MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation

Introduction

- SOTA challenging -

- the quality of video frames from generative models tends to be poor
- generalization beyond the training data is difficult
- not capable of simultaneously handling other video-related tasks
    - such as unconditional generation or interpolation

- objective -

- devise a video generation approach that **generates high-quality**, **time-consistent videos** , with computation times for training models measured in 1-12 days using ≤ 4 GPUs

Introduction

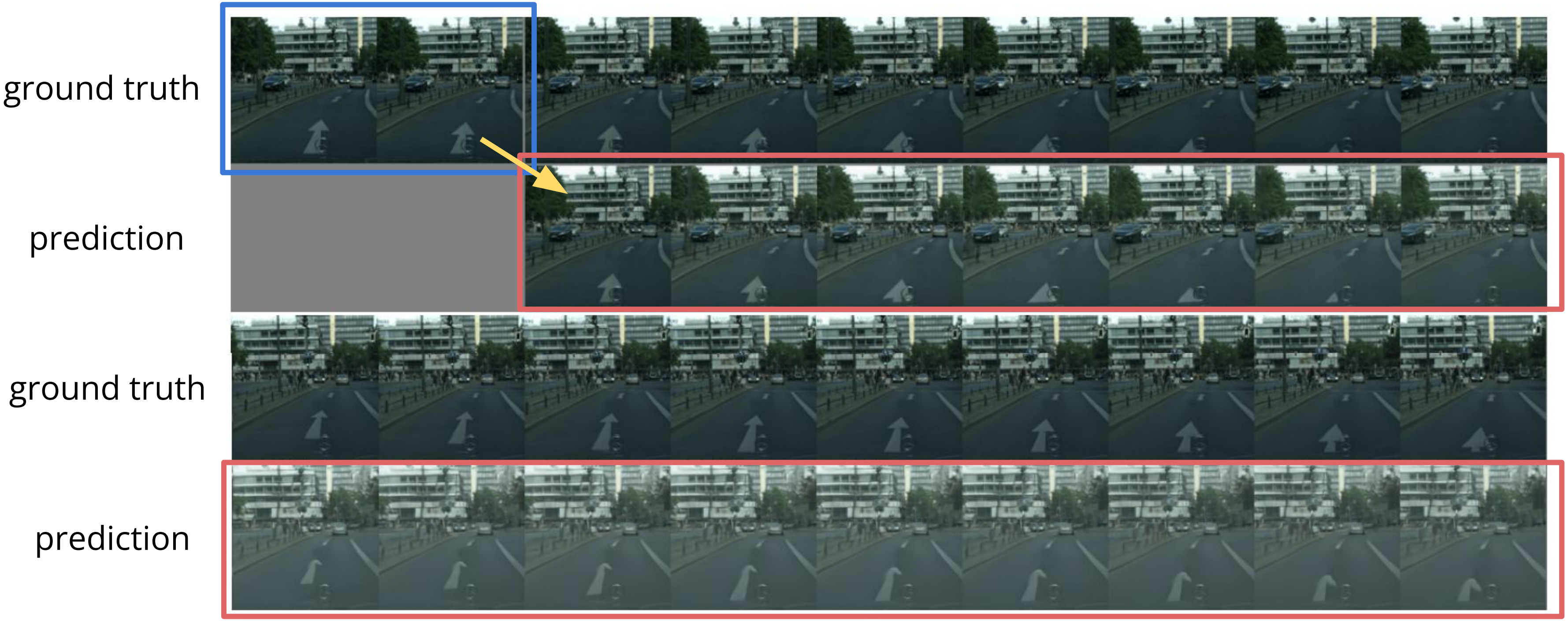
- general purpose framework with **Masked Conditional Video Diffusion (MCVD) models**
- using a probabilistic conditional **score-based denoising diffusion model**, conditioned on past and/or future frames
- models are built from simple non-recurrent 2D-convolutional architectures, conditioning on <u>blocks of frames</u>

Introduction

- video tasks -

- future / past prediction
- unconditional generation
- interpolation

Introduction



Related work

- Diffusion Model Family
  - Denoising Diffusion Probabilistic Models
  - Score-based Generative Models diverse data samples
- drawbacks
  - solving the reverse process is relatively slow

Conditional Diffusion - FDP

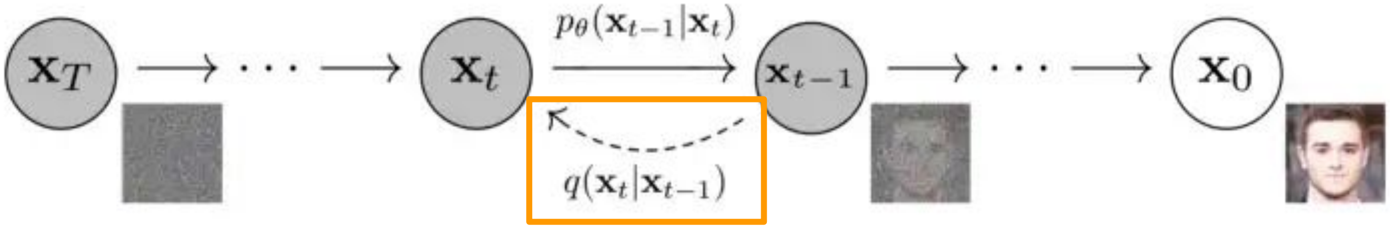
$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \xrightarrow{\text{推导}} p(\boldsymbol{x}_t|\boldsymbol{x}_0) \xrightarrow{\text{推导}} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \xrightarrow{\text{近似}} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

- Forward Diffusion Process -

- Transition kernel : $q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}),$
- Accumulated kernel :

$$q_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \implies \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.



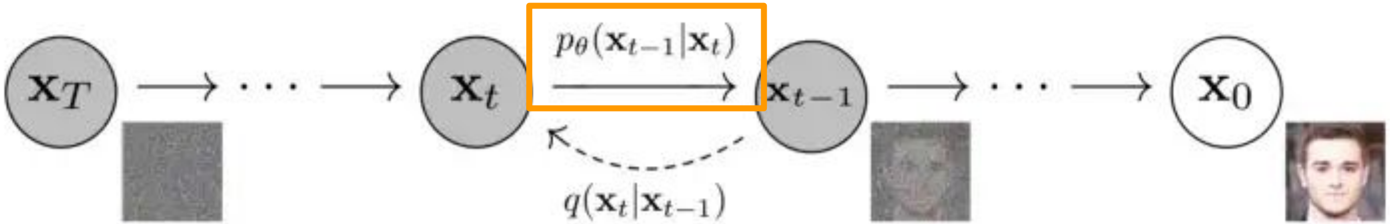
Conditional Diffusion - RDP

- Reverse Diffusion Process -

- Transition kernel :

$$p_t(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}),$$

$$\text{where} \quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \quad \text{and} \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$



Conditional Diffusion - Loss Function

- Loss function -

$$L(\theta) = \mathbb{E}_{t,\mathbf{x}_0 \sim p_{\text{data}},\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \mid t) \right\|_2^2 \right]$$

- Score function -

- definition : $\nabla_{\mathbf{x}} \log p(\mathbf{x}),$
- $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid \mathbf{x}_0) = -\frac{1}{1-\bar{\alpha}_t}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}$

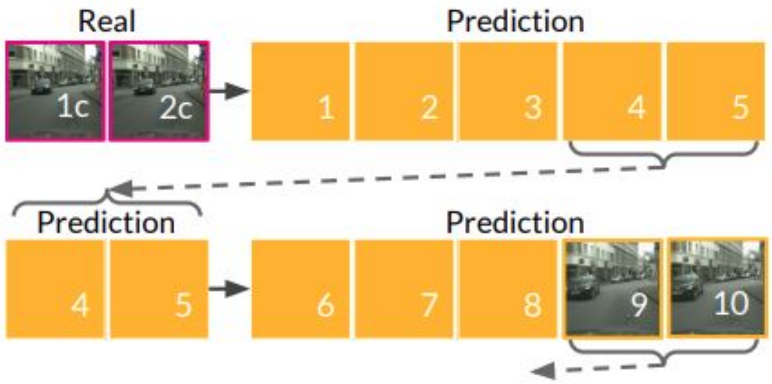
Conditional Diffusion for Video

- Score-based diffusion models **can be straightforwardly adapted to video** by considering the joint distribution of multiple continuous frame
- sufficient for unconditional video generation, other tasks such as video interpolation and prediction remain unsolved

Video Prediction

- p past frames : $p = \{p^i\}_{i=1}^{p}$
- k current frames (in immediate future) : $x_0 = \{x_0^i\}_{i=1}^{k}$
- loss function :

$$L_{\text{vidpred}}(\theta) = \mathbb{E}_{t,[\mathbf{p},\mathbf{x}_0]\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \mid \mathbf{p}, t)\right\|^2\right]$$



Video Prediction + Generation

- extend the same framework to **unconditional video generation**
- **masking (zeroing-out) the past frames** with probability $p_{mask} = 1/2$ using binary mask m_p
- loss function :

$$L_{\text{vidgen}}(\theta) = \mathbb{E}_{t,[\mathbf{p},\mathbf{x}_0]\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),m_p\sim\mathcal{B}(p_{\text{mask}})}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\mid m_p\mathbf{p},t)\right\|^2\right]$$

- improving the model's ability to perform predictions conditioned on the past
  - learns to predict the noise added without any past frames for context
- we can perform **conditional as well as unconditional** frame generation

Video Prediction + Generation + Interploation

- p past frames : $p = \{p^i\}_{i=1}^{p}$
- k current frames (in immediate future) : $x_0 = \{x_0^i\}_{i=1}^{k}$
- f future frames : $f = \{f^i\}_{i=1}^{f}$
- loss function :

$$L(\theta) = \mathbb{E}_{t,[\mathbf{p},\mathbf{x}_0,\mathbf{f}]\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),(m_p,m_f)\sim\mathcal{B}(p_{\text{mask}})}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \mid m_p\mathbf{p}, m_f\mathbf{f}, t)\right\|^2\right]$$

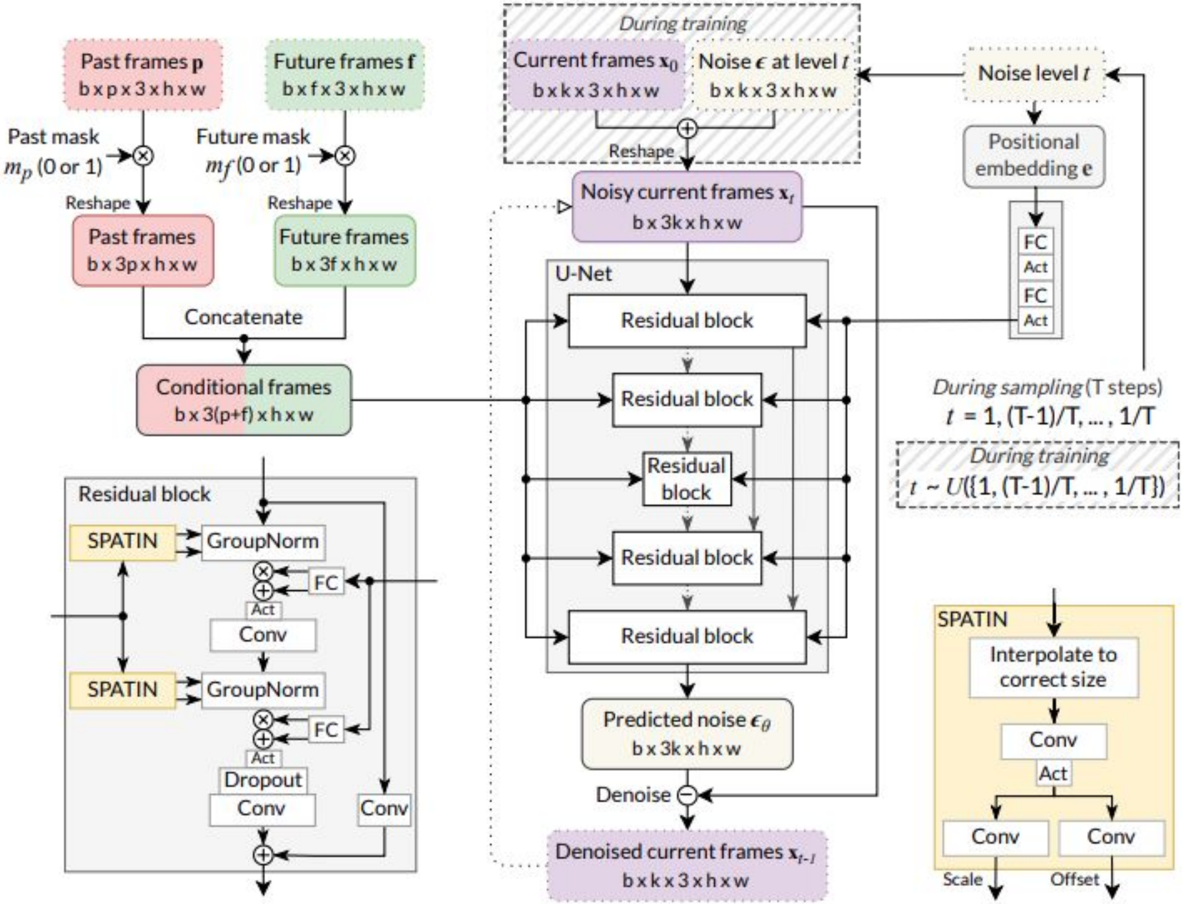
- randomly <u>mask the p past frames</u> with probability $p_{mask} = 1/2$, and similarly randomly <u>mask the f future frames</u> with the same probability (but sampled separately)

Video Prediction + Generation + Interploation

- masked -

- future prediction : only future frames are masked
- past prediction : only past frames are masked
- unconditional generation : past and future are masked
- video interpolation : no masked

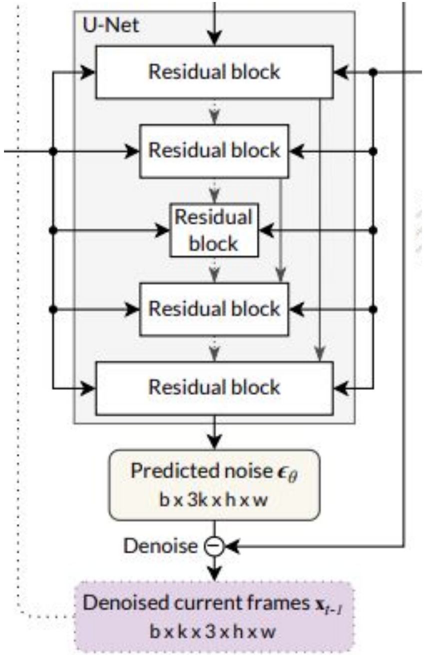
Architecture



Architecture

- Denoising Network -

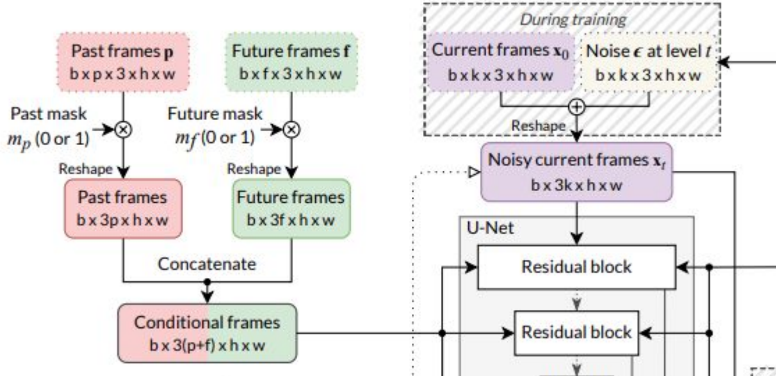
- U-net architecture combining the improvements
- this architecture use mix of 2D convolutions , multi-head self-attention, and adaptive group-norm
- use positional encodings of the noise level (t ∈ [0, 1]) and process it using a transformer style positional embedding :



$$\mathbf{e}(t) = \left[\ldots, \cos\left(tc^{\frac{-2d}{D}}\right), \sin\left(tc^{\frac{-2d}{D}}\right), \ldots\right]^{\mathrm{T}}$$

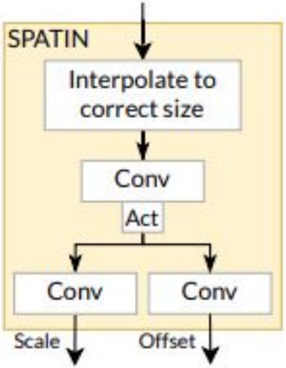
where $d = 1, \ldots, D/2$, D is the number of dimensions of the embedding, and $c = 10000$.

Architecture - Normalization



- SPAce-TIme-Adaptive Normalization (SPATIN) -

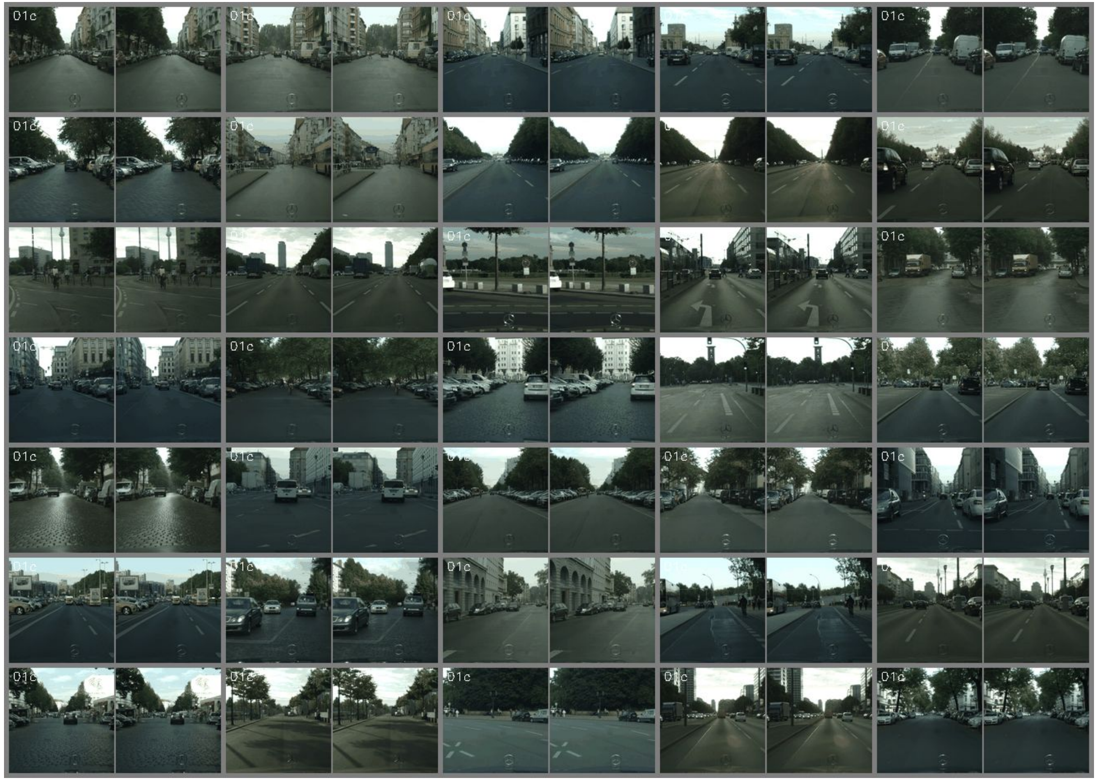
- noisy current frames (x_t)
  - passed directly to the network
- concatenated conditional frames
  - concatenate past (p) / future (f) conditional frames
  - passed through an embedding that influences the conditional normalization



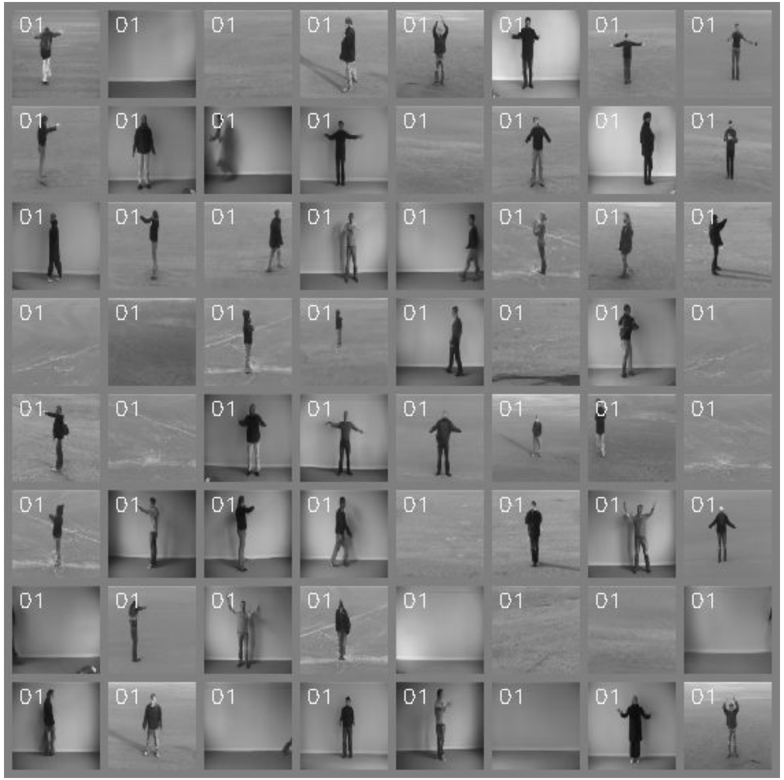
- Concat -

directly concatenating the conditional frames and noisy current frames together and passing them as the input

Results



Results



Results

Experiments - Dataset

in order of progressive difficulty :

1. SMMNIST : black-and-white digits
2. KTH : grayscale, single-humans
3. BAIR : color, multiple objects, simple scene
4. Cityscapes : color, natural complex, natural driving scene
5. UCF101 : color, 101 categories of natural scenes

Experiments

- sampling method : DDPM, DDIM
  - DDPM is better
- predict only **4-5 current frames at a time**, then autoregressively predict longer sequences for prediction or generation
  - to fit GPU memory budget
  - perform better than other models

Experiments - Metrics

- FVD (Fréchet Video Distance)
- PSNR :

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

- SSIM :

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

$$SSIM(x,y) = [l(x,y)^\alpha \cdot c(x,y)^\beta \cdot s(x,y)^\gamma]$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Experiments

SMMNIST [5 → 10; trained on k]	k	**FVD**↓	**SSIM**↑
SVG [Denton and Fergus, 2018]	10	90.81	0.688
vRNN 1L [Castrejón et al., 2019]	10	63.81	0.763
Hier-vRNN [Castrejón et al., 2019]	10	57.17	0.760
MCVD concat (Ours)	**5**	25.63	**0.786**
MCVD spatin (Ours)	**5**	**23.86**	0.780

KTH [10 → *pred*; trained on k]	k	*pred*	FVD↓	PSNR↑	SSIM↑
SAVP [Lee et al., 2018]	10	30	374 ± 3	26.5	0.756
MCVD concat (Ours)	**5**	30	323 ± 3	27.5	0.835
SLAMP [Akan et al., 2021]	10	30	228 ± 5	29.4	0.865
SRVP [Franceschi et al., 2020]	10	30	$\mathbf{222} \pm 3$	**29.7**	**0.870**
MCVD concat (Ours)	**5**	40	276.7	26.40	0.812
SAVP-VAE [Lee et al., 2018]	10	40	145.7	26.00	0.806
Grid-keypoints [Gao et al., 2021]	10	40	**144.2**	**27.11**	**0.837**

Experiments

- **Video prediction**

Table 3: 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.

BAIR (64×64) [past $p \rightarrow pred$; trained on k]	p	k	$pred$	FVD↓	PSNR↑	SSIM↑
LVT [Rakhimov et al., 2020]	1	15	15	125.8	–	–
DVD-GAN-FP [Clark et al., 2019]	1	15	15	109.8	–	–
MCVD spatin (Ours)	1	**5**	15	103.8	18.8	0.826
TrIVD-GAN-FP [Luc et al., 2020]	1	15	15	103.3	–	–
VideoGPT [Yan et al., 2021]	1	15	15	103.3	–	–
CCVS [Le Moing et al., 2021]	1	15	15	99.0	–	–
MCVD concat (Ours)	1	**5**	15	98.8	18.8	0.829
MCVD spatin past-mask (Ours)	1	**5**	15	96.5	18.8	0.828
MCVD concat past-mask (Ours)	1	**5**	15	95.6	18.8	**0.832**
Video Transformer [Weissenborn et al., 2019]	1	15	15	94-96[a]	–	–
FitVid [Babaeizadeh et al., 2021]	1	15	15	93.6	–	–
MCVD concat past-future-mask (Ours)	1	**5**	15	**89.5**	16.9	0.780
SAVP [Lee et al., 2018]	2	14	14	116.4	–	–
MCVD spatin (Ours)	2	**5**	14	94.1	19.1	0.836
MCVD spatin past-mask (Ours)	2	**5**	14	90.5	**19.2**	0.837
MCVD concat (Ours)	2	**5**	14	90.5	19.1	0.834
MCVD concat past-future-mask (Ours)	2	**5**	14	89.6	17.1	0.787
MCVD concat past-mask (Ours)	2	**5**	14	**87.9**	19.1	**0.838**
SAVP [Lee et al., 2018]	2	10	28	143.4	–	0.795
Hier-vRNN [Castrejón et al., 2019]	2	10	28	143.4	–	**0.822**
MCVD spatin (Ours)	2	**5**	28	132.1	17.5	0.779
MCVD spatin past-mask (Ours)	2	**5**	28	127.9	17.7	0.789
MCVD concat (Ours)	2	**5**	28	120.6	17.6	0.785
MCVD concat past-mask (Ours)	2	**5**	28	119.0	**17.7**	0.797
MCVD concat past-future-mask (Ours)	2	**5**	28	**118.4**	16.2	0.745

[a] 94 on only the first frames, 96 on all subsequences of test frames

Experiments

- **Video prediction**

Table 3: 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.

BAIR (64×64) [past $p \rightarrow pred$; trained on k]	p	k	$pred$	FVD↓	PSNR↑	SSIM↑
LVT [Rakhimov et al., 2020]	1	15	15	125.8	–	–
DVD-GAN-FP [Clark et al., 2019]	1	15	15	109.8	–	–
MCVD spatin (Ours)	1	**5**	15	103.8	18.8	0.826
TrIVD-GAN-FP [Luc et al., 2020]	1	15	15	103.3	–	–
VideoGPT [Yan et al., 2021]	1	15	15	103.3	–	–
CCVS [Le Moing et al., 2021]	1	15	15	99.0	–	–
MCVD concat (Ours)	1	**5**	15	98.8	18.8	0.829
MCVD spatin past-mask (Ours)	1	**5**	15	96.5	18.8	0.828
MCVD concat past-mask (Ours)	1	**5**	15	95.6	18.8	**0.832**
Video Transformer [Weissenborn et al., 2019]	1	15	15	94-96[a]	–	–
FitVid [Babaeizadeh et al., 2021]	1	15	15	93.6	–	–
MCVD concat past-future-mask (Ours)	1	**5**	15	**89.5**	16.9	0.780
SAVP [Lee et al., 2018]	2	14	14	116.4	–	–
MCVD spatin (Ours)	2	**5**	14	94.1	19.1	0.836
MCVD spatin past-mask (Ours)	2	**5**	14	90.5	**19.2**	0.837
MCVD concat (Ours)	2	**5**	14	90.5	19.1	0.834
MCVD concat past-future-mask (Ours)	2	**5**	14	89.6	17.1	0.787
MCVD concat past-mask (Ours)	2	**5**	14	**87.9**	19.1	**0.838**
SAVP [Lee et al., 2018]	2	10	28	143.4	–	0.795
Hier-vRNN [Castrejón et al., 2019]	2	10	28	143.4	–	**0.822**
MCVD spatin (Ours)	2	**5**	28	132.1	17.5	0.779
MCVD spatin past-mask (Ours)	2	**5**	28	127.9	17.7	0.789
MCVD concat (Ours)	2	**5**	28	120.6	17.6	0.785
MCVD concat past-mask (Ours)	2	**5**	28	119.0	**17.7**	0.797
MCVD concat past-future-mask (Ours)	2	**5**	28	**118.4**	16.2	0.745

[a] 94 on only the first frames, 96 on all subsequences of test frames

Experiments

- **unconditional**

Table 5: Unconditional generation of BAIR video frames.

BAIR (64 × 64) [0 → *pred*; trained on 5]	*pred*	FVD↓
MCVD spatin past-mask (Ours)	16	267.8
MCVD concat past-mask (Ours)	16	**228.5**
MCVD spatin past-mask (Ours)	30	399.8
MCVD concat past-mask (Ours)	30	**348.2**

Table 6: Unconditional generation of UCF-101 video frames.

UCF-101 (64 × 64) [0 → 16; trained on *k*]	*k*	FVD↓
MoCoGAN-MDP [Yushchenko et al., 2019]	16	1277.0
MCVD concat past-mask (Ours)	**4**	1228.3
TGANv2 [Saito et al., 2020]	16	1209.0
MCVD spatin past-mask (Ours)	**4**	1143.0
DIGAN [Yu et al., 2022]	16	**655.0**

Experiments

- **Video Interpolation**

Table 7: Video Interpolation results (64 × 64). Given p past + f future frames → interpolate k frames. Reporting average of the best metrics out of n trajectories per test sample. $\downarrow(p+f)$ and $\uparrow k$ is harder. We used MCVD spatin past-mask for SMMNIST and KTH, and MCVD concat past-future-mask for BAIR. We also include results on SMMNIST for a "pure" model trained without any masking.

	SMMNIST (64 × 64)					**KTH** (64 × 64)					**BAIR** (64 × 64)				
	$p+f$	k	n	PSNR↑	SSIM↑	$p+f$	k	n	PSNR↑	SSIM↑	$p+f$	k	n	PSNR↑	SSIM↑
SVG-LP Denton and Fergus [2018]	18	7	100	13.543	0.741	18	7	100	28.131	0.883	18	7	100	18.648	0.846
FSTN Lu et al. [2017]	18	7	100	14.730	0.765	18	7	100	29.431	0.899	18	7	100	19.908	0.850
SepConv Niklaus et al. [2017]	18	7	100	14.759	0.775	18	7	100	29.210	0.904	18	7	100	21.615	0.877
SuperSloMo Jiang et al. [2018]	18	7	100	13.387	0.749	18	7	100	28.756	0.893	–	–	–	–	–
SDVI full Xu et al. [2020]	18	7	100	16.025	0.842	18	7	100	29.190	0.901	18	7	100	21.432	0.880
SDVI Xu et al. [2020]	16	7	100	14.857	0.782	16	7	100	26.907	0.831	16	7	100	19.694	0.852
MCVD (Ours)	**10**	**10**	100	20.944	0.854	**15**	**10**	100	34.669	0.943	**4**	**5**	100	25.162	0.932
	10	**5**	**10**	27.693	0.941	**15**	**10**	**10**	34.068	0.942	**4**	**5**	**10**	23.408	0.914
	pure			18.385	0.802	**10**	**5**	**10**	35.611	0.963					

Conclusion

1. A conditional video diffusion approach for video prediction and interpolation that yields SOTA results.
2. A conditioning procedure based on masking past and/or future frames in a blockwise manner giving a single model the ability to solve multiple video tasks : **future/past prediction**, **unconditional generation**, and **interpolation**.
3. A convolutional U-net neural architecture integrating recent developments with a conditional normalization technique we call **SPAce-TIme-Adaptive Normalization**.

Limitations

- become blurry or inconsistent when the number of generated frames is very large
  - needed to scale these models to larger datasets with more diversity and with longer duration video (this work limited by 4-GPU)
  - need for faster sampling methods capable of maintaining quality over time