Research Article

Continuous Video Process: Modeling Videos as Continuous Multi-Dimensional Processes for Video Prediction

Gaurav Shrivastava[1], **Abhinav Shrivastava**[1]

1. University of Maryland, College Park, United States

Diffusion models have made significant strides in image generation, mastering tasks such as unconditional image synthesis, text-image translation, and image-to-image conversions. However, their capability falls short in the realm of video prediction, mainly because they treat videos as a collection of independent images, relying on external constraints such as temporal attention mechanisms to enforce temporal coherence. In our paper, we introduce a novel model class, that treats video as a continuous multi-dimensional process rather than a series of discrete frames. We also report a reduction of 75% sampling steps required to sample a new frame thus making our framework more efficient during the inference time. Through extensive experimentation, we establish state-of-the-art performance in video prediction, validated on benchmark datasets including KTH, BAIR, Human3.6M, and UCF101.[1]

Corresponding authors: Gaurav Shrivastava, gauravsh@umd.edu; Abhinav Shrivastava, abhinav@cs.umd.edu

1. Introduction

In the evolving landscape of machine learning and generative models, particularly in the domain of video representation[1][2][3][4][5][6][7], there exists a pivotal challenge in adequately capturing the dynamic transitions between consecutive frames. In this paper, we introduce a novel approach to video representation that treats the video as a continuous process in multi-dimensions. This methodology is anchored in the observation that transitions between consecutive frames in a video do not uniformly contain the same amount of motion. Modeling these transitions with a single-step process often leads to suboptimal quality in sampling. Our method, therefore, involves multiple predefined steps between two

 doi.org/10.32388/DM98UZ

consecutive frames, drawing inspiration from recent advancements in diffusion models for image data. This multi-step diffusion process has been instrumental in better modeling image data, and we aim to extend this success to video data.

Previous efforts in video modeling with diffusion models have tended to approach videos as a series of images, generating separate volumes of video frame sequences and applying external constraints such as applying temporal attention to maintain the temporal coherence. We argue that this approach overlooks the inherent continuity in video data, which can be more naturally conceptualized as a continuous multi-dimensional process. Our proposed method[8] defines this continuous process, beginning with two consecutive frames from a video sequence as endpoints this can be observed in Fig. 1. We delineate the forward process through interpolation between these endpoints, with a predefined number of steps guiding the transition from one point to another. To ensure the existence of $p(\mathbf{x}_t)$ at all points, we introduce a novel noise schedule that applies zero noise at both endpoints.

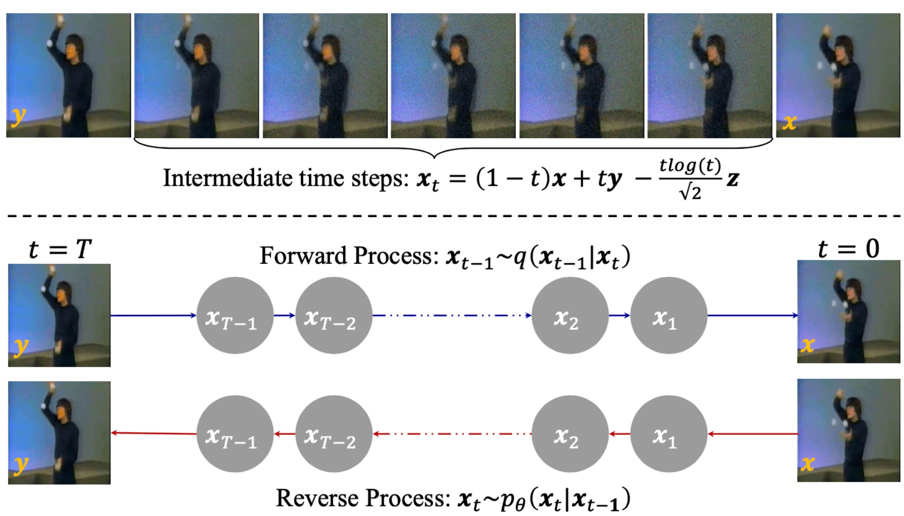

Figure 1. The figure is divided into two parts. The top portion of the figure illustrates the intermediate frames $\mathbf{x}_t$ between two consecutive frames. $\mathbf{x}, \mathbf{y}$ represents consecutive frames from a video sequence where $\mathbf{y} = \mathbf{x}^{j+1}$ and $\mathbf{x} = \mathbf{x}^j$. $\mathbf{x}^j$ denotes some frame at timestep j in the video sequence $\mathcal{V} = \{\mathbf{x}^i\}_{i=1}^N$. $\mathbf{z}$ denotes the white noise. The lower portion of the figure represents the directed graphical model considered in this work to represent the continuous video process.

We approximate each step between these endpoints using a Gaussian distribution, following the assumptions made in diffusion models for images by the paper[9][10][11][12]. In defining this forward

doi.org/10.32388/DM98UZ

process, we also lay the groundwork for estimating a reverse process. This paper presents a novel lower variational bound for estimating this reverse process.

To summarize, our contribution in this work is as follows:

- We introduce a novel approach for representing videos as multi-dimensional continuous processes.

- We introduce a novel approach for representing videos as multi-dimensional continuous processes.
- We derive a novel variational bound that efficiently estimates the reverse process in our proposed 'Continuous Video Process (CVP)' model.
- Our method employs a unique noise schedule for the continuous video process, characterized by zero noise at both endpoints, ensuring the existence of $p(\mathbf{x}_t)$ at all intermediate timesteps.
- We demonstrate the efficacy of our approach through state-of-the-art results in video prediction tasks across four different datasets namely, KTH action recognition, BAIR robot push, Human3.6M, and UCF101 datasets. Additionally, our model requires 75% fewer sampling steps when sampling a frame compared to a diffusion-based baseline.

2. Related Works

Understanding and predicting future states based on observed past data[13][14][15][16][17][18][19] is a cornerstone challenge in the domain of machine learning. It is crucial for video-based applications where capturing the inherent multi-modality of future states is vital, such as in autonomous vehicles. Early methods in this field, as noted by Yuen et al.[20] and Walker et al.[21], primarily focused on matching past frames within datasets to extrapolate future states, although these predictions were constrained to either symbolic trajectories or directly retrieved future frames. The advent of deep learning has significantly propelled advancements in this area. One of the seminal works by Srivastava et al.[22] leveraged a multi-layer LSTM network for deterministic representation learning of video sequences. Subsequent studies[23][24][25][26][27][28][29], have expanded the scope of this research by constructing models that account for the stochastic nature of future states, marking a notable shift from earlier deterministic approaches.

Recent research in this domain has explored both implicit and explicit probabilistic modeling approaches. Implicit probabilistic modeling, typified by GAN[30]-based models, has a substantial history. Nonetheless, these models[31][32][33] often grapple with training stability issues and mode collapse(where model only focuses on a few modes in the dataset) issues. On the other hand, explicit probabilistic modeling for video prediction encompasses a range of methodologies, including Variational Autoencoders (VAEs)[34], Gaussian processes, and Diffusion models. VAE-based video prediction methods[35][36][31] tend to average results to align with all potential future scenarios, which undermines the fidelity of predictions. Gaussian process-based models[1][37] exhibit proficiency with smaller datasets but encounter scalability issues owing to matrix inversion limitations when calculating training likelihood. While workarounds exist, they tend to compromise result fidelity.

Recent advancements in diffusion models[38][39][40][41] have positioned them as the preferred choice for video prediction tasks. These multi-step models offer superior sample quality and are resilient to mode collapse. However, even with such lucrative advantages, modeling videos with these models tends to have downsides. Majorly methods falling under this category enforce temporal consistency using artificial external constraints such as the introduction of temporal attention blocks. This might be effective but comes at a cost of significant computing power.

Another class of popular video prediction models is hierarchical prediction[42][43][44][45][2] models. These models are multistage models that decompose the problems into two stages. They first predict a high-level structure of a video, like a human pose, and then leverage that structure to make predictions at the pixel level. These models generally require additional annotation for the high-level structure for training, unlike ours that predicts future frames utilizing only the pixel-level information of context frames.

We also want to highlight some very recent works like InDI[46], and Cold diffusion[47] that provide an alternate approach to denoising diffusion models that is similar to our approach. However, their works only explored such formulation for image-based computational photography and image generation tasks.

3. Method

Instead of introducing noise iteratively to the frames until they conform to a Gaussian distribution, and adopting a reverse process such as denoising diffusion, a commonly employed technique for video prediction, we introduce a novel model category designed to depict videos as continuous processes. This section delves into the modeling of this continuous video process.

Suppose we have a video sequence denoted by $\mathcal{V} = \{\mathbf{x}^j\}_1^N$ where $\mathbf{x}^j \in \mathbb{R}^{c\times h\times w}$ is the frame at the timestep j. We represent this video sequence as a continuous process. The intermediate frames between $\mathbf{x} = \mathbf{x}^j$ and $\mathbf{y} = \mathbf{x}^{j+1}$ are given by the following equation.

$$\mathbf{x}_t = (1-t)\mathbf{x} + t\mathbf{y} - \frac{t\log(t)}{\sqrt{2}}\mathbf{z} \tag{1}$$

doi.org/10.32388/DM98UZ

Here, $\mathbf{z} \sim \mathcal{N}(0, I)$ denotes the white noise. From the above Eqn, it can be seen that at $t = 0$, we get the frame $\mathbf{x}^j$ and at $t = 1$, we get the frame $\mathbf{x}^{j+1}$. We utilize this continuous process of evolving $\mathbf{x}^j \to \mathbf{x}^{j+1}$ given by Eqn. 1 and derive both the forward and reverse processes. For defining the forward process, we take steps in the direction $t : T \to 0$ instead of the other way, which happens in denoising diffusion process[9]. The reason for this is we want the reverse process to start from past frame $\mathbf{x}$ and according to the Eqn. 1 $\mathbf{x}_t = \mathbf{x}$ at $t = 0$.

We can write the forward process, i.e., going from the start point $\mathbf{y}$ at $t = T$ to endpoint $\mathbf{x}$ at $t = 0$,

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + (\mathbf{y} - \mathbf{x})\Delta t - t\log(t)\mathbf{z} \tag{2}$$

From the above equation, we can write the posterior for the forward process as $q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{x}_{t+1} : \tilde{\mu}(\mathbf{x}_t, \mathbf{x}, \mathbf{y}), g^2(t)I)$. Where $g(t) = -t\log t$. The whole derivation is provided in the appendix.

For modeling our video diffusion process, we like to model the likelihood function $p_\theta(\mathbf{x}_T) := \int p_\theta(\mathbf{x}_{0:T})d\mathbf{x}_{0:T-1}$ and minimize the negative log-likelihood to obtain the best fit for our model. Here, $p_\theta(\mathbf{x}_{0:T})$ is the probability of the reverse process, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_0) = p_{data}(\mathbf{x})$. Important note about the notations $\mathbf{x}_0, \mathbf{x}_T$, unless specified consider $\mathbf{x}_0 = \mathbf{x}$ and $\mathbf{x}_T = \mathbf{y}$ where $\mathbf{x}$ is the frame in the video sequence at j^{th} position and $\mathbf{y}$ is the frame at $(j+1)^{th}$ position. One important assumption about the continuous video process is we assume the transition between the frames $\mathbf{x}$ and $\mathbf{y}$ to follow Markov chain, i.e., the current state at timestep t only depends on the previous state at timestep $t-1$. Leveraging this assumption we can define the reverse process as follows,

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_0)\prod_{t=1}^{T} p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3}$$

where, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_t; \mu_\theta(\mathbf{x}_{t-1}, t-1), \mathbf{\Sigma}_\theta(\mathbf{x}_{t-1}, t-1))$. We are interested in learning the reverse process to perform our video prediction task.

The forward process or the diffusion process is a fixed Markov chain that gradually transforms the frame $\mathbf{y}$ to frame $\mathbf{x}$.

$$q(\mathbf{x}_{0:T-1}|\mathbf{x}_T) := \prod_{t=1}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{4}$$

Training is performed by minimizing the variational bound on the negative log-likelihood.

 doi.org/10.32388/DM98UZ

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_T)] \leq \mathbb{E}_q[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{0:T-1}|\mathbf{x}_T)}] \qquad (5)$$

$$\leq \mathbb{E}_q[-\log p(\mathbf{x}_0) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}] \qquad (6)$$

$$=: L(\theta) \qquad (7)$$

This variational bound can be simplified to the following (we refer the readers to the appendix to follow the simplification of from Eqn. 7 to the following equation),

$$L(\theta) =: \sum_{t\geq 1} D_{KL}(q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}, \mathbf{y}) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x})) \tag{8}$$

In the above Eqn, the KL divergence term utilizes the comparison of $p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x})$ with forward process posterior term, which is tractable under the process given by Eqn. 2. The forward process posterior term is given by

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{x}_t : \tilde{\mu}(\mathbf{x}_{t-1}, \mathbf{x}, \mathbf{y}), g^2(t)I) \tag{9}$$

where, $\tilde{\mu}(\mathbf{x}_t, \mathbf{x}, \mathbf{y}) = \mathbf{x}_t + (\mathbf{y} - \mathbf{x})$ and $g(t) = -t\log(t)$. Consequently, all KL divergences in Eqn. 8 are comparisons between Gaussians, so they can be calculated in a Rao-Blackwellized fashion with closed-form expressions instead of high-variance Monte Carlo estimates. It is important to note while deriving the Eqn. 8, we ignore some terms that purely involve the forward process posteriors as q has no learnable parameters, so such terms are constants during training.

Now we discuss our choices in $p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \mu_\theta(\mathbf{x}_{t-1}, t-1, \mathbf{x}), \Sigma_\theta(\mathbf{x}_{t-1}, t-1, \mathbf{x}))$ for $1 < t \leq T$. First, we set $\Sigma_\theta(\mathbf{x}_{t-1}, t-1) = g^2(t)I$ to untrained time dependent constants. Experimentally, the choice of $g(t) = -t\log(t)$ works the best. This noise function has an interesting property that noise is absent both at the start and end points, i.e., $g(t) = 0 \ \ \forall t = \{0, 1\}$.

Second, to represent the mean $\mu_\theta(\mathbf{x}_t, t, \mathbf{x})$, we propose a specific parameterization motivated by the forward process posterior given by Eqn. 9. With $p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \mu_\theta(\mathbf{x}_{t-1}, t-1, \mathbf{x}), g^2(t)\mathbf{I})$, we can write:

$$L(\theta) := \mathbb{E}_q\left[\frac{1}{2g^2(t)}\|\tilde{\mu}(\mathbf{x}_t, \mathbf{x}, \mathbf{y}) - \mu_\theta(\mathbf{x}_t, t, \mathbf{x})\|^2\right] + C \tag{10}$$

where C is a constant that does not depend on θ. So, we see that the most straightforward parameterization of μ_θ is a model that predicts $\tilde{\mu}_t$, the forward process posterior mean.

However, we can simplify Eqn. 10 further and obtain a very simple training loss objective by delving in the term $\tilde{\mu}$. We further parameterize the term μ_θ as follows,

$$\mu_\theta(\mathbf{x}_t, t, \mathbf{x}) = \mathbf{x}_t + (\mathbf{y}_\theta(\mathbf{x}_t) - \mathbf{x}) \tag{11}$$

doi.org/10.32388/DM98UZ

When we substitute this $\mu_\theta(\mathbf{x}_t, t, \mathbf{x})$ parameterization in the Eqn. 10 we get the simplified version of the loss $L(\theta)$ as follows,

$$L_{simple}(\theta) := \mathbb{E}_{t,\mathbf{x}_t}\left[\frac{1}{2g^2(t)}\|\mathbf{y} - \mathbf{y}_\theta(\mathbf{x}_t, t)\|_2\right] \tag{12}$$

For training the video prediction model utilizing the above Eqn. 12 we obtain the $\mathbf{x}_t$ as a function of t by leveraging the Eqn. 1. The following equation gives a more generic form of the final loss function utilized to train the video prediction model,

$$\underset{\theta}{\arg\min}\,\mathbb{E}_{t,\mathbf{x},\mathbf{y}}\left[\frac{1}{2g^2(t)}\|\mathbf{y} - \mathbf{y}_\theta((1-t)\mathbf{x} + t\mathbf{y} + \frac{g(t)}{\sqrt{2}}\mathbf{z}, t)\|^2\right] \tag{13}$$

The whole training and sampling pipeline is described in the training Alg. 1, sampling Alg. 2 and depicted in Fig. 2.

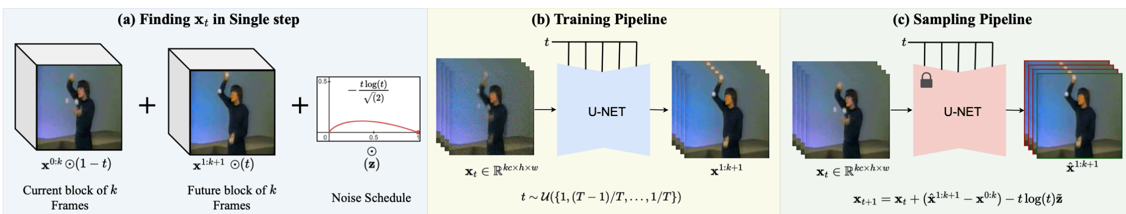

Figure 2. Fig. (a) demonstrates the methodology for estimating $\mathbf{x}_t$ in a single step, showcasing the specific computational process involved. Fig. (b) details the training pipeline of our Continuous Video Process (CVP) model, where $\mathbf{x}_t$ and t are fed as inputs to the U-Net architecture, and the anticipated output is $\hat{\mathbf{y}}$, with $\hat{\mathbf{y}} = \mathbf{x}^{1:k+1}$ in this scenario. Fig. (c) provides an overview of the sampling pipeline utilized in our CVP method, illustrating the sequential steps to predict the next frame of the video sequence given the context frames.

Algorithm 1 Training of CVP model

1: **repeat**
2: $\quad \mathbf{x}, \mathbf{y} \sim q_{\text{data}}(\mathbf{x}, \mathbf{y})$
3: $\quad t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on
$$\nabla_\theta \frac{1}{2g^2(t)}\left\|\mathbf{y} - \mathbf{y}_\theta((1-t)\mathbf{x} + t\mathbf{y} - (t\log(t)/\sqrt{2})\mathbf{z}, t)\right\|^2$$
6: **until** converged

doi.org/10.32388/DM98UZ

Algorithm 2 Sampling Algorithm

1: $\mathbf{x} \sim q_{\text{data}}(\mathbf{x})$
2: $\mathbf{x}_0 = \mathbf{x}$
3: $d = \frac{1}{N}$, Here N denotes number of steps.
4: **for** $t = 1, \ldots, N$ **do**
5: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}d)$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
6: $\mathbf{x}_{t+1} = \mathbf{x}_t + (\hat{y}(\mathbf{x}_t, t) - \mathbf{x})d - t\log(t)\mathbf{z}$
7: **end for**
8: **return** $\mathbf{x}_T$

4. Experiments

Video prediction task can be defined as given a few context frames, the model has to predict the subsequent future frames. In this section, we empirically demonstrate that our approach yields superior results in modeling the video prediction task.

4.1. Datasets

We chose 4 different types of datasets to demonstrate the efficacy of our approach. These are standard benchmarks for video prediction tasks. Dataset lists include KTH action recognition dataset[15], BAIR robot pushing dataset[19], Human3.6M[13] and UCF101[16] datasets. Training and architecture-specific details about the approach are included in the appendix.

KTH Action Recognition Dataset. The KTH action dataset[15] consists of video sequences of 25 people performing six different actions: walking, jogging, running, boxing, hand-waving, and hand-clapping. The background is uniform, and a single person is performing actions in the foreground. The foreground motion of the person in the frame is fairly regular. The frames in the video for this dataset consist of a single channel. The spatial resolution of the frames in the video is downsampled to the size of 64×64.

BAIR pushing Dataset. The BAIR robot pushing dataset[19] contains the videos of table mounted sawyer robotic arm pushing various objects around. The BAIR dataset consists of different actions given to the robotic arm to perform. The spatial resolution of the frames in the video is kept to be 64×64.

Human3.6M Dataset. Human3.6M[13] dataset consists of 10 subjects performing 15 different actions. The pose information from the dataset was not used in predicting next frame. The background is uniform, and a single person is performing actions in the foreground. The foreground motion of the person in the frame

doi.org/10.32388/DM98UZ

is fairly regular. The frames in the video for this dataset consist of ‘RGB’ channels. The spatial resolution of the frames in the video is downsampled to the size of 64×64.

UCF101 Dataset. This dataset[16] consists of 13,320 videos belonging to 101 different action classes. The video seems to have a variety of backgrounds and the frames of the video have three channels, namely ‘RGB’. We reshape the resolution of frames from the original size of 320×240 down to 128×128 for our video prediction tasks. The downsampling is done utilizing the bicubic downsampling.

4.2. Metrics

We primarily use the FVD[48] metric to determine the best-performing baseline when evaluating a video prediction task. FVD metric evaluates a baseline on both terms, the reconstruction quality and diversity of the generated samples. FVD is calculated as the frechet distance between the I3D embeddings of generated video samples and real samples. The I3D network used for obtaining the embeddings for real and generated video is trained on the Kinetics-400 dataset.

5. Setup and Results

Below, we describe in detail how the setup for our experiment looks compared to baselines. We also showcase our findings about the performance of our method and comparison to baselines in this section.

KTH action recognition dataset: For this dataset, we adhered to the baseline setup[38], which utilizes the first 10 frames as context frames. In baseline setup, these 10 frames are utilized to predict the subsequent 30 and 40 frames. A notable aspect of our experiment is we only used the last 4 frames from this sequence of 10 frames as context frames in our CVP model, while disregarding the information in the remaining 6 frames. This decision was taken to maintain consistency with the experimental setups used in prior baseline methodologies. The outcomes of this evaluation are summarized in Table 1.

KTH [10 $\rightarrow$ #pred; trained on k]	k	#pred	**FVD↓**	**PSNR↑**	**SSIM↑**
SVG-LP[35]	10	30	377	28.1	0.844
SAVP[31]	10	30	374	26.5	0.756
MCVD[38]	5	30	323	27.5	0.835
SLAMP[49]	10	30	228	29.4	0.865
SRVP[50]	10	30	222	29.7	0.870
RIVER[39]	10	30	180	**30.4**	0.86
CVP (Ours)	**1**	30	**140.6**	29.8	**0.872**
Struct-vRNN[51]	10	40	395.0	24.29	0.766
SVG-LP[35]	10	40	157.9	23.91	0.800
MCVD[38]	5	40	276.7	26.40	0.812
SAVP-VAE[31]	10	40	145.7	26.00	0.806
Grid-keypoints[52]	10	40	144.2	27.11	0.837
RIVER[39]	10	40	170.5	29.0	0.82
CVP (Ours)	**1**	40	**120.1**	**29.2**	**0.841**

Table 1. Video prediction results on KTH (64×64), predicting 30 and 40 frames using models trained to predict k frames at a time. All models condition on 10 past frames on 256 test videos.

It can be observed from the Table 1, our model's unique approach requires a significantly reduced number of frames for training. Contrary to other methods that train on an additional set of k frames (10[context frames]+k[future frames]), our model uses just one frame (effectively 4[context frames]+1[future frames]). We employ the 4 context frames to predict the immediate next frame and then autoregressively generate either 30 or 40 frames, depending on the evaluation requirement. This methodology is supported by our model's efficient handling of video sequences as continuous processes, which eliminates the need for external artificial constraints, such as temporal attention mechanisms.

The results, as shown in Table 1, clearly indicate that our method delivers state-of-the-art performance when compared to other baseline models. Additionally, the qualitative results for our CVP model on the KTH dataset can be observed in Fig. 3.

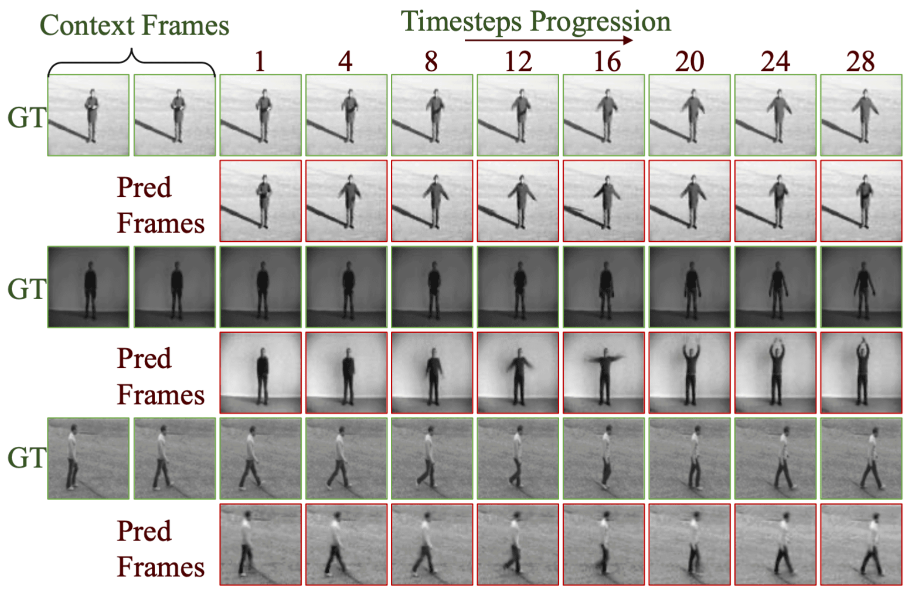

Figure 3. Figure represents qualitative results of our CVP model on the KTH dataset. The number of context frames used in the above setting is 4 for all three sequences. Every 4^{th} predicted future frame is shown in the figure.

BAIR Robot Push dataset: The BAIR Robot Push dataset is characterized by highly stochastic video sequences. In our study, we adhered to a baseline setup[38] with three main experimental settings: 1) using only one context frame to predict the next 15 frames, 2) employing two context frames to predict 14 future frames, and 3) utilizing two context frames to forecast the next 28 frames. The outcomes of these approaches are summarized in Table 2.

As observed in Table 2, a trend emerges where increasing the number of frames predicted at a time concurrently results in a degradation of prediction quality. This phenomenon is hypothesized to stem from an augmented disparity between the blocks of context frames and predicted future frames. Specifically, consider the scenario where two context frames are designated as $\mathbf{x}^{0:2}$, corresponding to $\mathbf{x}$ in the context of Eqn.1. Under the first experimental condition, where the model predicts a single frame at a time, the future frame prediction block is represented as $\mathbf{x}^{1:3}$, analogous to $\mathbf{y}$ in Eqn.1. Conversely, in the second condition, where two frames are predicted simultaneously, the future frame block extends to $\mathbf{x}^{2:4}$,

 doi.org/10.32388/DM98UZ

again paralleling $\mathbf{y}$ in the equation. This setup implies that in the former setting, interpolation occurs between adjacent frames (i.e., the transition from $\mathbf{x}^0 \rightarrow \mathbf{x}^1$ and $\mathbf{x}^1 \rightarrow \mathbf{x}^2$), while in the latter, interpolation spans a two-frame interval (i.e., the transition from $\mathbf{x}^0 \rightarrow \mathbf{x}^2$ and from $\mathbf{x}^1 \rightarrow \mathbf{x}^3$). The expanded interval in the second scenario is posited as the causative factor for the observed reduction in predictive performance, particularly in configurations where $k = 2$ and $p = 2$.

The results, as shown in Table 2, clearly indicate that our method delivers state-of-the-art performance compared to other baseline models. Additionally, the qualitative results for our CVP model on the BAIR dataset can be observed in Fig. 4.

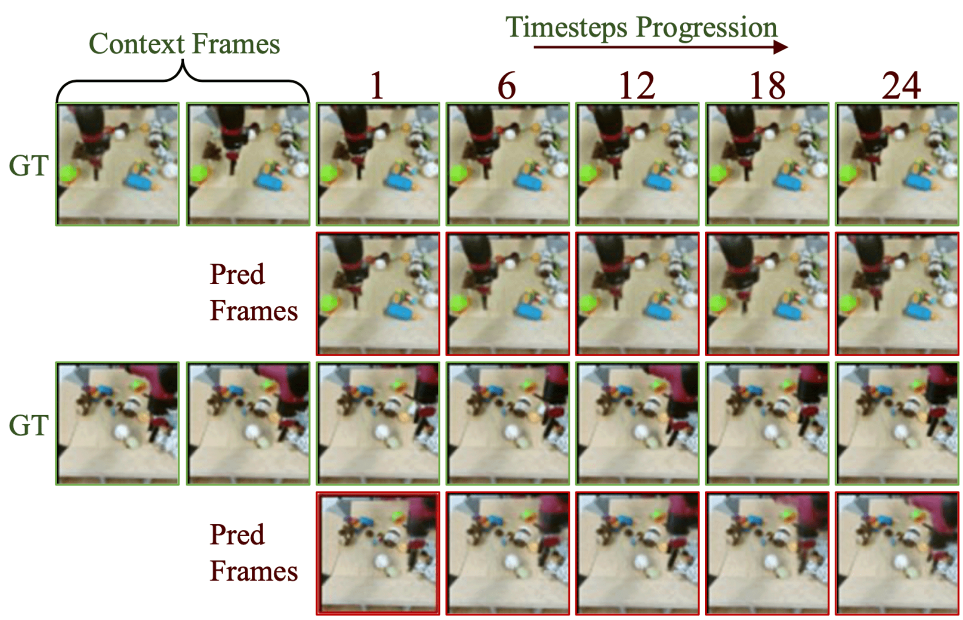

Figure 4. Figure represents qualitative results of our CVP model on the BAIR dataset. The number of context frames used in the above setting is two for both sequences. Every 6^{th} predicted future frame is shown in the figure.

doi.org/10.32388/DM98UZ

BAIR (64×64)	p	k	$\#$pred	**FVD↓**
LVT[53]	1	15	15	125.8
DVD-GAN-FP[32]	1	15	15	109.8
TrIVD-GAN-FP[33]	1	15	15	103.3
VideoGPT[54]	1	15	15	103.3
CCVS[55]	1	15	15	99.0
FitVid[56]	1	15	15	93.6
MCVD[38]	1	5	15	89.5
NÜWA[57]	1	15	15	86.9
RaMViD[41]	1	15	15	84.2
VDM[40]	1	15	15	66.9
RIVER[39]	1	15	15	73.5
CVP (Ours)	1	**1**	15	**70.1**
DVG[1]	2	14	14	120.0
SAVP[31]	2	14	14	116.4
MCVD[38]	2	5	14	87.9
CVP (Ours)	2	2	14	68.2
CVP (Ours)	2	**1**	14	**65.1**
SAVP[31]	2	10	28	143.4
Hier-vRNN[36]	2	10	28	143.4
MCVD[38]	2	5	28	118.4
CVP (Ours)	2	**2**	28	95.1
CVP (Ours)	2	**1**	28	**85.1**

Table 2. *BAIR* dataset evaluation. Video prediction results on BAIR (64×64) conditioning on p past frames and predicting $pred$ frames in the future, using models trained to predict k frames at at time.The common way to

compute the FVD is to compare 100×256 generated sequences to 256 randomly sampled test videos. Best results are marked in ***bold***.

Human3.6M dataset: Similar to the KTH dataset, the Human3.6M dataset features actors performing distinct actions against a static background. However, the Human3.6M dataset distinguishes itself by offering a greater variety of distinct actions within its videos and providing three-channel video frames, in contrast to the single-channel frames of the KTH dataset. For evaluating the Human3.6M dataset, we employed a similar setup to that used for the KTH dataset, where 5 frames are provided as context, and the model predicts the subsequent 30 frames based on these context frames. The results of this evaluation are summarized in Table 3.

An analysis of Table 3 reveals that our model, with its unique approach, requires a significantly lower number of frames for training, needing only a total of 6 frames per block to yield results that are considerably better than those of the baselines.

The results, as presented in Table 3, unequivocally demonstrate that our method outperforms other baseline models, establishing a new state-of-the-art on the Human3.6M dataset. Furthermore, the qualitative efficacy of our CVP model on the Human3.6M dataset is illustrated in Fig. 5, showcasing the model's ability to effectively capture and predict the dataset's varied actions.

doi.org/10.32388/DM98UZ

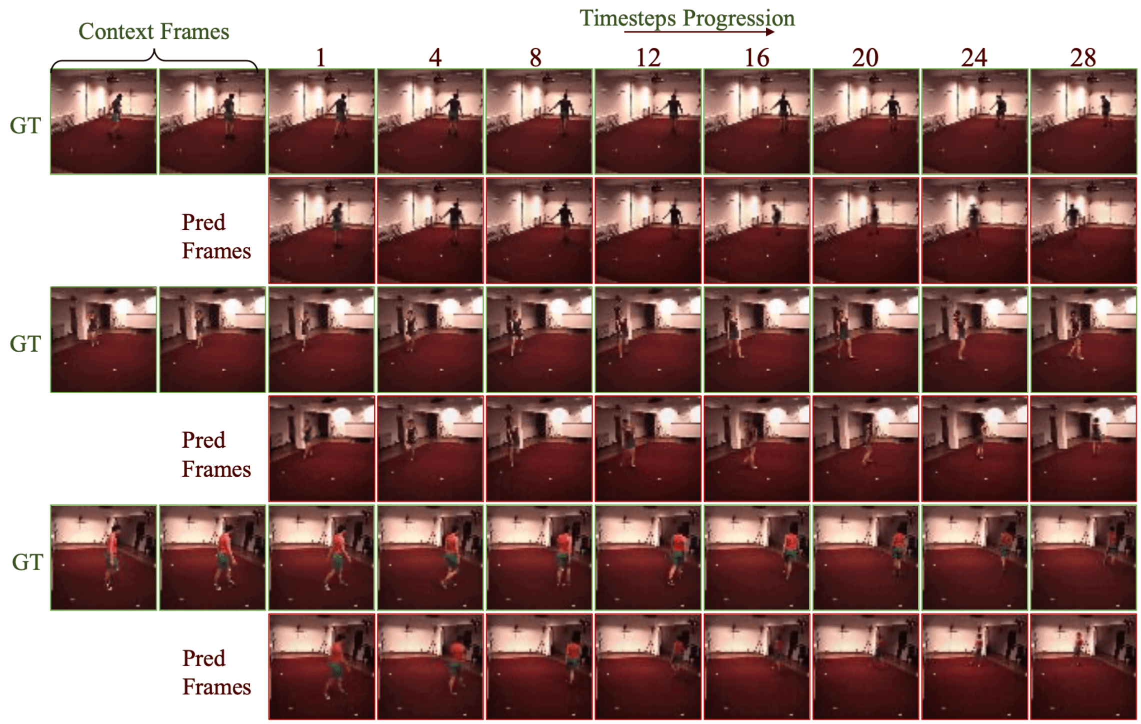

Figure 5. Figure represents qualitative results of our CVP model on the Human3.6M dataset. The number of context frames used in the above setting is 4 for all three sequences. Every 4^{th} predicted future frame is shown in the figure.

Human3.6M	p	k	#pred	FVD↓
SVG-LP[35]	**5**	10	30	718
Struct-VRNN[51]	**5**	10	30	523.4
DVG[1]	**5**	10	30	479.5
SRVP[50]	**5**	10	30	416.5
Grid keypoint[52]	**8**	8	30	166.1
CVP (Ours)	**5**	1	30	**144.5**

Table 3. Quantitative comparisons on the Human3.6M dataset. The best results under each metric are marked in bold.

UCF101 dataset: The UCF101 dataset presents a greater level of complexity compared to the KTH or Human3.6M datasets, owing to its substantially higher number of action categories, diverse backgrounds, and significant camera movements. Notably, we only use information from the context frames for our frame-conditional generation task. No extra information, like class labels, was used for the prediction task. In evaluating the UCF101 dataset, we adopted an approach similar to that used for the Human3.6M dataset, where 5 context frames are provided, and the model is tasked with predicting the next 16 frames based on these. The outcomes of this evaluation are detailed in Table 4.

An examination of Table 4 reveals that our CVP model surpasses the performance of other baseline models, thereby setting a new benchmark for the UCF101 dataset. Additionally, the qualitative performance of our CVP model on the UCF101 dataset is depicted in Fig. 6. This illustration showcases the model's proficiency in accurately capturing and predicting the diverse range of actions featured in the dataset.

doi.org/10.32388/DM98UZ

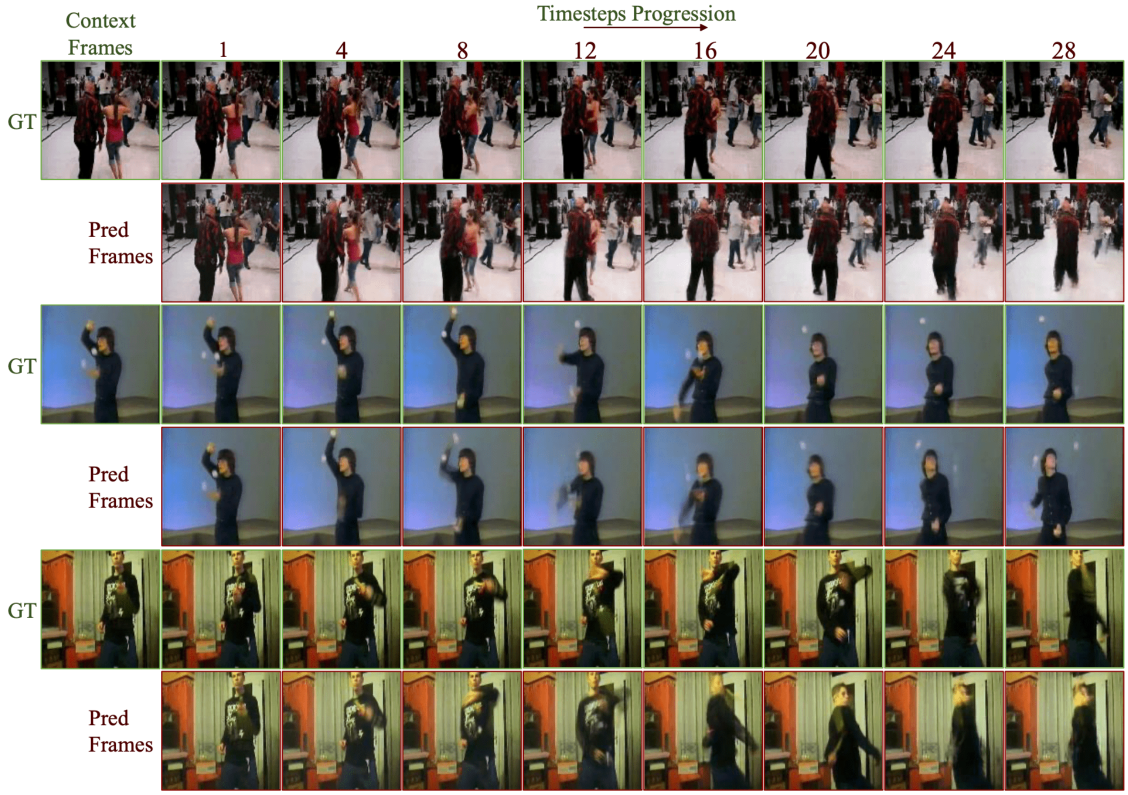

Figure 6. Figure represents qualitative results of our CVP model on the UCF dataset. The number of context frames used in the above setting is 5 for all three sequences. Every 4^{th} predicted future frame is shown in the figure.

UCF101 [$5 \rightarrow 16$]	p	k	#pred	**FVD**↓
SVG-LP[35]	5	10	16	1248
CCVS[55]	5	16	16	409
MCVD[38]	5	5	16	387
RaMViD[41]	5	4	16	356
CVP (Ours)	5	**1**	16	**245.2**

Table 4. Video prediction results on UCF (128×128), predicting 16 frames. All models are conditioned on 5 past frames.

doi.org/10.32388/DM98UZ

6. Ablation Studies

In this section, we present a series of ablation studies conducted to ascertain the impact of various components in our proposed methodology. These studies focus on three primary aspects: the modification of the noise schedule denoted as $g(t)$, the variation in the number of sampling steps, and the exploration of different strategies for sampling the timestep t. Our experimental framework utilizes the KTH dataset for these evaluations.

The outcomes of these experiments are systematically tabulated in Table 6, offering a comprehensive view of the results. The key insights derived from these ablation studies are threefold. Firstly, our analysis underscores the criticality of sampling the timestep t from a uniform square root distribution, specifically $t \sim \sqrt{\mathcal{U}[0,1]}$. This approach appears to significantly influence the model's performance.

Secondly, regarding the noise schedule $g(t)$, we find that the optimal formulation for the task of video prediction is given by $g(t) = \frac{-t\log(t)}{\sqrt{2}}$. This particular noise schedule is characterized by a zero initial and final noise level, with a peak near $t = 0$. Such a configuration is advantageous for our application.

Thirdly, our results, as detailed in Table 6, indicate that an increase in the number of sampling steps beyond 25 does not substantially improve the outcome. Our method outperforms MCVD by producing higher-quality frames in just 25 sampling steps, a 75% reduction compared to its 100 steps. This efficiency is attributed to our CVP method, which retains information from preceding frames, eliminating the need for regeneration from a Gaussian noise vector. Refer to the Table 5 for more details.

In summary, these ablation studies provide valuable insights into the dynamics of our model under varying conditions, highlighting the importance of specific parameter settings and offering guidance for future research directions.

Method	Sampling(Steps/Frame)	Time Taken(hrs)
MCVD	100	2
RaMVID	500	7.2
Ours	**25**	**0.45**

Table 5. Comparison with baselines on sampling steps and sampling time required for BAIR robot push dataset.

doi.org/10.32388/DM98UZ

KTH	Noise Schedule($\sqrt{2}g(t)$)	Sampling steps	t Distrbution	FVD↓
CVP Model Ablations	-	25	$\mathcal{U}[0,1]$	348.2
	$\sin(\pi t)$	25	$\mathcal{U}[0,1]$	278.2
	$\sin(\pi t)$	25	$\sqrt{\mathcal{U}[0,1]}$	237.7
	$t\sin(\pi t)$	25	$\mathcal{U}[0,1]$	240.7
	$t\sin(\pi t)$	25	$\sqrt{\mathcal{U}[0,1]}$	208.4
	$\sqrt{t\,(1-t)}$	25	$\mathcal{U}[0,1]$	209.6
	$\sqrt{t\,(1-t)}$	25	$\sqrt{\mathcal{U}[0,1]}$	187.8
	$-t\log(t)$	25	$\mathcal{U}[0,1]$	190.4
	$-t\log(t)*$	25∗	$\sqrt{\mathcal{U}[0,1]}*$	140.6∗
	$-t\log(t)$	5	$\sqrt{\mathcal{U}[0,1]}$	165.7
	$-t\log(t)$	10	$\sqrt{\mathcal{U}[0,1]}$	144.3
	$-t\log(t)$	50	$\sqrt{\mathcal{U}[0,1]}$	**139.4**

Table 6. Ablation study: Video prediction results on KTH (64×64), predicting 30 frames. All models condition on 4 past frames on 256 test videos. The method with settings marked with $*$ is reported in the main paper.

7. Limitation

While our method demonstrates promising results in video prediction, it is important to acknowledge its limitations to guide future research and application development.

A primary limitation of our approach is its reliance on a limited context frame window for predicting the next frame. Specifically, when a context vector, denoted as $\mathbf{x}^{0:4}$, comprising 4 video frames is used, the prediction of the subsequent frame is entirely dependent on this four-frame window. This model architecture performs adequately in scenarios involving uniform video sequences. However, its efficacy diminishes in a setting that requires more context to predict the future frame. Addressing this limitation

doi.org/10.32388/DM98UZ

requires a more adaptive approach that can handle varying contextual information, a challenge we have earmarked for future research.

Another constraint lies in the computational efficiency of our model. Currently, it necessitates multiple steps to sample a single frame, which could become a significant bottleneck, especially when a larger number of frame predictions are required. Although our method is more efficient in terms of the number of steps needed for frame sampling compared to diffusion-based counterparts, further optimization is necessary to reduce the computational overhead associated with this process.

Additionally, our experimental setup was constrained by the computational resources available to us. The model was developed and tested using just two A6000 GPUs. This limitation raises questions about the potential improvements that could be achieved with a more powerful computational setup. A larger model with an increased number of parameters, trained on more advanced hardware, could potentially unveil further advancements in video prediction capabilities. We recognize this as an important area for investigation and encourage labs with more substantial resources to explore this avenue.

In summary, while our model represents a significant step forward in video prediction, these limitations highlight crucial areas for future research and development, paving the way for more robust and versatile video prediction models.

8. Broader Impact

We used this method for video prediction; however, such modeling can make a major impact on many computational photography tasks. Here, one end of the CVP can be a corrupted image and the other end be a clean ground truth image. Additionally, a larger model with an increased number of parameters, trained on more advanced hardware, could potentially have advanced video prediction capabilities. This can lead to a significant increase in the creation of high-quality artificially generated content, further compounding the problems of fake content. However, a positive contribution of this approach can help with its application in autonomous driving.

9. Conclusion

In this work, we have presented a novel model class designed specifically for video representation, marking a significant advancement in the field of video prediction tasks. Our comprehensive experimental evaluations across various datasets, including KTH, BAIR, Human3.6M, and UCF101, have

doi.org/10.32388/DM98UZ

not only validated the effectiveness of our model but also established new benchmarks in state-of-the-art performance for video prediction tasks.

A notable aspect of our approach is its efficiency in terms of the required number of context and future frames for training. Moreover, our model's continuous video process capability uniquely operates without the need for additional constraints such as temporal attention, which are typically employed to ensure temporal consistency. This aspect of our model underscores its inherent ability to maintain temporal coherence, further simplifying the video prediction process while enhancing its effectiveness.

In conclusion, the innovations introduced in our model offer promising directions for future research in video representation and prediction. The achievements demonstrated in this paper not only contribute to the advancement of video prediction methodologies but also open avenues for exploring more efficient and effective ways of video representation in various real-world applications.

Appendix A. Extended derivations of Eq. (8)

Below is a derivation of Eq. (8), the reduced variance variational bound for our CVP models.

$$
\begin{aligned}
L &= \mathbb{E}_q\left[\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{0:T-1} \mid \mathbf{x}_T\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_0\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_0\right)\right. \\
&\quad \left. - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T\right)} \cdot \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T\right)}{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_T\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_0\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T\right)}\right. \\
&\quad \left. - \log \frac{q\left(\mathbf{x}_T \mid \mathbf{x}_0, \mathbf{x}_T\right)}{q}\left(\mathbf{x}_0 \mid \mathbf{x}_0, \mathbf{x}_T\right)\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_0\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_0\right)\right. \\
&\quad \left. - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T\right)} \cdot \frac{p\left(\mathbf{x}_0 \mid \mathbf{x}_{t-1}\right)}{p\left(\mathbf{x}_0 \mid \mathbf{x}_t\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathbf{x}_0\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T\right)}\right. \\
&\quad \left. - \log \frac{p\left(\mathbf{x}_0 \mid \mathbf{x}_0\right)}{p\left(\mathbf{x}_0 \mid \mathbf{x}_T\right)}\right] \\
&= \mathbb{E}_q\left[-\log \frac{p\left(\mathbf{x}_0\right)}{p\left(\mathbf{x}_0 \mid \mathbf{x}_T\right)} - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T\right)}\right]
\end{aligned}
$$

doi.org/10.32388/DM98UZ

Both $\mathbf{x}_0$ and $\mathbf{x}_T$ are observed variable hence, we ignore the first term in the RHS. We focus on the second term for training the parameters for our CVP models. Therefore, the resulting loss function becomes,

$$L(\theta) =: \sum_{t\geq 1} D_{KL}\left(q\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}, \mathbf{y}\right) \parallel p_\theta\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0\right)\right).$$

Appendix B. Extended derivation for Eq. (2)

Using Eq. (2) we can write the term $\mathbf{x}_{t+\Delta t}$ as follows,

$$\begin{aligned}\mathbf{x}_{t+\Delta t} &= (1-(t+\Delta t))\mathbf{x} + (t+\Delta t)\mathbf{y} \\ &- \frac{(t+\Delta t)\log(t+\Delta t)}{\sqrt{2}}\mathbf{z}_{t+\Delta t}\end{aligned}$$

Considering the term $(t+\Delta t)\log(t+\Delta t)$ we simplify further,

$$(t+\Delta t)\log(t+\Delta t) = t\left(1+\frac{\Delta t}{t}\right)\log t\left(1+\frac{\Delta t}{t}\right). \tag{14}$$

if Δt is infinitesimally small we can write $\left(1+\frac{\Delta t}{t}\right) \to 1$. Using this property we can rewrite $\mathbf{x}_{t+\Delta t}$ as,

$$\mathbf{x}_{t+\Delta t} = (1-(t+\Delta t))\,\mathbf{x} + (t+\Delta t)\,\mathbf{y} - \frac{t\log(t)}{\sqrt{2}}\mathbf{z}_{t+\Delta t} \tag{15}$$

Now, Subtracting $\mathbf{x}_{t+\Delta t}$(Eq. (16)) and $\mathbf{x}_t$(Eq. (1)) we get,

$$\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = (y-x)\,\Delta t - \frac{t\log(t)}{\sqrt{2}}\left(\mathbf{z}_{t+\Delta t} - \mathbf{z}_t\right) \tag{17}$$

Focusing on the term $\left(\mathbf{z}_{t+\Delta t} - \mathbf{z}_t\right)$. Here, $\mathbf{z}_t, \mathbf{z}_{t+\Delta t} \sim \mathcal{N}(0, \mathbf{I})$. Hence, we can write,

$$\left(\mathbf{z}_{t+\Delta t} - \mathbf{z}_t\right) = \sqrt{2}\mathbf{z}\text{where,~}\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{18}$$

Substituting this result back to Eq. (17) we get the following,

$$\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = (y-x)\,\Delta t - t\log(t)\mathbf{z}. \tag{19}$$

Rearranging the terms we get the Eq. (2).

doi.org/10.32388/DM98UZ

Module	Type	Num Inputs	Num Outputs
	Conv2D	$n \times c$	128
	DownBlock2D	128	128
	DownBlock2D	128	128
Encoder	DownBlock2D	128	256
	DownBlock2D	256	256
	AttnDownBlock2D	256	512
	ResnetDownsampleBlock2D	512	512
	ResnetUpsampleBlock2D	512	512
	AttnUpBlock2D	512	512
	UpBlock2D	512	256
Decoder	UpBlock2D	256	256
	UpBlock2D	256	128
	UpBlock2D	128	128
	Conv2d	128	$n \times c$

Table 7. U-NET: We utilize Hugging face diffusers library for our U-Net implementation. We utilize 'positional' type for timestep embeddings. We utilize 4 layers per block. The target resolution for KTH, BAIR and Human3.6M is kept at 64×64 and 128×128 for UCF101 dataset. Additionally, we keep the number of timesteps T as 100 given our compute resources. c denotes the number of channels present in the frame. n is the number of initial context frames based on which next frame is predicted,i.e., $\mathbf{x}^{0:n} \rightarrow \mathbf{x}^{1:n+1}$.

Appendix C. Training Details

For the optimization of our model, we harnessed the compute of two Nvidia A6000 GPUs, each equipped with 48GB of memory, to train our CVP model effectively. We adopted a batch size of 64 and conducted training for a total of 500,000 iterations. To optimize the model parameters, we employed the AdamW optimizer. Additionally, we incorporated a cosine decay schedule for learning rate adjustment, with warm-up steps set at 10,000 iterations. The maximum learning rate (Max LR) utilized during training was 5e-5.

Acknowledgements

This project was partially funded by the NSF CAREER Award (2238769) to AS. We also thank Shirley Huang for providing feedback on the manuscript.

Footnotes

[1] Navigate to the webpage for video results.

References

1. ^a, b, c, d^ *Shrivastava G, Shrivastava A (2021). "Diverse video generation using a Gaussian process trigger". arXiv preprint arXiv:2107.04619. Available from: https://arxiv.org/abs/2107.04619.*
2. ^a, b^ *Bodla N, Shrivastava G, Chellappa R, Shrivastava A (2021). "Hierarchical video prediction using relational layouts for human-object interactions". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 12146–12155.*
3. ^ *Gaurav Shrivastava, Ser-Nam Lim, Abhinav Shrivastava. "Video Dynamics Prior: An Internal Learning Approach for Robust Video Enhancements." In Thirty-seventh Conference on Neural Information Processing Systems, 2023. Available from: https://openreview.net/forum?id=CCq73CGMyV.*
4. ^ *Saini N, He B, Shrivastava G, Rambhatla SS, Shrivastava A. Recognizing actions using object states. In: ICLR 2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality; 2022.*
5. ^ *Shrivastava G, Lim SN, Shrivastava A. Video decomposition prior: Editing videos layer by layer. In: The Twelfth International Conference on Learning Representations; 2024. Available from: https://openreview.net/forum?id=nfMyERXNru.*
6. ^ *Shrivastava G. Diverse Video Generation [PhD thesis]. College Park: University of Maryland, College Park; 2021.*
7. ^ *Shrivastava G. Advanced video modeling techniques for generation and enhancement tasks [PhD thesis]. College Park: University of Maryland; 2024.*
8. ^ *Shrivastava G, Shrivastava A. Video prediction by modeling videos as continuous multi-dimensional processes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024. p. 7236–*

7245.

9. ^a, b^ *Ho J, Jain A, Abbeel P (2020). "Denoising diffusion probabilistic models". Advances in Neural Information Processing Systems.*
10. ^ *Dhariwal P, Nichol A (2021). "Diffusion models beat GANs on image synthesis". Advances in Neural Information Processing Systems.* ***34***.
11. ^ *Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. Score-based generative modeling through stochastic differential equations. International Conference on Learning Representations. 2021. Available from: arXiv:2011.13456.*
12. ^ *Song J, Meng C, Ermon S (2020). "Denoising Diffusion Implicit Models". International Conference on Learning Representations. Available from: arXiv:2010.02502.*
13. ^a, b, c^ *Ionescu C, Li F, Sminchisescu C (2011). "Latent Structured Models for Human Pose Estimation." In: International Conference on Computer Vision, 2011.*
14. ^ *Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09; 2009. Available from: http://www.image-net.org/papers/imagenet_cvpr09.bib.*
15. ^a, b, c^ *Schuldt C, Laptev I, Caputo B. Recognizing human actions: a local SVM approach. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. 2004 Aug;3:32-36 Vol.3. doi:10.1109/ICPR.2004.1334462.*
16. ^a, b, c^ *Soomro K, Zamir AR, Shah M (2012). "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild". arXiv. arXiv:1212.0402.*
17. ^ *Fize J, Shrivastava G, Ménard PA. "Geodict: an integrated gazetteer". In: Proceedings of Language, Ontology, Terminology and Knowledge Structures Workshop (LOTKS 2017); 2017.*
18. ^ *Roche M, Teisseire M, Shrivastava G (2017). "Valorcarn-TETIS: Terms extracted with Biotex". CIRAD Dataverse. 2017.*
19. ^a, b, c^ *Ebert F, Finn C, Lee AX, Levine S (2017). "Self-Supervised Visual Planning with Temporal Skip Connections". arXiv. arXiv:1710.05268 [cs.RO].*
20. ^ *Yuen J, Torralba A. A data-driven approach for event prediction. In: European Conference on Computer Vision. Springer; 2010. p. 707-720.*
21. ^ *Walker J, Gupta A, Hebert M (2014). "Patch to the future: Unsupervised visual prediction." In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 3302–3309.*
22. ^ *Srivastava N, Mansimov E, Salakhutdinov R. Unsupervised Learning of Video Representations Using LSTMs. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning -*

Volume 37. Lille, France: JMLR.org; 2015. p. 843-852. Available from: http://dl.acm.org/citation.cfm?id=3045118.3045209.

23. ^Oliu M, Selva J, Escalera S (2017). "Folded Recurrent Neural Networks for Future Video Prediction". CoRR. **abs/1712.00311**. Available from: http://arxiv.org/abs/1712.00311.
24. ^Cricri F, Ni X, Honkala M, Aksu E, Gabbouj M (2016). "Video Ladder Networks". CoRR. **abs/1612.01756**. Available from: http://arxiv.org/abs/1612.01756. DBLP: https://dblp.org/rec/bib/journals/corr/CricriNHAG16.
25. ^Villegas R, Yang J, Hong S, Lin X, Lee H (2017). "Decomposing motion and content for natural video sequence prediction". CoRR. **abs/1706.08033**. Available from: http://arxiv.org/abs/1706.08033. DBLP: https://dblp.org/rec/bib/journals/corr/VillegasYHLL17.
26. ^Elsayed N, Maida AS, Bayoumi M. Reduced-Gate Convolutional LSTM Architecture for Next-Frame Video Prediction Using Predictive Coding. In: 2019 International Joint Conference on Neural Networks (IJCNN); 2019. p. 1-9. doi:10.1109/IJCNN.2019.8852480.
27. ^Villegas R, Pathak A, Kannan H, Erhan D, Le QV, Lee H (2019). "High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks". arXiv. arXiv:1911.01655 [cs.CV].
28. ^Wang Y, Jiang L, Yang MH, Li LJ, Long M, Fei-Fei L (2019). "Eidetic 3D LSTM: A Model for Video Prediction and Beyond". International Conference on Learning Representations. Available from: https://openreview.net/forum?id=B1lKS2AqtX.
29. ^Castrejón L, Ballas N, Courville AC (2019). "Improved Conditional VRNNs for Video Prediction". CoRR. **abs/1904.12165**. Available from: http://arxiv.org/abs/1904.12165. DBLP: https://dblp.org/rec/bib/journals/corr/abs-1904-12165.
30. ^Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014). "Generative adversarial networks". arXiv. arXiv:1406.2661 [stat.ML].
31. a, b, c, d, e, fLee AX, Zhang R, Ebert F, Abbeel P, Finn C, Levine S (2018). "Stochastic adversarial video prediction". arXiv preprint arXiv:1804.01523.
32. a, bClark A, Donahue J, Simonyan K (2019). "Adversarial video generation on complex datasets". arXiv preprint arXiv:1907.06571. Available from: https://arxiv.org/abs/1907.06571.
33. a, bLuc P, Clark A, Dieleman S, de Las Casas D, Doron Y, Cassirer A, Simonyan K (2020). "Transformation-based adversarial video prediction on large-scale data". arXiv preprint arXiv:2003.04035.
34. ^Kingma DP, Welling M (2013). "Auto-Encoding Variational Bayes". CoRR. **abs/1312.6114**.
35. a, b, c, d, eDenton E, Fergus R (2018). "Stochastic Video Generation with a Learned Prior". arXiv. arXiv:1802.07687 [cs.CV].

36. ^a, b^ *Castrejon L, Ballas N, Courville A (2019). "Improved conditional vrnns for video prediction". Proceedings of the IEEE/CVF international conference on computer vision. 7608--7617.*

37. ^ *Bhagat S, Uppal S, Yin Z, Lim N (2020). "Disentangling Multiple Features in Video Sequences using Gaussian Processes in Variational Autoencoders". arXiv. arXiv:2001.02408.*

38. ^a, b, c, d, e, f, g, h, i^ *Voleti V, Jolicoeur-Martineau A, Pal C (2022). "MCVD-masked conditional video diffusion for prediction, generation, and interpolation". Advances in Neural Information Processing Systems.* **35***: 23371–23385.*

39. ^a, b, c, d^ *Davtyan A, Sameni S, Favaro P (2023). "Efficient Video Prediction via Sparsely Conditioned Flow Matching". In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 23263–23274.*

40. ^a, b^ *Ho J, Salimans T, Gritsenko A, Chan W, Norouzi M, Fleet DJ (2022). "Video diffusion models". arXiv. arXiv:2204.03458.*

41. ^a, b, c^ *Höppe T, Mehrjou A, Bauer S, Nielsen D, Dittadi A (2022). "Diffusion Models for Video Prediction and Infilling". arXiv. arXiv:2206.07696 [cs.CV].*

42. ^ *Walker J, Marino K, Gupta A, Hebert M. "The Pose Knows: Video Forecasting by Generating Pose Futures". In: International Conference on Computer Vision; 2017.*

43. ^ *Villegas R, Yang J, Zou Y, Sohn S, Lin X, Lee H (2017). "Learning to Generate Long-term Future via Hierarchical Prediction". arXiv. arXiv:1704.05831 [cs.CV].*

44. ^ *Wichers N, Villegas R, Erhan D, Lee H (2018). "Hierarchical Long-term Video Prediction without Supervision". arXiv. arXiv:1806.04768.*

45. ^ *Cai H, Bai C, Tai YW, Tang CK (2018). "Deep video generation, prediction and completion of human action sequences". Lecture Notes in Computer Science. page 374–390.*

46. ^ *Delbracio M, Milanfar P (2023). "Inversion by direct iteration: An alternative to denoising diffusion for image restoration". arXiv preprint arXiv:2303.11435. Available from: https://arxiv.org/abs/2303.11435.*

47. ^ *Bansal A, Borgnia E, Chu HM, Li JS, Kazemi H, Huang F, Goldblum M, Geiping J, Goldstein T (2022). "Cold diffusion: Inverting arbitrary image transforms without noise". arXiv preprint arXiv:2208.09392.*

48. ^ *Unterthiner T, van Steenkiste S, Kurach K, Marinier R, Michalski M, Gelly S (2018). "Towards Accurate Generative Models of Video: A New Metric and Challenges". arXiv. arXiv:1812.01717.*

49. ^ *Akan AK, Erdem E, Erdem A, Güney F (2021). "SLAMP: Stochastic Latent Appearance and Motion Prediction". Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 14728--14737.*

50. ^a, b^ *Franceschi JY, Delasalles E, Chen M, Lamprier S, Gallinari P. Stochastic latent residual video prediction. In: International Conference on Machine Learning. PMLR; 2020. p. 3233–3246.*

 doi.org/10.32388/DM98UZ

51. ^[a, b] *Minderer M, Sun C, Villegas R, Cole F, Murphy KP, Lee H (2019). "Unsupervised learning of object structure and dynamics from videos". Advances in Neural Information Processing Systems.* ***32****.*

52. ^[a, b] *Gao X, Jin Y, Dou Q, Fu CW, Heng PA. Accurate grid keypoint learning for efficient video prediction. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2021. p. 5908-5915.*

53. ^ *Rakhimov R, Volkhonskiy D, Artemov A, Zorin D, Burnaev E (2020). "Latent video transformer". arXiv preprint arXiv:2006.10704.*

54. ^ *Yan W, Zhang Y, Abbeel P, Srinivas A (2021). "Videogpt: Video generation using vq-vae and transformers". arXiv preprint arXiv:2104.10157.*

55. ^[a, b] *Le Moing G, Ponce J, Schmid C (2021). "CCVS: context-aware controllable video synthesis". Advances in Neural Information Processing Systems.* ***34****: 14042–14055.*

56. ^ *Babaeizadeh M, Saffar MT, Nair S, Levine S, Finn C, Erhan D (2021). "Fitvid: Overfitting in pixel-level video prediction". arXiv preprint arXiv:2106.13195.*

57. ^ *Liang J, Wu C, Hu X, Gan Z, Wang J, Wang L, Liu Z, Fang Y, Duan N (2022). "NUWA-infinity: Autoregressive over autoregressive generation for infinite visual synthesis". Advances in Neural Information Processing Systems.* ***35****: 15420–15432.*

Declarations

Funding: This project was partially funded by the NSF CAREER Award (2238769) to AS. We also thank Shirley Huang for providing feedback on the manuscript.

Potential competing interests: No potential competing interests to declare.

 doi.org/10.32388/DM98UZ