNR w/o" refers to results obtained using models from the first stage (refer to Sec. 3.3), whereas "PSNR w" represents results from the third stage, incorporating long-term propagation. We see that long-term propagation introduces PSNR improvement to all the three preliminary models. This indicates that *long-term is effective in improving interpolation accuracy*. Furthermore, a relatively less robust preliminary model tends to benefit more from long-term propagation. For instance, *Linear+PWCNet*, which adopts a simpler motion model compared to *Quadratic+PWCNet*, experiences a larger PSNR increase. Similarly, *Quadratic+PWCNet* sees a greater PSNR improvement over *Quadratic+RAFT*. This pattern will be explored more comprehensively in Sec. 4.5.

4.4 Qualitative Comparisons

The qualitative results on the REDS dataset [21] evaluation clips highlight the impact of long-term propagation on video frame interpolation.

(a) Qualitative Comparisons (Alignment). The green lines are reference lines that show the discrepancy in object positioning relative to the ground truth when long-term propagation is absent. **(1) "w/o"**: Before applying long-term propagation, objects are not well aligned with the ground-truth. **(2) "w":** The incorporation of long-term propagation leads to a noticeable improvement in object alignment, suggesting that our method enhances the accuracy of intermediate flow estimation, which is the source for maintaining alignment with the ground truth.



(b) Qualitative Comparisons (Blur). The absence (*i.e.*, "w/o") of long-term propagation results in blurred synthesis outcomes, while applying it (*i.e.*, "w") yields clearer and sharper images. The cause of bluriness is that the estimated intermediate flows from two neighboring frames fail to point to the same pixel locations, and fusing the warpings results in ghosting residues which appear to be bluriness. Thus, the enhanced clarity and sharpness with long-term propagation indicate significant improvements in intermediate flow estimation accuracy.

Fig. 6: Qualitative Comparisons. We show the comparisons of before and after applying long-term propagation.

In Fig. 6a, the green lines are reference lines that show the discrepancy in object positions relative to the ground truth when long-term propagation is absent. We observe that before applying long-term propagation, objects are not well aligned with the ground truth. The incorporation of long-term propagation leads to a noticeable improvement in object alignment, suggesting that our method enhances the accuracy of intermediate flow estimation, which is the source for maintaining alignment with the ground truth.

Fig. 6b shows that the absence of long-term propagation results in blurry outputs, while applying it yields clearer and sharper images. We conjecture that the two preliminary source frames (*i.e.*, $I_{0\to t}$ and $I_{1\to t}$) warped from the two input frames (*i.e.*, I_0 and I_1) respectively are not perfectly aligned, thus fusing them results in blurry outputs. The misalignment of preliminary source frames is due to inaccurate intermediate flow estimation. That is, $F_{i\to i+t}$ and $F_{i+1\to i+t}$ fail to accurately point to the same pixel locations.

Please refer to the ***demo video*** *in our supplementary materials for more qualitative results and comparisons.*

4.5 Improved Intermediate Flow Estimation

We conduct analytical experiments to understand the mechanism behind the improvements in interpolation results brought by long-term propagation. Specifically, we hope to answer these questions: (1) *How does long-term propagation improve PSNR? That is, what is the source of PSNR improvement?* (2) *Does*

Table 3: PSNR with and without Long-Term Propagation. We report the PSNR of each preliminary model, without (denoted as "PSNR w/o") and with (denoted as "PSNR w") long-term propagation respectively.

Preliminary Model	PSNR w/o	PSNR w	PSNR Increase ↑
Linear+PWCNet	26.90	**28.03**	1.14
Quadratic+PWCNet	28.33	**28.80**	0.47
Quadratic+RAFT	28.97	**29.12**	0.16

long-term propagation offer a more accurate estimation of the motions from source frames to the intermediate frames?

To address these questions, we study the relationship between long-term propagation and the accuracy of intermediate flow estimation. We employ End Point Error (EPE) as the metric for flow estimation accuracy. EPE is obtained by first computing the Euclidean distance between the estimated flow and the ground-truth flow at each pixel location, and then taking the average of all Euclidean distances in an image. A lower EPE indicates a more accurate flow estimation. For generating the ground-truth optical flows, we use RAFT [29], one of the state-of-the-art optical flow estimators.

Table 4 shows the EPE of the intermediate flows predicted by the interpolation networks, before and after applying long-term propagation. Specifically, we compute the EPE for both $F^r_{i\to i+t}$ and $F^r_{i+1\to i+t}$, and then average these two values to report "avg EPE" for each model. Notably, across all models, incorporating long-term propagation results in a reduction of EPE, compared to the counterpart before adding long-term propagation. This reduction in EPE demonstrates the effectiveness of long-term propagation in producing more accurate intermediate flows. Consequently, *the observed PSNR improvements across all models further support that the enhanced performance can be attributed to more accurate intermediate flow estimations facilitated by long-term propagation.*

Table 5 illustrates how the accuracy of intermediate flow predictions—measured by End Point Error (EPE) without the use of long-term propagation—affects the degree of PSNR improvement when long-term propagation is applied. We observe that a higher initial EPE, reflecting less accurate flow predictions, correlates with a more substantial PSNR increase introduced by long-term propagation. This suggests that a larger initial EPE signifies greater potential for improvement in flow estimation accuracy. Consequently, this pattern supports the hypothesis that *long-term propagation enhances video frame interpolation results by enabling more accurate intermediate flow estimations.*

4.6 Further Analysis

We further study how different factors, namely sequence length (*i.e.*, temporal receptive field) and motion size, affect long-term propagation's effectiveness. Detailed analysis are provided in the supplementary file.

Table 4: Accuracy of Intermediate Flow Estimation. For each preliminary model before and after applying long-term propagation, we present the EPE of $F^r_{i\to i+t}$, $F^r_{i+1\to i+t}$, and the average of these two. Lower "avg EPE" indicates more accurate intermediate flow estimation. We find that by adding long-term propagation to a preliminary model, the PSNR increases, and the "avg EPE" decreases, which implies that long-term propagation improves the accuracy of intermediate flow estimation.

Models	**PSNR** ↑	$F^r_{i\to i+t}$ **EPE** ↓	$F^r_{i+1\to i+t}$ **EPE** ↓	**avg EPE** ↓
Linear+PWCNet w/o	26.90	1.52	1.51	1.51
Linear+PWCNet w	**28.03**	1.40	1.58	**1.49**
Quadratic+PWCNet w/o	28.33	1.39	1.53	1.46
Quadratic+PWCNet w	**28.80**	1.37	1.38	**1.37**
Quadratic+RAFT w/o	28.97	1.21	1.38	1.29
Quadratic+RAFT w	**29.12**	1.16	1.33	**1.25**

Table 5: Initial EPE vs PSNR Increase. For each preliminary model, we show the "PSNR increase" brought by long-term propagation, and End Point Error (EPE) of intermediate optical flows as predicted by the network prior to the application of long-term propagation. We observe that a larger initial EPE is linked with a larger PSNR increase, indicating long-term propagation's ability to correct the initially inaccurate intermediate flow estimations, thereby directly contributing to improved interpolation quality.

Models	**PSNR w/o**	**PSNR w**	**PSNR increase** ↑	**EPE w/o** ↓
Linear+PWCNet	26.90	28.03	**1.14**	**1.51**
Quadratic+PWCNet	28.33	28.80	**0.47**	**1.46**
Quadratic+RAFT	28.97	29.12	**0.16**	**1.29**

Sequence Length. We studied how sequence length, which defines the temporal receptive field, affects interpolation results using the *Quadratic+PWCNet* model with long-term propagation. Fig. 8 shows that PSNR increases with sequence length up to 19 intervals, beyond which it plateaus, suggesting that longer sequences improve motion estimation up to a point. However, sequence lengths beyond 20 intervals show no further PSNR improvement, indicating that additional information becomes irrelevant and increases GPU memory demand and inference time. The optimal sequence length for our scenario is around 19 intervals.

Motion Size. We evaluated the impact of motion size on the effectiveness of long-term propagation by quantifying motion size using the optical flow length between consecutive frames. In Fig. 7, we plotted the PSNR increase due to long-term propagation against flow length for three models. The scatter plots show little correlation, indicating that long-term propagation remains robust across different motion sizes. Interestingly, the most significant PSNR improvements

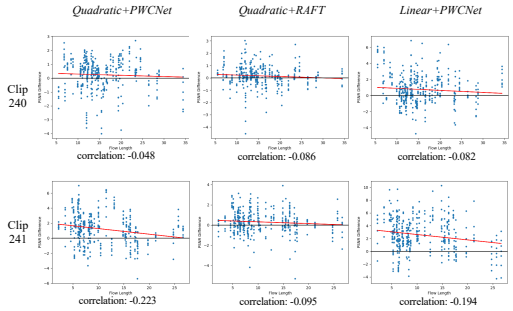


Fig. 7: Long-Term Propagation Effectiveness vs Motion Size. In each scatter plot, we show the PSNR increase brought by long-term propagation (y-axis) against motion size (x-axis). Different scatter plot represents a different preliminary model and evaluation video clip. We observe that long-term propagation is robust across different motion sizes.

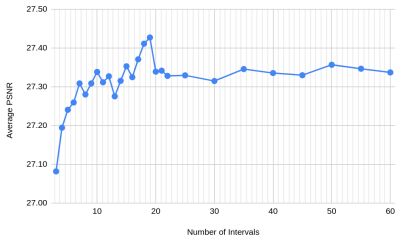


Fig. 8: Long-Term Propagation vs Sequence Length. Interpolation performance (*i.e.*, PSNR) increases with sequence lengths from one to 19 intervals, beyond which the increase plateaus.

occur in the medium to small motion range, where long-term propagation is particularly effective.

5 Conclusion

In this work, we propose ***long-term propagation*** for robust video frame interpolation. Unlike existing methods that rely on strong assumptions on between-frame motions, we propagate the motion information through the video sequence for general and robust motion estimation. Our method consistently yields more accurate intermediate motion estimations, thereby improving video frame interpolation quality. As a plug-and-play module, long-term propagation can be seamlessly integrated into existing video frame interpolation methods for performance gain. Beyond the introduction of long-term propagation, we also conducted extensive experimental analysis on the influence of motion models on video frame interpolation, explored the underlying mechanisms of long-term propagation's effectiveness. We believe our study on long-term propagation is beneficial for future research in video frame interpolation, motion estimation, and the broader domain of video restoration.

Acknowledgments. This study is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOET2EP20221-0012), NTU NAP, and under the RIE2020 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

1. Baker, S., Roth, S., Scharstein, D., Black, M.J., Lewis, J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV (2007)
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: CVPR (2019)
3. Bao, W., Lai, W.S., Zhang, X., Gao, Z., Yang, M.H.: MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. IEEE TPAMI (2018)
4. Bégaint, J., Galpin, F., Guillotel, P., Guillemot, C.: Deep frame interpolation for video compression. In: 2019 Data Compression Conference (DCC) (2019)
5. Chan, K.C., Wang, X., Yu, K., Dong, C., Loy, C.C.: BasicVSR: The search for essential components in video super-resolution and beyond. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2021)
6. Cheng, X., Chen, Z.: Video frame interpolation via deformable separable convolution. AAAI (2020)
7. Chi, Z., Mohammadi Nasiri, R., Liu, Z., Lu, J., Tang, J., Plataniotis, K.N.: All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In: ECCV (2020)
8. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: RIFE: Real-time intermediate flow estimation for video frame interpolation. arXiv preprint arXiv:2011.06294 (2020)
9. Huawei: 7680 fps super slow-motion videos — make time stand still. `https://consumer.huawei.com/en/community/details/7680-FPS-Super-Slow-Motion-Videos-Make-Time-Stand-Still/topicId_114694/` (2020)
10. Isobe, T., Jia, X., Gu, S., Li, S., Wang, S., Tian, Q.: Video super-resolution with recurrent structure-detail network. In: ECCV (2020)
11. Isobe, T., Zhu, F., Wang, S.: Revisiting temporal modeling for video super-resolution. In: BMVC (2020)
12. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In: CVPR (2018)
13. Kalluri, T., Pathak, D., Chandraker, M., Tran, D.: FLAVR: Flow-agnostic video representations for fast frame interpolation. arXiv preprint arXiv:2012.08512 (2021)
14. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
15. Li, Z., Zhu, Z.L., Han, L.H., Hou, Q., Guo, C.L., Cheng, M.M.: Amt: All-pairs multi-field transforms for efficient frame interpolation. In: CVPR (2023)
16. Liu, Y., Xie, L., Li, S., Sun, W., Qiao, Y., Dong, C.: Enhanced quadratic video interpolation. In: ECCV Workshop (2020)
17. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: ICCV (2017)
18. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
19. Meyer, S., Djelouah, A., McWilliams, B., Sorkine-Hornung, A., Gross, M., Schroers, C.: Phasenet for video frame interpolation. In: CVPR (2018)
20. Meyer, S., Wang, O., Zimmer, H., Grosse, M., Sorkine-Hornung, A.: Phase-based frame interpolation for video. In: CVPR (2015)
21. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: CVPRW (2019)

22. Niklaus, S., Liu, F.: Context-aware synthesis for video frame interpolation. In: CVPR (2018)
23. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: CVPR (2020)
24. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: CVPR (2017)
25. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. In: ICCV (2017)
26. Reda, F., Sun, D., Dundar, A., Shoeybi, M., Liu, G., Shih, K., Tao, A., Kautz, J., Catanzaro, B.: Unsupervised video interpolation using cycle consistency. In: ICCV (2019)
27. Ren, Z., Gallo, O., Sun, D., Yang, M.H., Sudderth, E.B., Kautz, J.: A fusion approach for multi-frame optical flow estimation. In: WACV (2019)
28. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: CVPR (2017)
29. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020)
30. Xu, X., Li, S., Sun, W., Yin, Q., Yang, M.H.: Quadratic video interpolation. In: NeurIPS (2019)
31. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV (2019)