Video Prediction by Modeling Videos as Continuous Multi-Dimensional Processes

Supplementary Material

A. Extended derivations of Eq. (8)

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

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

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,

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

B. Extended derivation for Eq. (2)

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

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

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,

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

$$
\tag{16}
$$

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

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

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

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

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).

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.

Table 5. **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., $\mathbf{x}^{0:n} \rightarrow \mathbf{x}^{1:n+1}$.

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

D. 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. 7, 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 7, 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 6 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.

Table 6. Comparison with baselines on sampling steps and sampling time required for BAIR robot push dataset.

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

Table 7. **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.

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**

E. 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.