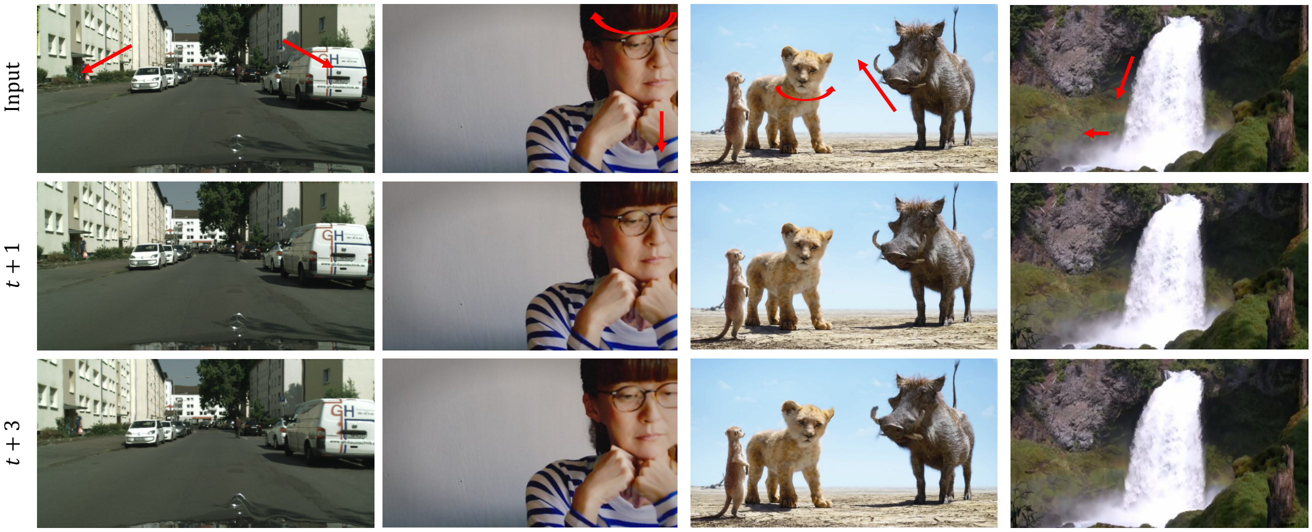
Optimizing Video Prediction via Video Frame Interpolation

Yue Wu Qiang Wen Qifeng Chen
The Hong Kong University of Science and Technology



(a) Driving scene (b) Human motion (c) Animation (d) Natural scene

Figure 1. Our method can produce plausible video prediction results in diverse scenarios without external training, such as driving scenes, human motion, animation, and natural scenes. We use red arrows to indicate the motions from input frames. Video results are presented in the supplementary material.

Abstract

Video prediction is an extrapolation task that predicts future frames given past frames, and video frame interpolation is an interpolation task that estimates intermediate frames between two frames. We have witnessed the tremendous advancement of video frame interpolation, but the general video prediction in the wild is still an open question. Inspired by the photo-realistic results of video frame interpolation, we present a new optimization framework for video prediction via video frame interpolation, in which we solve an extrapolation problem based on an interpolation model. Our video prediction framework is based on optimization with a pretrained differentiable video frame interpolation module without the need for a training dataset, and thus there is no domain gap issue between training and test data. Also, our approach does not need any additional information such as semantic or instance maps, which makes our framework applicable to any video. Extensive experiments on the Cityscapes, KITTI, DAVIS, Middlebury, and Vimeo90K datasets show that our video prediction results are robust in general scenarios, and our approach outperforms other video prediction methods that require a large amount of training data or extra semantic information.

1. Introduction

Video prediction is an extrapolation task to predict future video frames given some past frames. Video prediction has broad applications including robotics planning, autonomous driving and video manipulations [6,23,39,41]. For instance, predicted videos can help autonomous robots to better plan future actions with future visual information. Video prediction is also a fundamental task for unconditional video synthesis that can be decomposed into image synthesis and future video prediction.

Video prediction is a challenging extrapolation problem.

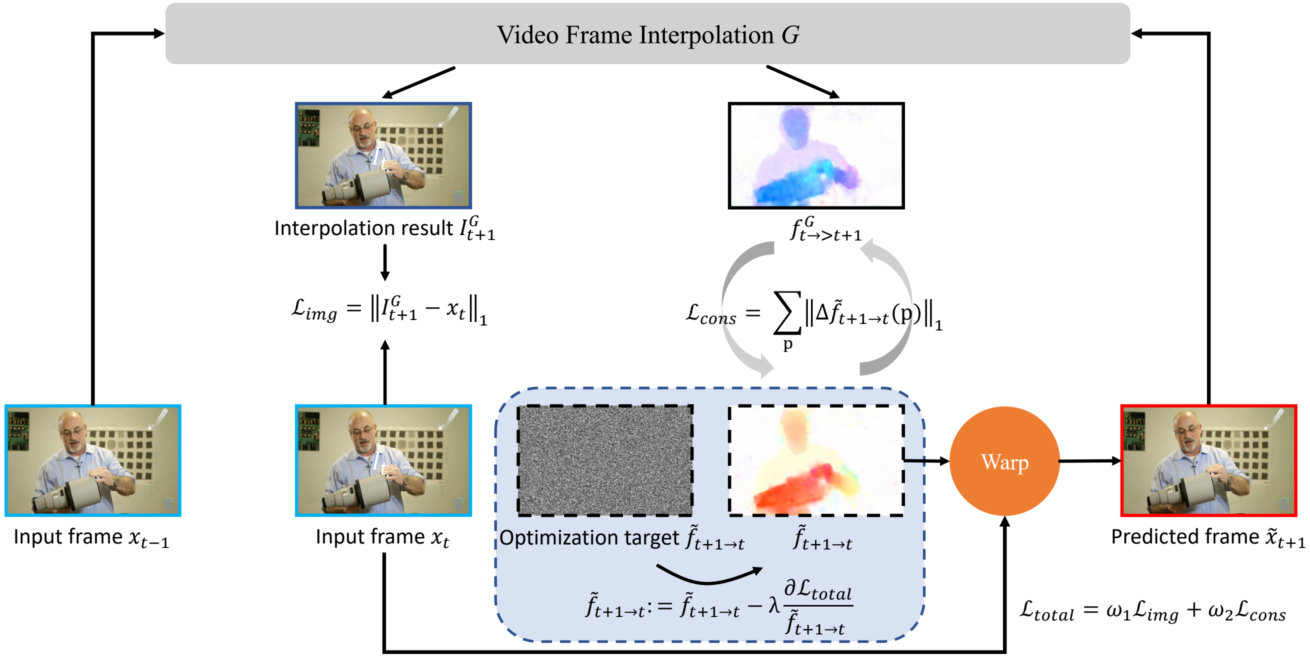

Video Frame Interpolation G
Interpolation result I^G_{t+1}
$f^G_{t->t+1}$
$\mathcal{L}_{img} = \|I^G_{t+1} - x_t\|_1$
$\mathcal{L}_{cons} = \sum_p \|\Delta \tilde{f}_{t+1\to t}(p)\|_1$
Input frame x_{t-1}
Input frame x_t
Optimization target $\tilde{f}_{t+1\to t}$
$\tilde{f}_{t+1\to t}$
Warp
Predicted frame $\tilde{x}_{t+1}$
$\tilde{f}_{t+1\to t} := \tilde{f}_{t+1\to t} - \lambda \frac{\partial \mathcal{L}_{total}}{\tilde{f}_{t+1\to t}}$
$\mathcal{L}_{total} = \omega_1 \mathcal{L}_{img} + \omega_2 \mathcal{L}_{cons}$

Figure 2. Overview of our method. We optimize optical flow $\tilde{f}_{t+1\to t}$ by a video frame interpolation G [11]. Our optimization objective is image-level distance $\mathcal{L}_{img}$ and a consistency constraint between our predicted flow $\tilde{f}_{t+1\to t}$ and the flow $f^G_{t\to t+1}$ generated by G.

Several studies [18, 20, 22, 42] only take RGB frames as input for video prediction and find that the video prediction problem is difficult to solve, because of the inherent high complexity of video prediction and the uncertainty of future states. Thus, recently, a lot of constraints and assumptions about the modeling scene have been employed to simplify the problem. However, these assumptions reduce the generalization ability of these video prediction models. FVS [45] and Lee *et al.* [15] require semantic maps to decompose the scene. Bei *et al.* [4] first predict semantic maps and then synthesize future frames. Qi *et al.* [35] require depth maps to reconstruct 3D point clouds. However, such additional information is often hard to obtain or estimate correctly in general scenarios. These strong assumptions limit these methods to be only applicable to data when these assumptions hold. For example, failing to detect some objects (including unseen objects) will lead to performance degradation. When this extra information is unavailable or of poor quality, these methods suffer from performance degradation and cannot be even applied in diverse videos in the real world.

Moreover, these external methods usually need to train one model for each specific scenario, making them difficult generalize to other scenarios. For example, it is hard to apply a video prediction model trained on a driving scene to a human moving dataset, as the motion difference is huge.

To address these issues, we propose an optimization-based video prediction method, without the requirement of external training (no external dataset is needed for training), and can produce state-of-the-art results. Our insight is that we can cast the video prediction problem as a video frame interpolation (VFI) based optimization problem. Inspired by the recent success of VFI, we connect these two problems to solve the video prediction problem in a new way. Our method does not require any assumption such as semantic segmentation and can be applied to any video. We evaluate our method on multiple datasets, and our method can outperform the state of the arts. Our contributions can be summarized as follows:

- We present the first optimization framework for video prediction. We cast the extrapolation problem of video prediction as an optimization problem with VFI.
- Our framework is highly flexible as it does not require any semantic or instance maps, prior knowledge about the scene, or external training. Our method is applicable to video prediction in any scene at any resolution.
- Our method obtains outstanding performance on various datasets and outperforms state-of-the-art video prediction approaches that require additional information. Our method surpasses external learning methods taking only RGB frames as input by a large margin.

2. Related work

2.1. Video Prediction

Early works, taking RGB frames as input, employ several mechanisms to improve video prediction. MCNet [42] decomposes a scene into content and motion components. DVF [20] proposes deep voxel flow to synthesize future frames. However, video prediction in the wild is still quite hard due to its inherent high complexity and uncertainty. Thus, explicit modeling, constraints, and assumptions about the scene are introduced. Qi *et al*. [35] utilize depth maps to reconstruct 3D point clouds. Gao *et al*. [7] use semantic maps to enforce layout consistency. FVS [45] needs semantic segmentation and instance segmentation to decompose a scene into background and foreground identities. Bei *et al*. [4] and Lee *et al*. [15] also require semantic maps.

Although the performance of video prediction has been gradually improved, the generalization ability of these approaches is arguably reduced. It is hard to apply these methods to data without these extra annotations. Moreover, these methods may suffer from performance degradation when the test data is from a different domain than the train data. For example, it is hard to apply a model trained on robot scenarios to driving scenes because the motion in these two domains is quite different. Thus, we propose an optimization-based video prediction method, which can produce state-of-the-art results without external training (thus no domain gap).

2.2. Video Frame Interpolation

Contrary to the challenges of video prediction problems, VFI has gained great success recently, which motivates our model design. VFI aims to interpolate intermediate frames between successive input frames. There are three categories of algorithms: kernel-based [2,3,29,30], phase-based [24, 25], and motion-based [2, 3, 10, 13, 19, 21, 27, 28, 32, 33]. These motion-based methods use bi-directional optical flows to warp two consecutive frames forward and backward to obtain an intermediate frame. Our method adopts a motion-based framework. Super SloMo [13] linearly combines bi-directional flows as an initial approximation of the intermediate flows for further refinement. Park *et al*. [33] conduct asymmetric bilateral motion estimation. Sim *et al*. [37] first handle the VFI for 4K videos with large motion. RIFE [11] is a real-time interpolation algorithm that estimates the intermediate flows in a coarse-to-fine fashion. Different from the domain gap problem in video prediction, there are much fewer domain constraints in VFI. These methods do not require additional information, such as semantic maps and depth maps, and can produce outstanding interpolation performance even with complex motion. Inspired by the success of VFI, we cast video prediction as an optimization problem based on VFI.

2.3. Optimization-based Methods

Since learning-based methods suffer from the domain gap between training data and test data, optimization-based methods on test data are still competitive nowadays. Gatys *et al*. [8] propose the first optimization-based method for neural style transfer. Shaham *et al*. [36] optimize a generative model on a single image and can generate high quality and diverse samples from the image. Lei *et al*. [16, 17] optimize a network to improve the temporal consistency. Mildenhall *et al*. [26] optimize a fully-connected network on a sparse set of input views from a scene for novel view synthesis. These optimization-based methods inspire us to propose a framework that not only tackles the domain gap problem but also provides an on-the-fly control for users during optimization.

3. Method

3.1. Problem Formulation

Let x_t be the video frame at time step t. The input to our framework includes two recent RGB frames x_{t-1} and x_t. Our goal is to predict the future frames $\{\tilde{x}_{t+1}, \tilde{x}_{t+2}, \ldots\}$. We adopt a pretrained video interpolation network [11] denoted as G. We will focus on predicting the next frame $\tilde{x}_{t+1}$ first as we can predict future frames one by one sequentially. During the optimization process, the parameters of G are unchanged. Our primary objective is

$$\tilde{x}^*_{t+1} = \underset{\tilde{x}_{t+1}}{\operatorname{argmin}}\, E(G(x_{t-1}, \tilde{x}_{t+1}), x_t), \tag{1}$$

where E is an objective function that measures image similarity. Here we utilize a VFI network G to constrain the relationship among x_{t-1}, $\tilde{x}_{t+1}$, and x_t.

To ease the optimization process, we choose to optimize optical flow $\tilde{f}_{t+1\to t}$ between the predicted frame $\tilde{x}_{t+1}$ and the last observed frame x_t instead of directly optimize $\tilde{x}_{t+1}$. $\tilde{x}_{t+1}$ is computed using backward warping [12]:

$$\tilde{x}_{t+1} = warp(x_t, \tilde{f}_{t+1\to t}). \tag{2}$$

Then Eq. 1 can be rewritten as

$$\tilde{f}^*_{t+1\to t} = \underset{\tilde{f}_{t+1\to t}}{\operatorname{argmin}}\, E(G(x_{t-1}, warp(x_t, \tilde{f}_{t+1\to t})), x_t). \tag{3}$$

Flow initialization. To ease the optimization of Eq. 3, it is a good practice to start with a flow $\tilde{f}_{t+1\to t}$ that produces an approximate motion. Therefore, we initialize it utilizing the negative flow of $f_{t\to t-1}$:

$$\tilde{f}_{t+1\to t} = \delta(-f_{t\to t-1}). \tag{4}$$

We first compute $-f_{t\to t-1}$ as a rough approximation of $f_{t->t+1}$. Then we initialize $\tilde{f}_{t+1\to t}$ as the inversion of $-f_{t\to t-1}$. δ represents the operation similar to the flow reversal layer [46] to convert a forward flow to a backward flow (details in the supplement).

However, directly optimizing Eq. 3 is still difficult, because the constraint towards $\tilde{f}_{t+1\to t}$ is indirect, and the optimization process is difficult to converge.

3.2. Video Frame Interpolation Network

Thus, we propose to utilize the intermediate results of network G. Given x_{t-1} and $\tilde{x}_{t+1}$ as input, G generates optical flows of two directions $f^G_{t\to t-1}$, $f^G_{t\to t+1}$, and a mask m^G. The superscript G denotes that it is the output of network G. The video interpolation network warps x_{t-1} and $\tilde{x}_{t+1}$ towards time step t:

$$\begin{aligned} I^G_{t-1} &= warp(x_{t-1}, f^G_{t\to t-1}), && (5)\\ I^G_{t+1} &= warp(\tilde{x}_{t+1}, f^G_{t\to t+1}), && (6)\\ I^G_t &= I^G_{t-1} \times m^G + I^G_{t+1} \times (1 - m^G), && (7) \end{aligned}$$

where I^G_{t-1} is the intermediate interpolation frame by warping x_{t-1} using $f^G_{t\to t-1}$. I^G_{t+1} is the intermediate interpolation frame by warping $\tilde{x}_{t+1}$ using $f^G_{t\to t+1}$. The final interpolation result is a weighted sum of I^G_{t-1} and I^G_{t-1}.

We utilize I^G_{t+1} instead of I^G_t since I^G_{t+1} has a closer relation with $\tilde{x}_{t+1}$ and can dismiss the effect of m^G. We employ a L_1 distance between I^G_{t+1} and x_t:

$$\mathcal{L}_{img} = \left\| I^G_{t+1} - x_t \right\|_1 . \qquad (8)$$

We also argue that there is a forward-backward consistency relationship between $f^G_{t\to t+1}$ and $\tilde{f}_{t+1\to t}$. This constraint means that after the forward and backward propagation, the pixels should go back to the original locations:

$$\mathcal{L}_{cons} = \sum_{\mathbf{p}} \left\| \Delta \tilde{f}_{t+1\to t}(\mathbf{p}) \right\|_1 , \qquad (9)$$

where $\Delta \tilde{f}_{t+1\to t}(\mathbf{p})$ is the discrepancy obtained from forward and backward flow check at pixel location $\mathbf{p}$:

$$\begin{aligned} \Delta \tilde{f}_{t+1\to t}(\mathbf{p}) &= \mathbf{p} - \left(\mathbf{p}' + f^G_{t\to t+1}(\mathbf{p}')\right), && (10)\\ \mathbf{p}' &= \mathbf{p} + \tilde{f}_{t+1\to t}(\mathbf{p}). && (11) \end{aligned}$$

Our overall objective function is $\mathcal{L}_{total} = \omega_1 \mathcal{L}_{img} + \omega_2 \mathcal{L}_{cons}$, where ω_1 and ω_2 are the loss weights.

3.3. Flow Inpainting

There always exist occlusion areas in optical flow: some pixels do not have corresponding pixels in successive frames. The estimated optical flow in occlusion areas is unreliable. Thus, we set a threshold α to mask out these areas, and using flow inpainting to fill in the holes:

$$\phi(\mathbf{p}) = \begin{cases} 1 & if \quad \left\| \Delta \tilde{f}_{t+1\to t}(\mathbf{p}) \right\|_1 > \alpha, \\ 0 & \quad otherwise. \end{cases} \qquad (12)$$

If $\phi(\mathbf{p})$ is 1, its optical flow is treated unreliable, and is inpainted by a linear combination of neighboring valid flow values, whose weights are inversely proportional to the distance between the invalid pixel and the valid pixels.

	External learning methods			
	External training	Semantic	Instance	Depth
PredNet [22]	✓	×	×	×
MCNET [42]	✓	×	×	×
DVF [20]	✓	×	×	×
Vid2vid [43]	✓	✓	×	×
Qi *et al.* [35]	✓	✓	×	✓
Seg2vid [31]	✓	✓	×	×
FVS [45]	✓	✓	✓	×
HVP [15]	✓	✓	×	×
SADM [4]	✓	✓	×	×
	Optimization methods			
Ours	×	×	×	×

Table 1. Comparison with other video prediction methods in terms of methods' requirements.

We try to use adaptive weights like Softmax Splatting [28] to resolve the occlusions and find it unsuitable for our framework. Detailed analysis is described in the supplement.

3.4. Implementation

Multi-frame prediction. For multi-frame prediction, we choose to optimize the next frame recurrently. We try to set the optimization target as multiple optical flows and optimize multiple future frames together. The result turns out recurrently optimizing the next frame is more stable.

The input frame length for all datasets is set to 2. The hyperparameters ω_1, ω_2, α are set to 1.0, 3.0, 1.5 empirically. We adopt RAFT [40] to estimate optical flow used for optimization target initialization. The Adam optimizer [14] is used with a learning rate of 0.1 for 3000 iterations for each future frame. We adopt the VFI method RIFE [11], which is pretrained only on Vimeo90K [47], for all the experiments. During the process of optimization, the network weight of RIFE [11] is fixed.

4. Experiments

We compare our method with state-of-the-art methods as shown in Table 1 in terms of methods' requirements. Due to the complexity of future video prediction, many recent methods add some extra assumptions for video prediction, such as semantic maps [4, 15, 31, 35, 43, 45], instance maps [45], and depth maps [35]. These assumptions may improve prediction performance but vastly decrease their generalization capability. Furthermore, these methods demand a training dataset to train a neural network. However, our method can avoid these restrictions by casting the video prediction problem as optimization. We do not require external training and do not have any assumptions about the data. Our method is very general and outperforms previous external RGB-based methods and methods using additional assumptions [31, 43, 45]. Statistical analysis for long-term prediction and more visual comparisons are provided in the supplement.

		Cityscapes						KITTI					
		MS-SSIM (×1e−2)↑			LPIPS (×1e−2)↓			MS-SSIM (×1e−2)↑			LPIPS (×1e−2)↓		
	Input	t+1	t+3	t+5	t+1	t+3	t+5	t+1	t+3	t+5	t+1	t+3	t+5
External learning methods													
PredNet [22]	RGB	84.03	79.25	75.21	25.99	29.99	36.03	56.26	51.47	47.56	55.35	58.66	62.95
MCNET [42]	RGB	89.69	78.07	70.58	18.88	31.34	37.34	75.35	63.52	55.48	24.05	31.71	37.39
DVF [20]	RGB	83.85	76.23	71.11	17.37	24.05	28.79	53.93	46.99	42.62	32.47	37.43	41.59
Vid2vid [43]	RGB+S.	88.16	80.55	75.13	10.58	15.92	20.14	N/A	N/A	N/A	N/A	N/A	N/A
Seg2vid [31]	RGB+S.	88.32	N/A	61.63	9.69	N/A	25.99	N/A	N/A	N/A	N/A	N/A	N/A
FVS [45]	RGB+S.+I.	89.10	81.13	75.68	8.50	12.98	**16.50**	79.28	67.65	60.77	18.48	24.61	30.49
Optimization methods													
Ours	No external training	**94.54**	**86.89**	**80.40**	**6.46**	**12.50**	17.83	**82.71**	**69.50**	**61.09**	**12.34**	**20.29**	**26.35**

Table 2. Comparison with state-of-the-art methods on the Cityscapes and KITTI datasets. S. and I. denote that the method requires semantic maps or instance maps as input. Our method can outperform previous video prediction methods by a large margin.

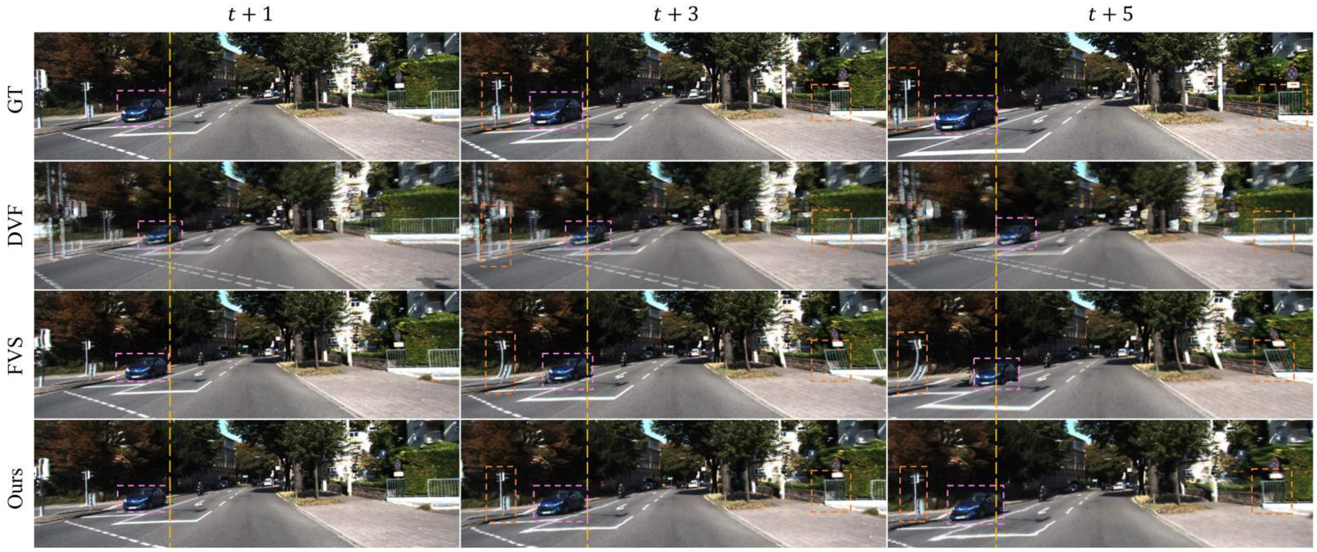


Figure 3. Multi-frame prediction comparison on KITTI. As the pink boxes show, DVF [20] fails to separate the motion of the blue car from the background and produces a "zooming-in" effect, which is the major motion exhibited in the dataset, caused by the forward movement of the running car. FVS [45] wrongly predicts the motion of the blue car at $t+3$ and $t+5$, resulting in incorrect frame prediction results. Our model can correctly capture the motion of the blue car without external training.

4.1. Evaluation on Driving Datasets

We first evaluate our approach and relevant baselines on driving datasets, where semantic information is available, because some baselines require additional semantic maps.

Datasets. Cityscapes [5] and KITTI datasets [9] contain driving sequences. Our evaluation setting follows [45].

Baselines. All baselines are trained on the corresponding training set in Cityscapes and KITTI. We categorize baselines into two types. One type is the methods that take only RGB frames as input, such as PredNet [22], MCNet [42], and DVF [20]. Another type is the methods that require some additional information, such as semantic maps, instance maps, including Vid2vid [43], Seg2vid [31], and FVS [45]. However, these assumptions restrict these methods to be only applicable when this additional context is accessible, degrading their potential generalization ability. We use Multi-scale Structural Similarity Index Measure (MS-SSIM) [44] and LPIPS [48] as evaluation metrics. Higher MS-SSIM and lower LPIPS indicate better performance.

Quantitative results. Although our method does not use the training set in Cityscapes and KITTI, our method can still produce outstanding results. As shown in Table 2, our method can outperform RGB-based methods by a large margin in both short-term and long-term video prediction. Our method improves DVF [20] by 12.75%, 13.98%, 13.06% in MS-SSIM, 62.81%, 48.02%, 38.07% in of LPIPS on the $t+1$, $t+3$, $t+5$ predictions. Moreover, compared with the methods utilizing semantic or instance segmentation, our method can still outperform FVS [45] by 6.11%, 7.09%, 6.24% in MS-SSIM on Cityscapes, and

	DAVIS				Middlebury				Vimeo90K	
	MS-SSIM (×1e−2)↑		LPIPS (×1e−2)↓		MS-SSIM (×1e−2)↑		LPIPS (×1e−2)↓		MS-SSIM (×1e−2)↑	LPIPS (×1e−2)↓
	t+1	t+3	t+1	t+3	t+1	t+3	t+1	t+3	t+1	t+1
External learning methods										
DVF [20]	68.61	55.47	23.23	34.22	83.98	65.54	13.57	25.70	92.11	7.73
DYAN [18]	78.96	70.41	13.09	21.43	92.96	83.91	7.98	15.03	N/A	N/A
Optimization methods										
Ours	**83.26**	**73.85**	**11.40**	**18.21**	**94.49**	**87.96**	**6.07**	**10.82**	**96.75**	**3.59**

Table 3. Evaluation on diverse datasets. Comparison with state-of-the-art methods on DAVIS, Middlebury, and Vimeo90K.

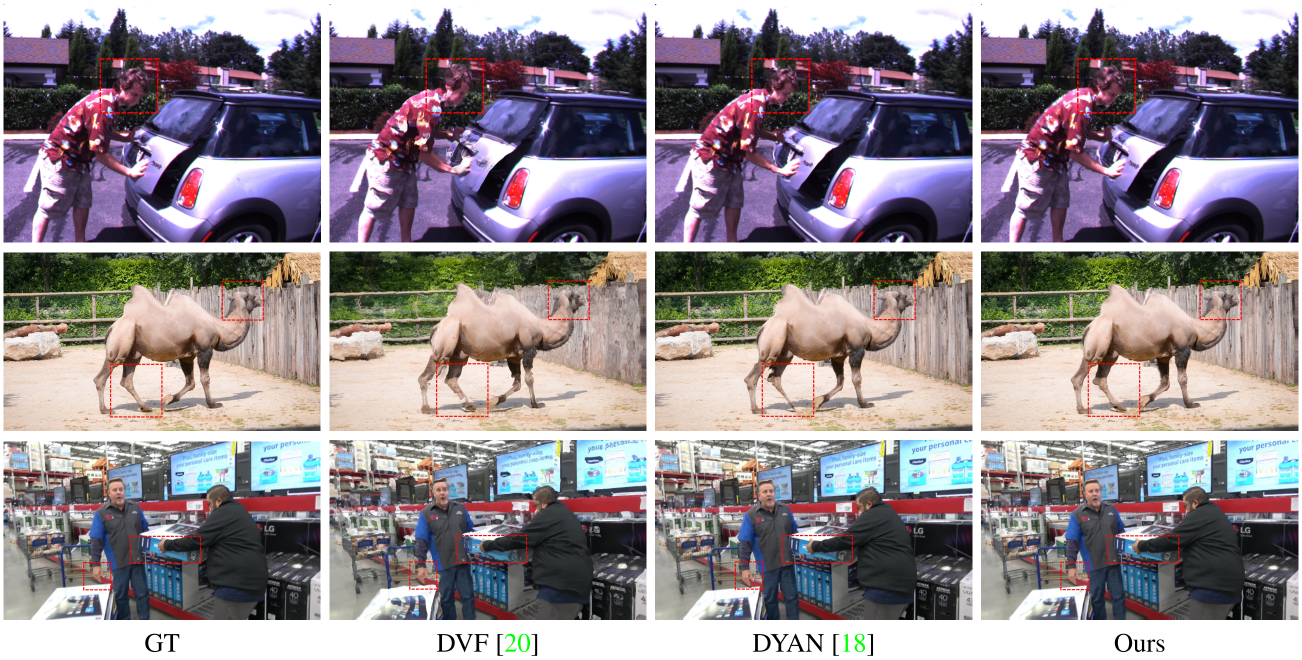
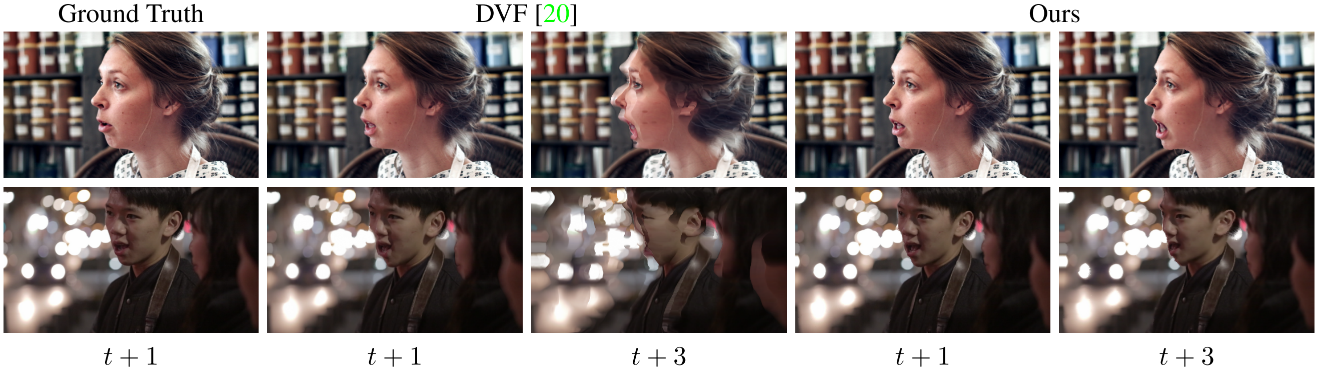

GT DVF [20] DYAN [18] Ours

Figure 4. Visual comparison on DAVIS and Middlebury datasets.


Ground Truth DVF [20] Ours

$t+1$ $t+1$ $t+3$ $t+1$ $t+3$

Figure 5. Multi-frame prediction on Vimeo90K. Since Vimeo90K contains triplets and we use two frames as input, there is no ground-truth corresponding to $t+3$.

33.21%, 17.55%, 13.58% in LPIPS on KITTI, for the $t+1$, $t+3$, $t+5$ predictions.

The quantitative results demonstrate that our method without external training on the training set of Cityscapes and KITTI, can achieve better performance in both short-term and long-term video prediction. This is because our method is based on the powerful constraints from VFI: recent VFI models can produce superior interpolation results that can be arguably treated as ground truth. While other methods use handcrafted loss functions, these methods may

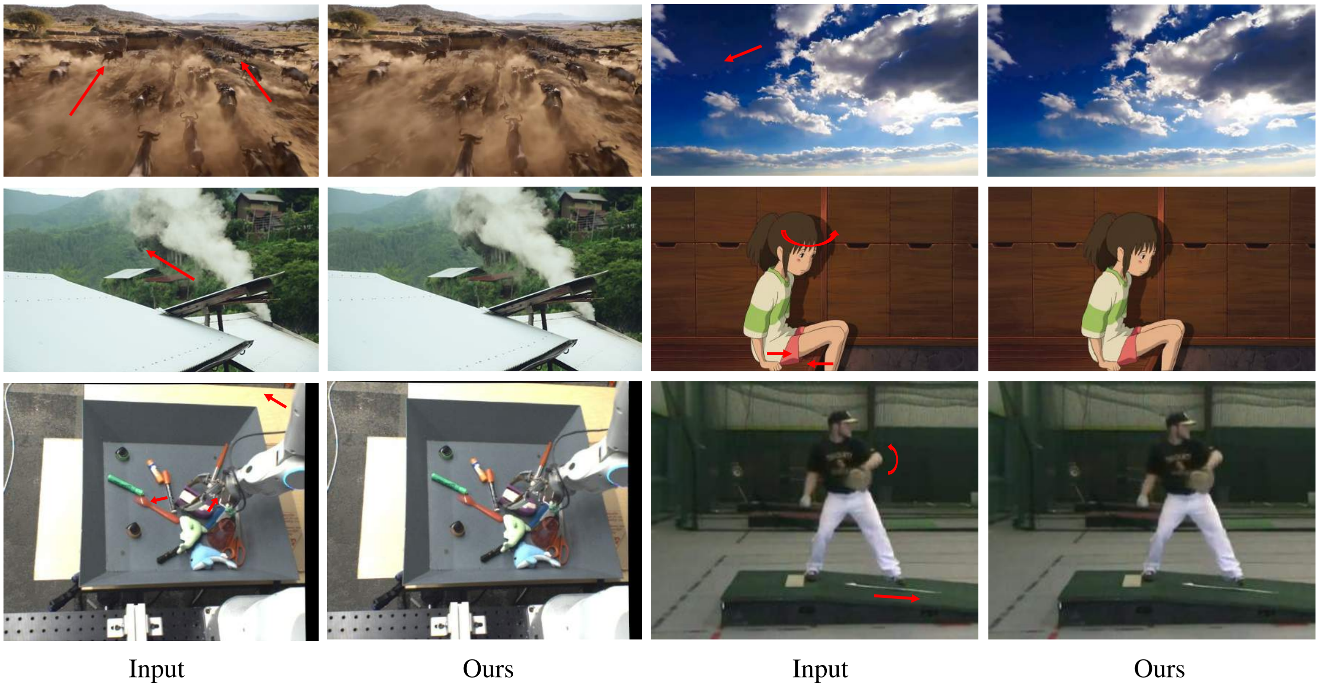


Figure 6. The next frame prediction performance on diverse data. The left image is the last observed frame, and we use red arrows to indicate the motion from the input frames. The right image is the predicted next frame by our method.

overfit to the major motion exhibited in the training set, the "zooming-in" effect, which is caused by the moving of the forwarding car, instead of learning the real motion.

Qualitative Results. In Fig. 3, our method is compared to recent video prediction methods, DVF [20] (RGB) and FVS [45] (RGB + semantic + instance). DVF [20] tends to be dominated by "zooming-in" motion without predicting the true motion, since the driving datasets are captured by a forward moving camera. FVS [45] uses a handcrafted 2D affine transformation to approximate the motion of moving cars. However, complex motion, including nonrigid deformation and 3D rotation, can not be captured by 2D affine transformation. Moreover, DVF and FVS rely on semantic and instance segmentation, and their performance degrades when these assumptions do not hold. Visual comparisons on Cityscapes are presented in the supplement.

4.2. Evaluation on Diverse Datasets

Since our optimization framework does not need external training, our method can be generalized to any video at any resolution. Meanwhile, previous external methods trained on dataset A may have performance degradation when applied to dataset B, caused by the domain gap between A and B. To demonstrate the generality of our method, we conduct a cross-dataset evaluation on diverse datasets.

Datasets. We evaluate our methods on multiple datasets, including DAVIS [34], Middlebury-Other [1], and Vimeo90K [47] datasets. **DAVIS [34]:** there are 30 sequences in the validation set with resolutions around 854 × 480 . **Middlebury [1]:** there are 10 videos with resolutions around 640 × 480. **Vimeo90K [47]:** there are 3782 triplets in the test set with a resolution of 448 × 256. By taking one clip every ten clips, we form the test set.

Components	SSIM↑	PSNR↑	LPIPS↓
Zero	0.7719	23.79	0.1230
Noise	0.7669	23.64	0.1228
Without $\mathcal{L}_{img}$	0.8939	28.86	0.0660
Without $\mathcal{L}_{cons}$	0.8732	28.43	0.1221
$\mathcal{L}_{img}$ with MSE	0.8877	28.97	0.1232
With $\mathcal{L}_{interp}$	0.8963	28.87	0.0693
Long-term $\mathcal{L}_{img}$	0.6978	21.75	0.1657
W/o flow inpainting	0.8882	28.94	0.1139
Full Model	0.8975	29.10	0.0646

Table 4. The ablation study.

Baselines. We compare our method with two latest external methods, DVF [20] and DYAN [18], which take only RGB frames as input, since they can be applied to these datasets. Other methods that require additional assumptions cannot be compared because their assumptions do not hold. We test these two models on these datasets using their pretrained model on UCF101 [38].

Quantitative results. As exhibited in Table 3, our method outperforms DYAN [18] by 12.90%, 15.0% in DAVIS, and by 23.97%, 28.00% on Middlebury in LPIPS for $t + 1$, $t + 3$ predictions. Note that our method is still robust in long-term prediction. On the Vimeo90K dataset, our method outperforms DVF [20] by 53.56% in LPIPS for next-frame prediction. Although these two baselines may perform well on UCF-101, there is a domain gap problem between UCF-101 and these test videos. This domain gap phenomenon commonly exists in video prediction tasks, so most video prediction methods [4, 18, 20, 22, 31, 42, 45] usually train a separate model for each dataset and even employ different assumptions for each dataset. Differently, the domain gap in the VFI task does not appear to be an issue. VFI methods [11, 33] can use one dataset for training and can produce excellent results on various datasets. With the powerful constraints provided by VFI, our method does not have the domain gap problem (no external training): each sequence is independently optimized by employing the FVI network as a constraint.

Qualitative results. As shown in Fig. 4 and Fig. 5, our method yields better prediction results than baselines. The baselines produce distortion artifacts around object boundary or fail at prediction when motion is complicated. Meanwhile, our method can robustly predict future frames. As the Vimeo90K dataset only provides three frames, we take the first two frames as input for future frame prediction (there is only ground truth for the $t + 1$ prediction). Our method can produce high-quality prediction results in the long term, as shown in Fig. 5. The intricate details, such as the hair of the woman, remain clear in our results.

To demonstrate the generality of our method, we collect some real videos from YouTube, such as the movie clips as shown in Fig. 6. We also present some visual results on the BAIR Pushing [6] and Penn Action [49] datasets. Our method can produce strong results on diverse data.

4.3. Ablation Study

We conduct an ablation study to demonstrate the importance of each component, as shown in Table 4. Our ablation study is conducted on Cityscapes. We use the performance of the next-frame prediction to evaluate different components. Visual comparisons are presented in the supplement.

Initialization. If we set the optimization target as the prediction frame, then the optimization cannot converge. Thus, we choose to optimize optical flow, which eases the optimization process since pixels can be copied from observed frames. If we initialize optical flow as zeros or Gaussian noise, then our performance becomes worse. If we initialize the optical flow as the copy of $f_{t\rightarrow t-1}$, then the performance is similar to our full model. This demonstrates that good initialization helps the optimization to converge.

Loss functions. We conduct experiments with loss functions with several variants. Without $\mathcal{L}_{img}$: supervised only by $\mathcal{L}_{cons}$. Without $\mathcal{L}_{cons}$: supervised only by $\mathcal{L}_{img}$. $\mathcal{L}_{img}$ with MSE: using MSE loss rather than L_1 loss in Eq. 8. With $\mathcal{L}_{interp}$: use the interpolation result of VFI I_t^G rather than the intermediate output I_{t+1}^G in Eq. 8. Long-term $\mathcal{L}_{img}$: setting the input frame length as 4 and add a long-term constraint between x_{t-3}, $\tilde{x}_{t+1}$ and x_{t-1}. The results show that the combination of $\mathcal{L}_{img}$ and $\mathcal{L}_{cons}$ performs best. $\mathcal{L}_{img}$ provides a more direct constraint than $\mathcal{L}_{interp}$. Long-term $\mathcal{L}_{img}$ has performance degradation because when the motion is too large, the accuracy of VFI decreases .

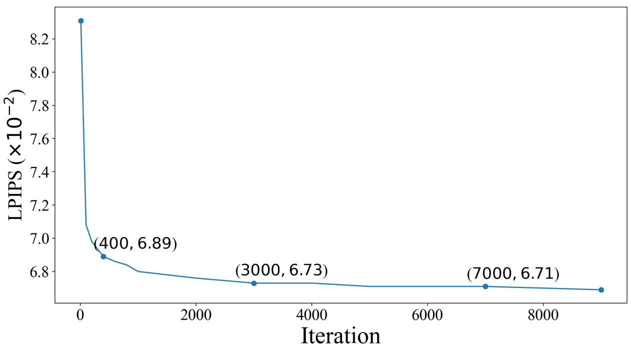


Figure 7. Our model with different optimization iterations.

Flow inpainting. If we remove the optical flow inpainting procedure, then our performance also drops because optical flow inpainting effectively corrects invalid flow values.

Other pretrained VFI models. We also try utilizing Super SloMo [13] as our VFI backbone and find this method also works for our framework. For the next frame prediction on Cityscapes, the MS-SSIM for Super SloMo [13] and RIFE [11] is 0.9199 and 0.9454. Thus, we choose RIFE [11] as our VFI backbone.

4.4. Convergence Analysis

As shown in Fig. 7, we conduct the convergence analysis on Cityscapes with a resolution of 256 × 512 . In the first 400 iterations, the optimization converges fast. After 400 iterations, the prediction result gradually improves. The prediction result is still slowly improving after 3000 iterations.

5. Conclusion

We propose the first video prediction optimization method by casting the video prediction problem as a VFI based optimization problem, which addresses the domain gap issue in most video prediction methods. Our method can outperform state-of-the-art methods and can be adapted to any video at any resolution. Although our method relieves domain gap problem and presents impressive performance, optimizing every frame by our method costs more time than other external learning-based methods. The comparison between our and other methods in terms of the model size and inference time is presented in the supplement. As we observe, most run time in our model is spent on the gradient propagation inside the VFI network [11], which inspires us to design a more efficient backbone for acceleration in the future.

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. 7

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, 2019. 3

[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. *TPAMI*, 43(3):933–948, Mar. 2021. 3

[4] Xinzhu Bei, Yanchao Yang, and Stefano Soatto. Learning semantic-aware dynamics for video prediction. In *CVPR*, 2021. 2, 3, 4, 8

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5

[6] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016. 1, 8

[7] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *ICCV*, 2019. 3

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 3

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *I. J. Robotics Res.*, 2013. 5

[10] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. FeatureFlow: Robust video interpolation via structure-to-texture generation. In *CVPR*, 2020. 3

[11] 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. 2, 3, 4, 8

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NIPS*, 2015. 3

[13] 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. 3, 8

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[15] Wonkwang Lee, Whie Jung, Han Zhang, Ting Chen, Jing Yu Koh, Thomas E. Huang, Hyungsuk Yoon, Honglak Lee, and Seunghoon Hong. Revisiting hierarchical approach for persistent long-term video prediction. In *ICLR*, 2021. 2, 3, 4

[16] Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. In *NeurIPS*, 2020. 3

[17] Chenyang Lei, Yazhou Xing, Hao Ouyang, and Qifeng Chen. Deep video prior for video consistency and propagation. *TPAMI, year = To Appear*. 3

[18] Wenqian Liu, Abhishek Sharma, Octavia Camps, and Mario Sznaier. Dyan: A dynamical atoms-based network for video prediction. In *ECCV*, 2018. 2, 6, 7, 8

[19] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *AAAI*, 2019. 3

[20] Ziwei Liu, Raymond Yeh, Yiming Liu Xiaoou Tang, , and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 2, 3, 4, 5, 6, 7, 8

[21] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 3

[22] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017. 2, 4, 5, 8

[23] Pauline Luc, Natalia Neverova, Camille Couprie, Jacob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017. 1

[24] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. PhaseNet for video frame interpolation. In *CVPR*, 2018. 3

[25] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, 2015. 3

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 3

[27] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, 2018. 3

[28] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, 2020. 3, 4

[29] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017. 3

[30] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 3

[31] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. In *CVPR*, 2019. 4, 5, 8

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

[33] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *ICCV*, 2021. 3, 8

[34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, abs/1704.00675, 2017. 7

[35] Xiaojuan Qi, Zhengzhe Liu, Qifeng Chen, and Jiaya Jia. 3d motion decomposition for RGBD future dynamic scene synthesis. In *CVPR*, 2019. 2, 3, 4

[36] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *ICCV*, 2019. 3

[37] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: Extreme video frame interpolation. In *ICCV*, 2021. 3

[38] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 7

[39] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 1

[40] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020. 4

[41] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V. Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *NeurIPS*, 2018. 1

[42] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017. 2, 4, 5, 8

[43] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 4, 5

[44] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. IEEE, 2003. 5

[45] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *CVPR*, 2020. 2, 3, 4, 5, 7, 8

[46] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, 2019. 3

[47] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 4, 7

[48] 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. 5

[49] Weiyu Zhang, Menglong Zhu, and Konstantinos Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 8