Stochastic Dynamics for Video Infilling

Qiangeng Xu[1] Hanwang Zhang[2] Weiyue Wang[1] Peter N. Belhumeur[3] Ulrich Neumann[1]
[1]University of Southern California [2]Nanyang Technological University [3]Columbia University
{qiangenx,weiyuewa,uneumann}@usc.edu hanwangzhang@ntu.edu.sg belhumeur@cs.columbia.edu

Abstract

In this paper, we introduce a stochastic dynamics video infilling (SDVI) framework to generate frames between long intervals in a video. Our task differs from video interpolation which aims to produce transitional frames for a short interval between every two frames and increase the temporal resolution. Our task, namely video infilling, however, aims to infill long intervals with plausible frame sequences. Our framework models the infilling as a constrained stochastic generation process and sequentially samples dynamics from the inferred distribution. SDVI consists of two parts: (1) a bi-directional constraint propagation module to guarantee the spatial-temporal coherence among frames, (2) a stochastic sampling process to generate dynamics from the inferred distributions. Experimental results show that SDVI can generate clear frame sequences with varying contents. Moreover, motions in the generated sequence are realistic and able to transfer smoothly from the given start frame to the terminal frame.

1. Introduction

Video temporal enhancement is generally achieved by synthesizing frames between every two consecutive frames in a video. Recently, most studies [21, 15] focus on interpolating videos with frame rate above 20 fps. The between-frame intervals of these videos are short-term and the consecutive frames only have limited variations. Instead, our study focuses on the **long-term interval infilling** for videos with frame rate under 2 fps. This study can be applied on recovering low frame rate videos recorded by any camera with limited memory, storage, network bandwidth or low power supply (e.g., outdoor surveillance devices and webcam with an unstable network).

The difference between video interpolation and video infilling is shown in Figure 1. Conditioned on frame 7 and 8, video interpolation generates transitional frames containing similar content for short intervals. However, video infilling generates frames in a long-term interval (from frame 8 to 12) and requires the model to produce varying content. At each timestamp, the model needs to sample a plausible dynamic sequence out of many possible movements.

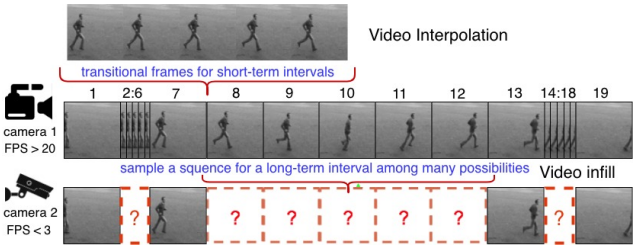

Figure 1: Difference between video interpolation and video infilling. Camera 1 captures frames 1 to 19. Video interpolation aims to generate 5 frames between frame 7 and 8. A low frame rate camera 2 only captures frame 1, 7, 13 and 19. Video infilling focuses on generating a plausible intermediate dynamic sequence for camera 2 (a plausible sequence can be different from the frames 8 to 12).

Figure 2 illustrates the stochastic nature of the long-term intermediate sequence. We observe the following two phenomena: (1) Compared with Scenario 1, since both the interval length and the difference between the two reference frames are larger, the uncertainties in the long-term interval (Scenario 2) are greater. (2) Taken frame 5 and 9 as references, both the red and the green motions between frame 5 and 9 are plausible. If we also add frame 1 and 13 as references, only the green motion is plausible. Consequently, utilizing long-term information (frame 1 and 13 in Figure 2) can benefit the dynamics inference and eliminate the uncertainties. Given start and end frames of a long-term interval in a video, we introduce stochastic dynamic video infilling (SDVI) framework to generate intermediate frames which contain varying content and transform smoothly from the start frame to the end frame.

1.1. Task Formulation

Following the standard input setting of temporal super-resolution, we formulate our task as follows: For a sequence $\mathbf{X}$, only one out of every u frames ($u = T - S$) is captured. The goal of SDVI is to infill a sequence $\tilde{X}_{S+1:T-1}$ between reference frames X_S and X_T. In Figure 1, X_S and X_T are

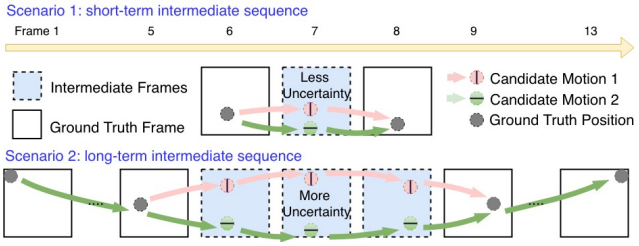

Figure 2: The difference of the randomness between short-term and long-term intervals: The camera in scenario 1 can capture every other frame and the camera in scenario 2 captures 1 frame for every 4 frames. The red and the green trajectories indicate two possible motions in each scenario.

frame 7 and 13. We also use additional frames (frame 1 and 19) as extended references. X_S, X_T and extended reference frames (here we choose i=1) form the reference set "window of reference" $\mathbf{X_{WR}}$.

$$\underbrace{X_{S-i\times u}, ..., X_{S-u}}_{\text{extended reference frames}}, X_S, \underbrace{X_{S+1:T-1}}_{\text{our target}}, X_T, \underbrace{X_{T+u}, ..., X_{T+i\times u}}_{\text{extended reference frames}}$$

Different from all existing methods, SDVI inference $P(X_{S+1:T-1}|\mathbf{X_{WR}})$ instead of $P(X_{S+1:T-1}|X_S, X_T)$.

1.2. Model Overviews

Most video interpolation methods [23, 18] rely on estimating the short-term pixel movements. Our task is also related to video prediction. Video prediction models [34, 20, 7] can generate long-term sequences by explicit dynamics modeling, but do not take discontinuous inputs and are not optimized for bi-directional constraints. On the contrary, our model explicitly inference the motion dynamics of the interval and propagate constraints from both sides (start and end frames).

Different from both video interpolation and video prediction, the task has three major challenges:

1. The inputs of the video prediction are consecutive frames so the initial momentum is given. However, the inputs of video infilling are sparse and discontinuous (X_S and X_T), which makes the task more challenging.

2. The observation of the last frame becomes a long-term coherence requirement, which gives more constraints to our model. Video prediction only needs to generate visually plausible frames smoothly transferred from previous frames, while video infilling is also required to guarantee the coherence between the previous sequence ($\tilde{X}_{S+1:T-1}$) and the terminal frame X_T.

3. As illustrated in Figure 2, compared with interpolation tasks, an interval in video infilling has more uncertainties, even with more reference frames (frame 1 and 13).

To inference the initial and final momentum, we expose extended reference frames both from the past (frame 1 and 7 in Figure 1) and the future (frame 13 and 19) to the model. To achieve long-term coherence, we introduce $RBConvLSTM$, a multi-layer bi-directional ConvLSTM with residual connections between adjacent layers. The dynamics from both sides are gradually propagated to the middle steps and create dynamic constraint vectors to guide the inference step by step.

To model the uncertainty in the interval, we propose a stochastic model under the bi-directional constraints. At step t, a distribution for an embedding vector is inferred, conditioned on previously generated frames and the reference frames. We sample an embedding vector from the distribution and use a decoder to generate the frame at step t.

We design our objective function by optimizing a variational lower bound (see 3.6). SDVI achieves state-of-the-art performance on 4 datasets. We also infill every between-frame interval of a real-world video (2fps) and connect them to create an enhanced long video of 16fps (See the video in `https://xharlie.github.io/projects/project_sites/SDVI/video_results.html`). To summarize, our contributions are:

- To the best of our knowledge, it is the first stochastic model and the first study utilizes the extended frames away from the interval to solve the video infilling.
- A module $RBConvLSTM$(see 3.1) is introduced to enforce spatial-temporal coherence.
- A spatial feature map is applied in the sampling to enable spatial independence of different regions.
- A metric LMS (see 4) is proposed to evaluate the sequence temporal coherence.

2. Related Works

Most studies of video interpolation [13, 15] focus on generating high-quality intermediate frames in a short-term interval. Since we focus on long-term sequence infilling, our framework adopts long-term dynamics modeling. Therefore we also refer to the studies of video prediction which have explored this area from various perspectives.

2.1. Video Interpolation

Video interpolation generally has three approaches: optical flow based interpolation, phase-based interpolation, and pixels motion transformation. Optical flow based methods [12, 36, 13] require an accurate optical flow inference. However, the optical flow estimation is known to be inaccurate for a long time interval. Estimating motion dynamics becomes a more favorable option. The phase-based methods such as [22] modify the pixel phase to generate intermediate frames. Although the strategy of propagating phase information is elegant, the high-frequency and drastic changes cannot be properly handled. The inter-frame change will be more significant in our long-term setting. Currently studies [18, 25, 15] use deep learning methods to infer the motion

flows between the two frames. By far, this branch of approaches achieves the best result and has the potential to solve our task. In our evaluation, we use SepConv [24] and SuperSloMo [15] as comparisons.

2.2. Deterministic Video Prediction

The mainstream video prediction methods take short consecutive sequences as input and generate deterministic futures by iteratively predicting next frame. [32, 20] use a convolutional network to generate each pixel of the new frame directly. Studies such as [28, 7] use a recurrent network to model the dynamics and improve the result drastically. [35] introduces ConvLSTM, which has been proved to be powerful in spatial sequence modeling. [31, 5] propose to model the content and the dynamics independently to reduce the workload for the networks. [30, 31] incorporate GANs [9] into their model and improve the quality. Notably, two of the generative models [19] and [3] can also conduct video completion. However both methods, due to their forward generation mechanism, cannot hold the coherence between the last frame and the generated sequence. SDVI adopts the decomposition of the motion and the content, uses the ConvLSTM in the motion inference and iteratively generates the frame. However, we do not use GANs since our study focuses more on dynamics generation. We also compare SDVI with FSTN in [19], a prediction model that can also handle video infilling.

2.3. Stochastic Video Generation

After [1, 11, 33] shows the importance of the stochasticity in video prediction, later studies such as [17, 4] also conduct the prediction in the form of stochastic sampling. The stochastic prediction process consists of a deterministic distribution inference and a dynamic vector sampling. We also adopt this general procedure. Since SVG-LP introduced in ([4]) is one of the state-of-the-art models and very related to our study, we use the SVG-LP to compare with SDVI. A concurrent work [14] can generate an intermediate frame between two given frames. However, the model inclines to generate the frame at the time with low-uncertainty. Therefore their model cannot solve the infilling task since the generated sequence does not have a constant frame density.

3. Model Details

As illustrated in Figure 3, SDVI consists of 3 major modules: Reference, Inference and Posterior modules. Given reference frames $\mathbf{X_{WR}}$, Reference module propagates the constraints and generate a constraint vector $\hat{h}_t$ for time t. Inference module takes $\hat{h}_t$ and inference an embedding distribution P_{infr} based on $X_{S:t-1}$. Posterior module inference another embedding distribution P_{pst} based on $X_{S:t}$. We sample an embedding vector z_t from P_{infr} and another

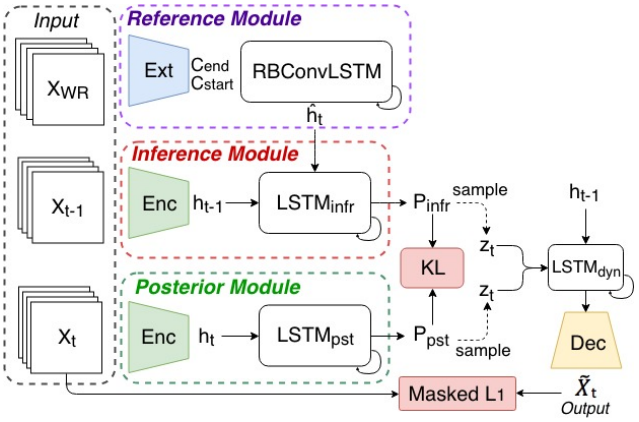

Figure 3: Training of SDVI: All $Encoder$ (green) share the same weights. The blue and the yellow network are $Extractor$ and $Decoder$. Reference module creates dynamic constraint $\hat{h}_t$ at each step. At step t, Inference module takes X_{t-1} and $\hat{h}_t$, while Posterior module takes X_t. Inference module and Posterior module will produce different z_t and therefore different output frames $\tilde{X}_t^{infr}$ and $\tilde{X}_t^{pst}$.

z_t from P_{pst}. A decoder is used to generate a frame $\tilde{X}_t$ for a given z_t. During training, we use KL divergence to minimize the distance between P_{infr} and P_{pst}. At test, Posterior module is not required and z_t is sampled from P_{infr}. We list the notations as follows:

* t: a time step between start step S and terminal step T.
* S:t: the sequence start from step S to step t.
* X_t: The ground truth frame at time step t.
* $\tilde{X}_t$: The frame generated on step t.
* C_{start} and C_{end}: Momentum vectors extracted from $\mathbf{X_{WR}}$, used as initial cell states for $RBConvLSTM$.
* h_t: The dynamic vector extracted from X_t.
* $\hat{h}_t$: The constraint vector at the step t.
* P_{infr} and P_{pst}: The distributions of the embedding vector generated by Inference and Posterior module.
* z_t: The embedding vector on time step t. z_t^{infr} is sampled from P_{infr} and z_t^{pst} is sampled from P_{pst}.

3.1. Reference Module

Reference module includes an $Extractor$ and a $RBConvLSTM$. Given all the frames in $\mathbf{X_{WR}}$, the $Extractor$ learns the momentum and output two vectors C_{start} and C_{end}. With the dynamics and momentum of X_S and X_T, $RBConvLSTM$ outputs a constraint vector $\hat{h}_t$ for each intermediate step t. The whole sequence of the constraint vector has a conditional distribution $P(\hat{h}_{S:T}|\mathbf{X_{WR}})$.

RBConvLSTM $RBConvLSTM$, a residual bi-directional ConvLSTM, is based on the studies of seq2seq [29, 2, 35]. As shown in Figure 4, the first layer of $RBConvLSTM$ uses C_{start} as the initial state of the

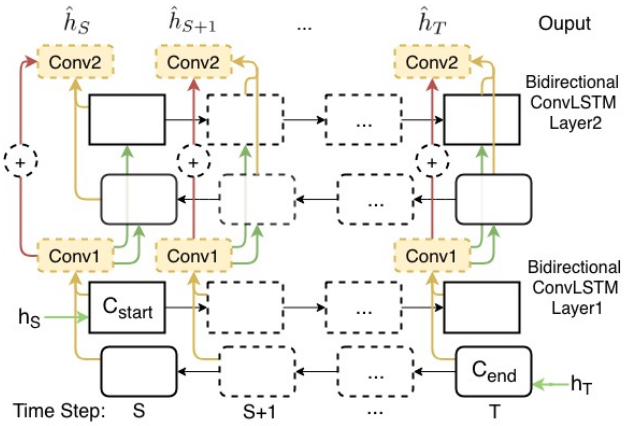

Figure 4: A two layers RBConvLSTM: The initial cell states of the first layer are assigned as C_{start} and C_{end}. h_S and h_T are taken as inputs. Combined with the residuals (red arrows), each layer's outputs (yellow arrows) would go through a convolution module and become the inputs (green arrows) to the next layer.

forward cell and C_{end} for the backward cell. We need to propagate sparse constraints h_S, h_T to every time step from $S+1$ to $T-1$ to get outputs $\hat{h}_S, \hat{h}_{S+1}, ..., \hat{h}_T$ as constraint vectors for Inference Module. They are critical to achieve the long-term coherence. Since the input features to the bottom layer $h_S, \mathbf{0}, ..., h_T$ share the same feature space with $\hat{h}_{S:T}$, inspired by [10], we add an residual connection between each two layers to elevate the bottom features directly to the top. In the end, RBConvLSTM combines all the three structures: the ConvLSTM, the bi-directional RNN and the residual connections.

3.2. Inference Module

As shown in Figure 4, We extract a dynamic vector $h_t - 1$ from each $X_t - 1$. $LSTM_{infr}$ takes the h_{t-1} and the constraint vector $\hat{h}_t$, then infers a distribution P_{infr} of a possible dynamic change. This module resembles the prior distribution learning of stochastic prediction, however, P_{infr} here is written as $P_{infr} = P(z_t|X_{S:t-1}, \mathbf{X_{WR}})$.

3.3. Posterior Module

A generated sequence $\tilde{X}_{S+1:T-1}$ can still be valid even it is different from the ground truth $X_{S+1:T-1}$. Therefore our model need to acquire a target distribution P_{pst} for step t, so Inference module can be trained by matching P_{infr} to the target. Here we expose the frame X_t to Posterior module, so it can generate a posterior distribution $P_{pst} = P(z_t|X_{S:t})$ for P_{infr} to match.

3.4. Training and Inference

From P_{pst}, we can sample a embedding vector vector z_t^{pst}. Conditioned on the previous ground truth frames and

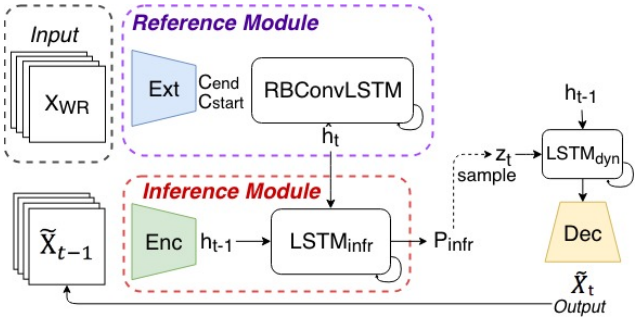

Figure 5: Inference of SDVI: Without ground truth frame X_{t-1}, the generated frame $\tilde{X}_{t-1}$ serves as the input to Inference module on step t.

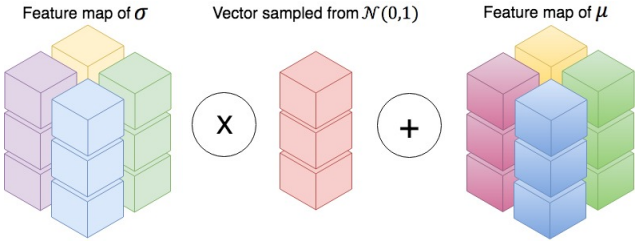

Figure 6: The sampled vector (in the middle) is applied on all locations.

z_t^{pst}, the $Decoder$ generates the $\tilde{X}_t^{pst}$. Separately, we also sample a vector z_t^{infr} from P_{infr}, and generate the $\tilde{X}_t^{infr}$ in the same way.

Since the ground truth frames X_t is not available at time t, we can only use Inference module to sample a z_t. The inference pipeline is shown in Figure 5.

3.5. Dynamic Spatial Sampling

Using the re-parameterization trick [16], we model P_{pst} and P_{infr} as Gaussian distributions $N_{pst}(\mu_t, \sigma_t)$ and $N_{infr}(\mu_t, \sigma_t)$. Different locations in one frame may have different levels of uncertainty. Uniformly draw a sample following the same distribution everywhere will hinder the modeling (see SDVI non-spatial in Table 1). Consequently, we introduce a spatial sampling process (Figure 6). Instead of using vectors [4], we use spatial feature maps for μ_t and σ_t. To get the z_t, we multiply the sampled vector on each location of σ_t, then add the μ_t on the product.

3.6. Loss Function

Pixel Loss To make the $\tilde{X}_t^{pst}$ reconstruct real X_t, we introduce a pixel reconstruction loss $L_1(X_t, \tilde{X}_t^{pst})$. We also observe that imposing a pixel prediction loss to the $\tilde{X}_t^{infr}$ after the P_{infr} getting stable can further improve the video quality during inference.

KL Divergence Loss $P_{pst}(z_t|X_{S:t})$ carries the dynamics from the h_t to reconstruct the X_t. Since we use only Inference module during inference, the $P_{infr}(z_t|X_{S:t-1}, \mathbf{X_{WR}})$ needs to predict the embedding vector alone. Therefore, we also add two KL divergences between P_{infr} and P_{pst}:

$$L_{KL} = D_{KL}(P_{pst}||P_{infr}) + D_{KL}(P_{infr}||P_{pst}) \quad (1)$$

Both the forward and the reverse KL-divergence of P_{infr} and P_{pst} achieve the minimum when the two distributions are equal. However, according to [8], since $D_{KL}(P_{pst}||P_{infr})$ is sampled on P_{pst}, it will penalize more when P_{pst} is large and P_{infr} is small. Therefore this term will lead to a P_{infr} with higher diversity. On the other hand, $D_{KL}(P_{infr}||P_{pst})$ will make the inference more accurate when P_{infr} has large value. To better serve our task, we decide to keep both terms to strike a balance between accuracy and diversity.

Full Loss Overall, our final objective is to minimize the combined full loss:

$$L_C = \sum_{t=S+1}^{T-1} [\beta \cdot \underbrace{L_1(X_t, \tilde{X}_t^{pst})}_{\text{pixel reconstruction loss}} + (1-\beta) \cdot \underbrace{L_1(X_t, \tilde{X}_t^{infr})}_{\text{pixel prediction loss}}$$
$$+ \alpha \cdot \underbrace{D_{KL}(P_{pst}||P_{infr})}_{\text{inclusive KL loss}} + \alpha \cdot \underbrace{D_{KL}(P_{infr}||P_{pst})}_{\text{exclusive KL loss}}] \quad (2)$$

The β balances the posterior reconstruction and the inference reconstruction, while the α determines the trade-off between the reconstruction and the similarity of the two distributions. To show the effectiveness of these loss terms, we also compare the full loss (2) with a loss only composed of the pixel reconstruction loss and the inclusive KL loss (similar to the loss in [4]), shown as "SDVI loss term 1&3" in Table 1.

4. Experiments

Datasets We first test SDVI on 3 datasets with stochastic dynamics: Stochastic Moving MNIST(SM-MNIST) [4] with random momentum after a bounce, KTH Action Database [26] for deformable objects and BAIR robot pushing dataset [6] for sudden stochastic movements. We also compare with the interpolation models on a challenging real-world dataset, UCF101[27].

Last Momentum Similarity and Other Metrics Three metrics are used for quantitative evaluation: structural similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Last Momentum Similarity (LMS).

An infilled sequence that is different from the ground truth (low SSIM and PSNR) is still valid if it can guarantee the long-term coherence between $\tilde{X}_{S:T-1}$ and X_T. Thus we introduce the last momentum similarity (LMS) calculated by the mean square distance between the optical flow from X_{T-1} to X_T and the optical flow from $\tilde{X}_{T-1}^{infr}$ to X_T. We find LMS a good indicator of the video coherence since no matter how the dynamic being sampled, both the object's position and speed should make a smooth transition to X_T.

Movement Weight Map During training, we apply a movement weight map to each location of the pixel loss to encourage movement generation. For a ground truth frame X_t, if a pixel value stays the same in X_{t-1}, the weight is 1 on that location. Otherwise, we set the weight to be $\eta > 1$ to encourage the moving region. This operation helps us to prevent the generation of sequences.

Main Comparison We compare our model with the-state-of-the-art studies of both video interpolation and video prediction (with modification). Except for SuperSloMo, all models are trained from scratch under the same conditions for all datasets.

We select two high-performance interpolation models SepConv[24] and SuperSloMo[15]. Due to SepConv's limitation (sequences must have the length of $2^n - 1$), all the following evaluations are under the generation of 7 frames. Following their instruction, we complete the training code of SepConv. We can't get the code of SuperSloMo, but we acquire the results from the authors of [15].

Two prediction models are picked: FSTN[19], a deterministic generation model; SVG-LP[4], an advanced stochastic prediction model. Since FSTN and SVG-LP are not designed to solve the infilling task, we concatenate the representation of the last frame X_T to their dynamic feature maps in each step. Then the SVG-LP simulates SDVI without Reference module, and the FSTN is equivalent to SDVI without Reference and Posterior module.

Ablation Studies Ablation studies are as follows:

1. To show that the spatial sampling enables spatial independence, we replace the feature map by a vector in the dynamic sampling process and denote it by "SDVI non-spatial". If we up-sample a vector, the information from one area would have an equivalent influence to another area. Therefore it tends to generate a sequence with a single movement (Figure 8 non-spatial).

2. To show the benefit of extra reference frames, we remove the extended reference frames in $\mathbf{X_{WR}}$. We denote this setting by "SDVI without 0 & 24".

3. Our loss has two more terms than another stochastic model [4]. Therefore we also conduct experiments with only the pixel reconstruction loss and the inclusive KL loss. We denote this setting by "SDVI loss term 1 & 3".

Since our model is stochastic, we draw 100 samples for each interval as in [1, 5, 17] and report a sample with the best SSIM. More video results for various settings (see the

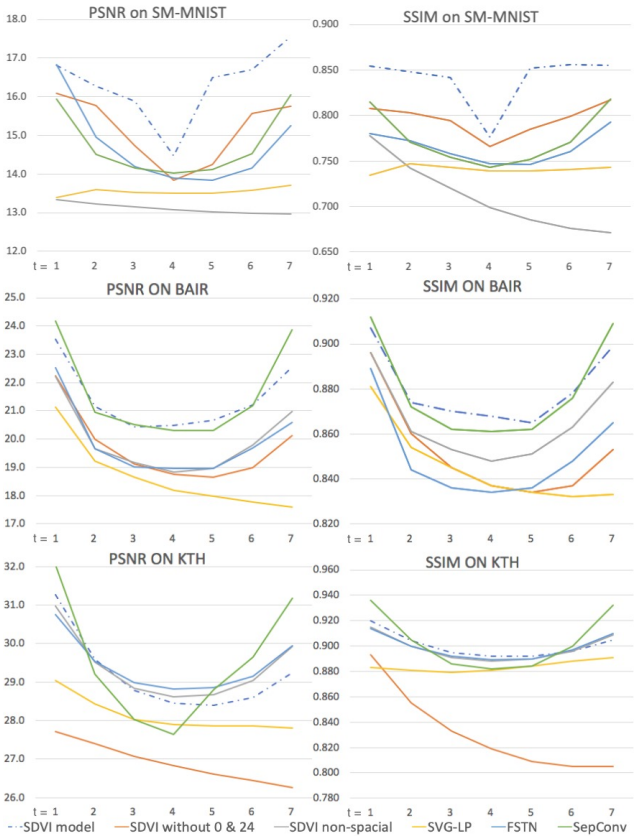

Figure 7: Average PSNR and SSIM at each step in test sets.

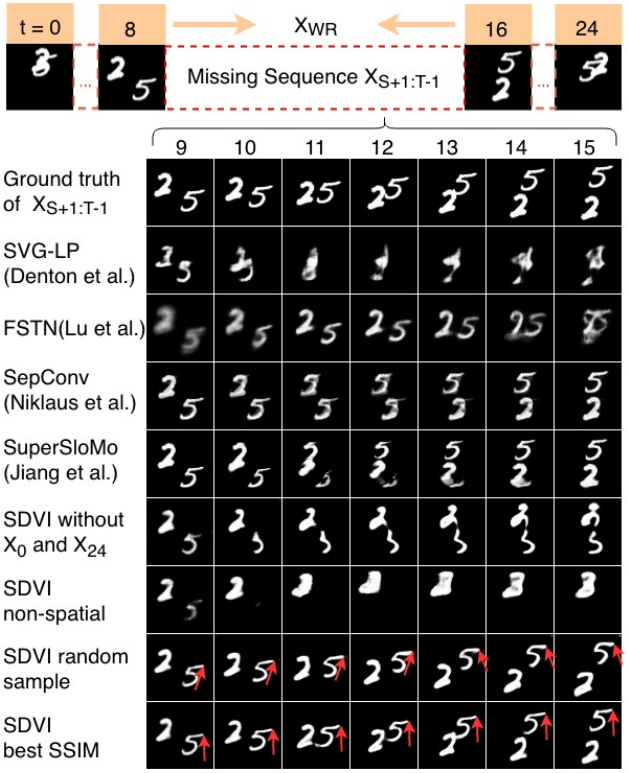

Figure 8: The digit 5 in our best sequence follows the upward trajectory of the ground truth. In another sampled sequence, the 5 goes upper right and then bounce upper left.

video web page), dataset details (see Appendix B), network architectures and the training details (see Appendix A) can be found in the supplemental material.

4.1. Stochastic Moving MNIST (SM-MNIST)

Digits in SM-MNIST introduced by [4] will bounce off the wall with a random speed and direction. The uncertainty of the outcome after a bounce makes it a challenging task for all methods. The Avg PSNR, SSIM and LMS over all test frames are shown in Table 1. We also plot the metric values averaging on each step in Figure 7. Figure 8 shows the qualitative evaluation for all comparisons. When the two digits in frames 8 and 16 having significant position differences, interpolation models such as SepConv and SuperSloMo would still choose to move the pixel based on the proximity between the two frames: the digits 2 and 5 gradually transfer to each other since the 2 in frame 8 is closer to the 5 in frame 16. Because the deterministic model FSTN cannot handle the uncertainty after a bounce, the model gets confused and generates a blurry result. The SVG-LP cannot converge in this setting since it doesn't have a constraint planning module like the $RBConvLSTM$ to lead the sequence to the final frame. Without spatial independence, a non-spatial representation cannot sample different dynamics for different areas. The two digits in the result of "SDVI non-spatial" collapse into one, then move toward the final frame. Finally, our full model can learn the bouncing rule and provide plausible alternative sequences. Although our randomly sampled sequence diverges from the ground truth, this sequence can still keep the coherence with frame 8 and 16 under plausible dynamics.

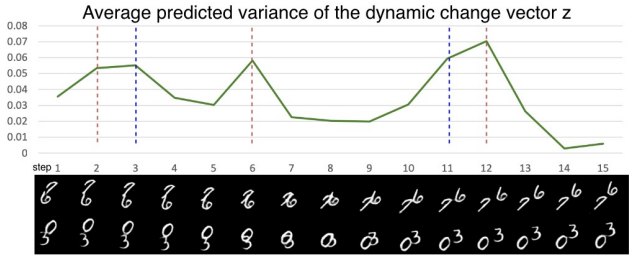

Figure 9: SDVI generates higher variances coincident to the "wall bouncing" event, indicated by the two dash lines(e.g. first sequence: red lines mark the bounces of the digit 6 and blue ones mark the bounces of 7).

We also study how good our method models the uncertainty as in [4]. In 768 test sequences, we randomly select two digits for each sequence and synchronize all sequences'

	SM-MNIST			BAIR			KTH			UCF101		
	PSNR	SSIM	LMS	PSNR	SSIM	LMS	PSNR	SSIM	LMS	PSNR	SSIM	LMS
SDVI full model	**16.025**	**0.842**	**0.503**	21.432	**0.880**	**1.05**	29.190	0.901	**0.248**	**16.635**	**0.598**	**15.678**
SDVI without 0 & 24	14.857	0.782	1.353	19.694	0.852	1.360	26.907	0.831	0.478	—	—	—
SDVI non-spatial	13.207	0.752	6.394	19.938	0.865	1.159	29.366	0.896	0.276	—	—	—
SDVI loss term 1&3	15.223	0.801	0.632	19.456	0.849	1.277	28.541	0.854	0.320	—	—	—
SVG-LP	13.543	0.741	5.393	18.648	0.846	1.891	28.131	0.883	0.539	—	—	—
FSTN	14.730	0.765	3.773	19.908	0.850	1.332	**29.431**	0.899	0.264	—	—	—
SepConv	14.759	0.775	2.160	**21.615**	0.877	1.237	29.210	**0.904**	0.261	15.588	0.443	20.054
SuperSloMo	13.387	0.749	2.980	—	—	—	28.756	0.893	0.270	15.657	0.471	19.757

Table 1: Metrics averaging over all 7 intermediate frames. We report the scores of the best-sampled sequences for SDVI.

trajectories. Figure 9 shows the normalized average variance of the distribution of z_t for frames 2 to 14 (generated), while frame 1 and 15 are the ground truth frames.

4.2. BAIR robot pushing dataset

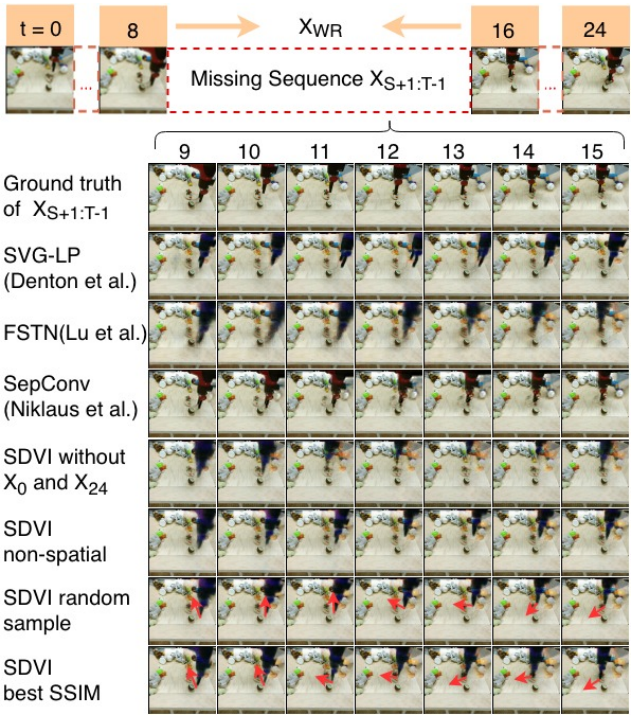

Figure 10: The arm in the best sequence follows the same movements in ground truth: first upward left then downward left. In another sampled sequence, the arm firstly goes straight up and then straight left, finally downward left.

The BAIR robot pushing dataset [6] contains sequences of a robot arm pushing various objects in the RGB domain. The movements of the arm do not follow smooth trajectories, and the movement changes are prompt. As shown in Table 1, although our SDVI marginally outperforms other models on SSIM, the SepConv achieves the best PSNR. As shown in Figure 10, since the SepConv relies more on pixel proximity, the shapes of the static objects in this method are nicely preserved. However, SepConv can't model the stochasticity while its movement is simplified to a straight sliding. The frames in the middle suffer the most in all metrics (Figure 7). The stochasticity of the movement makes it hard for SVG-LP's arm to go back to the final frame and for FSTN to generate sharp shapes. The objects created by SDVI without spatial sampling are more blurry since all the areas will be disturbed by the change of the dynamics. On the other hand, the result of SDVI without using reference frames 0 and 24 diverges too much away from the ground truth movement. Our full model cannot only sample a similar sequence to the ground truth, but sequences with reasonably varied movements (last two rows in Figure 10).

4.3. KTH Action Dataset

The KTH Action dataset [26] contains real-world videos of multiple human actions. In our setting, all actions are trained together. Although the background is uniform, there is still some pixel noise. We found setting the map's weight to 3 on moving pixels is beneficial. Since most actions such as waving follow fixed patterns, the FSTN and the SepConv can achieve the best scores in PSNR and SSIM (Table 1). However, if the object in frame 8 and 16 has a similar body pose, the SepConv and the SuperSloMo will freeze the object's body and slide the object to its new position (Figure 13). SDVI without frame 0 and 24 suffers from the uncertainty of the initial state (Figure 13). The result of FSTN (in 12) contains blurry pixels on moving region although it keeps the static parts sharp. Our sequence with best SSIM has a similar pattern as the ground truth. Even the random sampled sequence shown in Figure 12 has different dynamics in the middle, its initial and final movements still stick to the ground truth. Therefore our model still achieves an outstanding LMS over other methods (Table 1).

4.4. UCF101

Collected from YouTube, UCF101 contains realistic human actions captured under various camera conditions. We train both our model and SepConv on the training set of UCF101, and we get the result from the authors of [15]. Our results over 557 test videos outperform both SepConv and

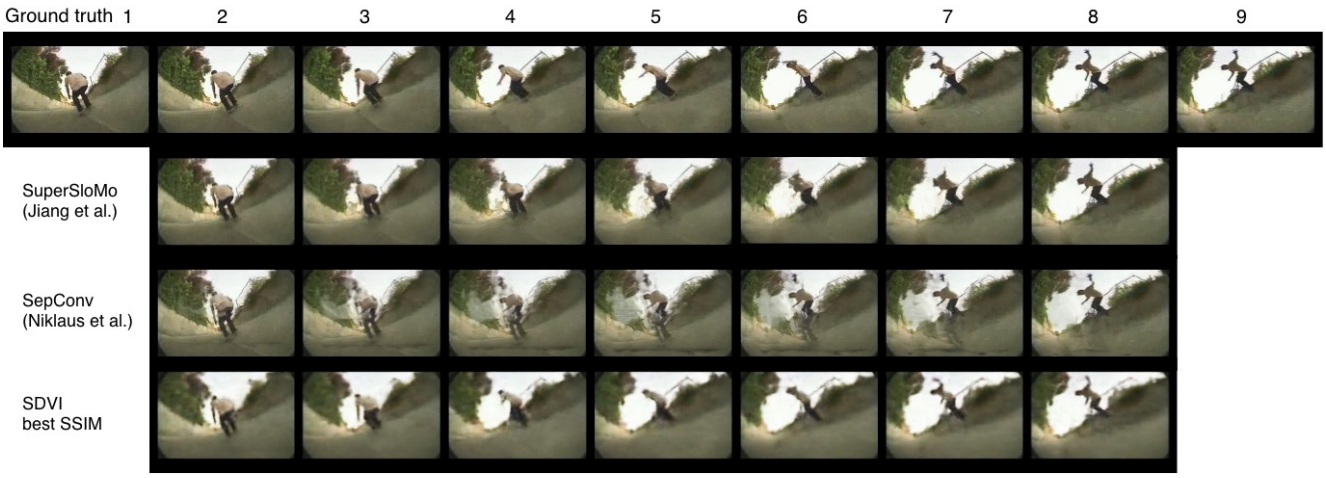

Figure 11: Best view in color. See Appendix C in the supplemental material for more comparisons on UCF101.

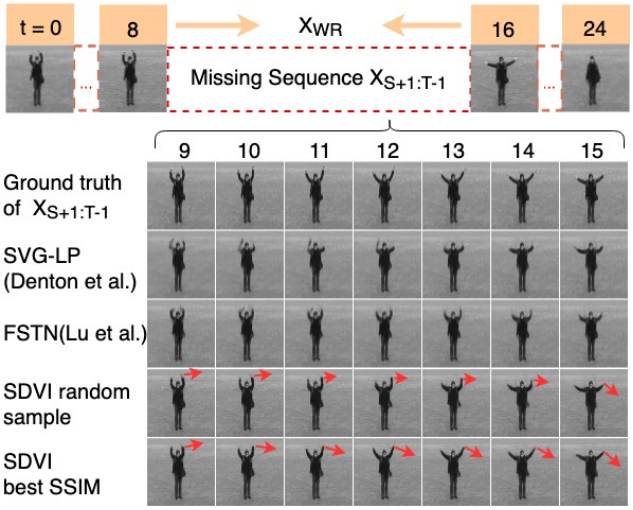

Figure 12: Our best-sampled sequence keeps the arm straight. In a randomly sampled sequence, the forearm bends first then stretches straight in the end.

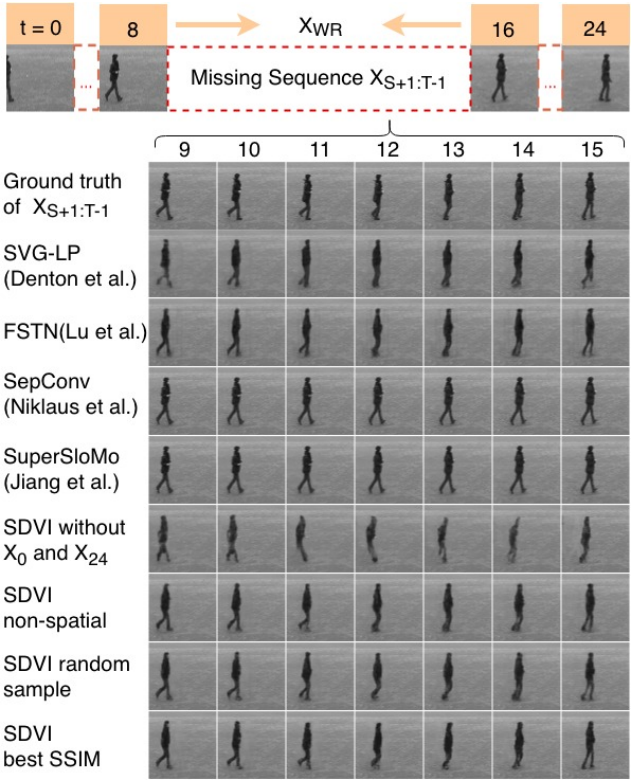

Figure 13: The sliding tendency of SepConv will cause motion errors and high LMS.

SuperSloMo (Table 1). In Figure 11, our model can infer the people's motion consistently. However, it's challenging for SepConv and SuperSloMo to estimate the pixel movement for the middle frames(frame 4 to 6), even though they can generate sharper frames near the two reference frames.

5. Conclusions and Future Work

We have presented a stochastic generation framework SDVI that can infill long-term video intervals. To the best of our knowledge, it is the first study using extended reference frames and using a stochastic generation model to infill long intervals in videos. Three modules are introduced to sample a plausible sequence that preserves the coherence and the movement variety. Extensive ablation studies and comparisons with the state-of-the-art methods demonstrate our good performance on 4 datasets. A metric LMS is proposed to evaluate the sequence coherence. Although currently SDVI can be iteratively applied to infill an interval with any numbers of frames, its flexibility could be further improved. Another direction is to enhance the generality of the model to work across different video domains.

References

[1] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang. Deep video generation, prediction and completion of human action sequences. *arXiv preprint arXiv:1711.08682*, 2017.

[4] E. Denton and R. Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[5] E. L. Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pages 4414–4423, 2017.

[6] F. Ebert, C. Finn, A. X. Lee, and S. Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017.

[7] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016.

[8] C. W. Fox and S. J. Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. corr, vol. abs/1512.03385, 2015.

[11] M. Henaff, J. Zhao, and Y. LeCun. Prediction under uncertainty with error-encoding networks. *arXiv preprint arXiv:1711.04994*, 2017.

[12] E. Herbst, S. Seitz, and S. Baker. Occlusion reasoning for temporal interpolation using optical flow. *Department of Computer Science and Engineering, University of Washington, Tech. Rep. UW-CSE-09-08-01*, 2009.

[13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[14] D. Jayaraman, F. Ebert, A. A. Efros, and S. Levine. Time-agnostic prediction: Predicting predictable video frames. *arXiv preprint arXiv:1808.07784*, 2018.

[15] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. *arXiv preprint arXiv:1712.00080*, 2017.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[18] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4473–4481, 2017.

[19] C. Lu, M. Hirsch, and B. Schölkopf. Flexible spatio-temporal networks for video prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6523–6531, 2017.

[20] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[21] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 498–507, 2018.

[22] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1418, 2015.

[23] S. Niklaus and F. Liu. Context-aware synthesis for video frame interpolation. *arXiv preprint arXiv:1803.10967*, 2018.

[24] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[25] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. *CoRR*, abs/1708.01692, 2017.

[26] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[27] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[28] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[29] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[30] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. *arXiv preprint arXiv:1707.04993*, 2017.

[31] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[32] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[33] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[34] N. Wichers, R. Villegas, D. Erhan, and H. Lee. Hierarchical long-term video prediction without supervision. *arXiv preprint arXiv:1806.04768*, 2018.

[35] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances*

in neural information processing systems, pages 802–810, 2015.

[36] Z. Yu, H. Li, Z. Wang, Z. Hu, and C. W. Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(7):1235–1248, 2013.