August 12, 2022

Solving Video Tasks using Denoising Diffusion Models

arxiv.org/abs/2205.09853

mask-cond-video-diffusion.github.io

Vikram Voleti

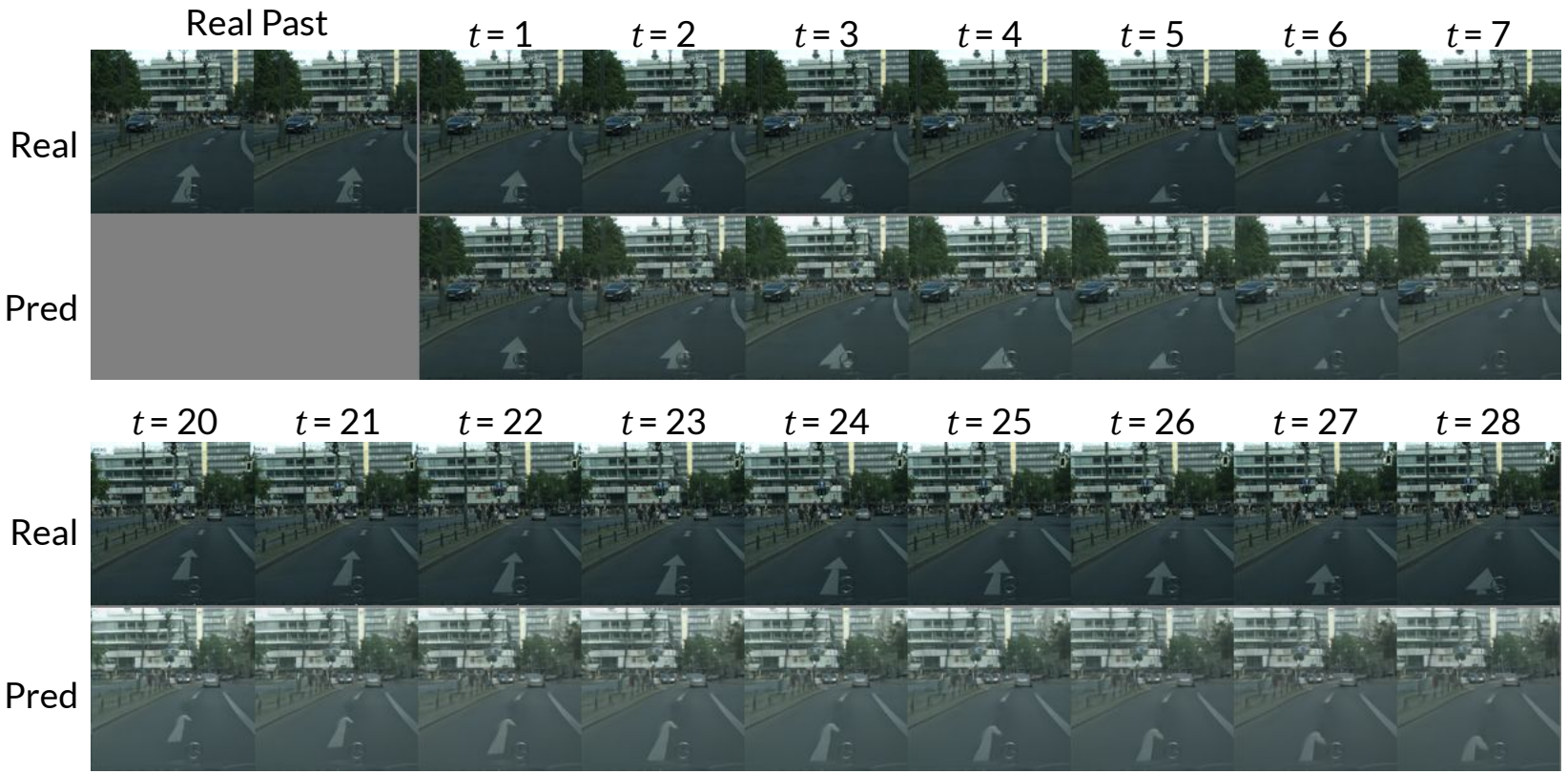
PhD student, Mila, University of Montreal

Supervisor: **Prof. Christopher Pal**

voletiv.github.io

@ (virtual) Samsung AI Center, Toronto, Canada

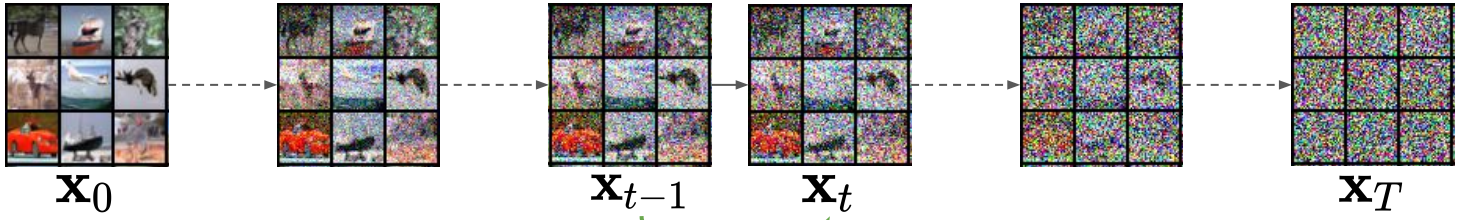
MCVD: Masked Conditional Video Diffusion



1. **Score-based Denoising Diffusion Models**

2. MCVD: Masked Conditional Video Diffusion

Forward Process (DDPM)



$$q_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

$\mathbf{x}_0$ $\mathbf{x}_t$

$$q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad ; \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$$

$$\implies \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \quad ; \epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})$$

arxiv.org/abs/2006.11239

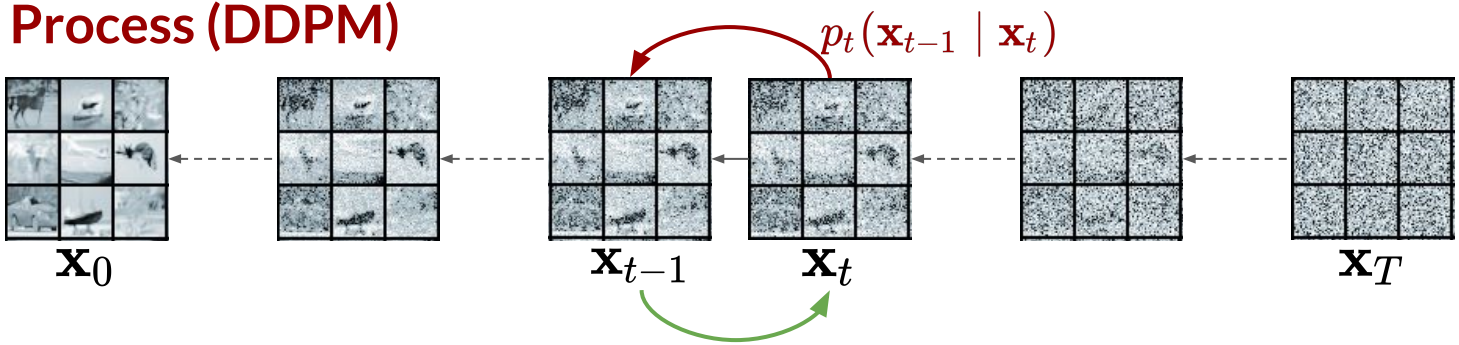
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

(1) $$\mathbf{x}_0 = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon)/\sqrt{\bar{\alpha}_t}$$

(2) $$\epsilon = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\right)$$

arxiv.org/abs/2006.11239

Reverse Process (DDPM)

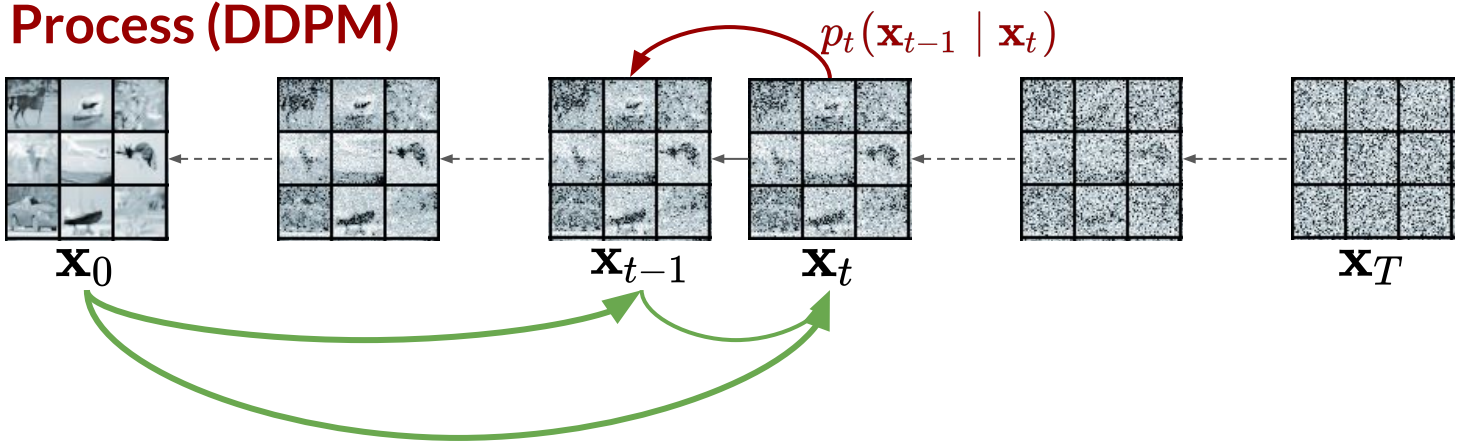


$$p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \frac{q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \checkmark \; p(\mathbf{x}_{t-1})?}{p(\mathbf{x}_t)?}$$

(Bayes' theorem)

arxiv.org/abs/2006.11239

Reverse Process (DDPM)

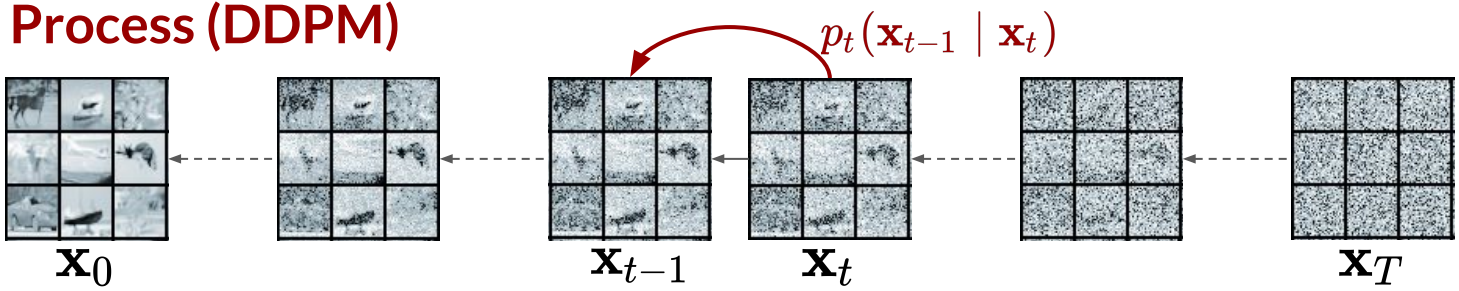


$$p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \frac{q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \; q_t(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q_t(\mathbf{x}_t|\mathbf{x}_0)}$$

(Condition on $\mathbf{x}_0$)

arxiv.org/abs/2006.11239

Reverse Process (DDPM)



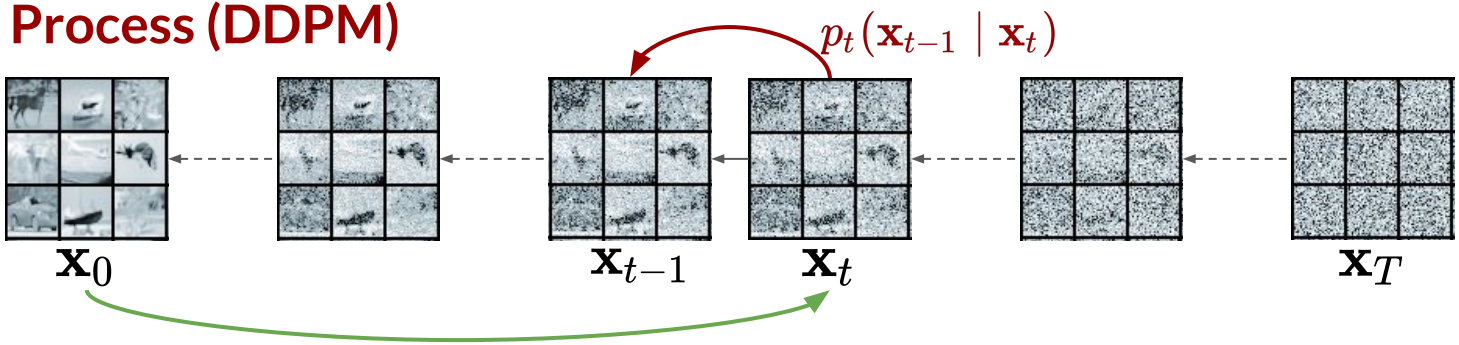
$p(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \mathbf{mu}, \mathbf{\Lambda}^{-1})$,
$p(\mathbf{v} \mid \mathbf{u}) = \mathcal{N}(\mathbf{v} \mid \mathbf{A}\mathbf{u} + \mathbf{b}, \mathbf{L}^{-1})$
$\Rightarrow p(\mathbf{v}) = \mathcal{N}(\mathbf{v} \mid \mathbf{A}\mu + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{A}^T)$,
$\Rightarrow p(\mathbf{u} \mid \mathbf{v}) = \mathcal{N}(\mathbf{u} \mid \mathbf{C}(\mathbf{A}^T\mathbf{L}(\mathbf{v} - \mathbf{b}) + \mathbf{\Lambda}\mathbf{mu}), \mathbf{C})$
$[\mathbf{C} = (\mathbf{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}]$
where $\mathbf{u} = \mathbf{x}_{t-1} \mid \mathbf{x}_0$, $\mathbf{v} = \mathbf{x}_t$

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

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

arxiv.org/abs/2006.11239

Reverse Process (DDPM)



1. $\hat{\mathbf{x}}_0 = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\ \epsilon\ )/\sqrt{\bar{\alpha}_t}$ (?)

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

From (1):

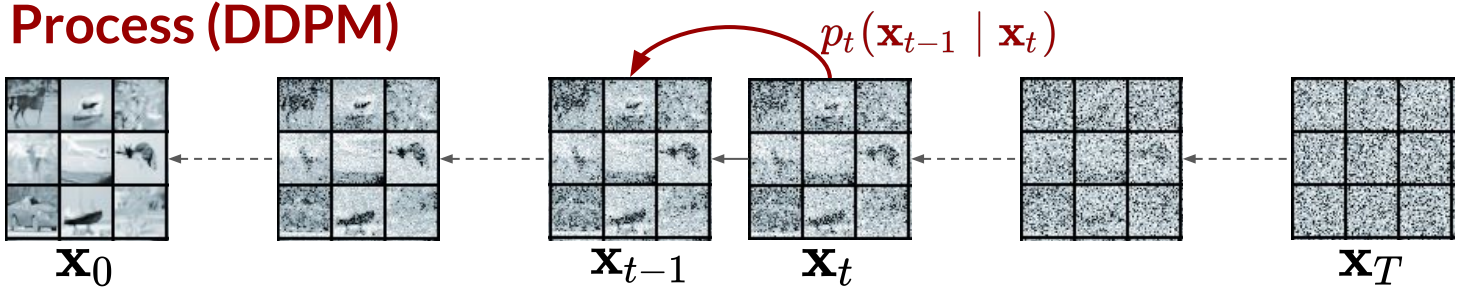
$$q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

$$\text{(1)}\quad \mathbf{x}_0 = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon)/\sqrt{\bar{\alpha}_t}$$

arxiv.org/abs/2006.11239

Reverse Process (DDPM)



Deep neural network!

1. $\hat{\mathbf{x}}_0 = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t))/\sqrt{\bar{\alpha}_t}$
2. $p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \hat{\mathbf{x}}_0), \tilde{\beta}_t \mathbf{I})$

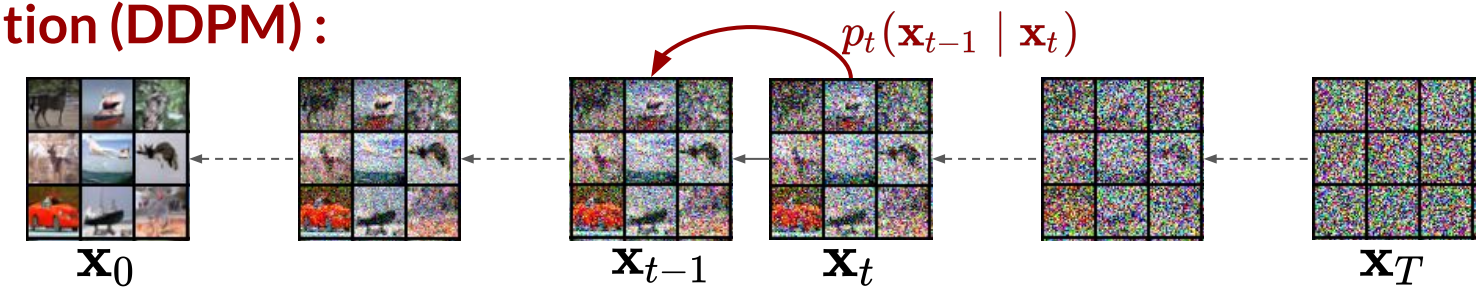
arxiv.org/abs/2006.11239

Objective function:

$$\mathbb{E}_{t,\ \mathbf{x}_0 \sim p_{\text{data}},\, \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})} \left|\left|\epsilon - \epsilon_\theta\left(\mathbf{x}_t \mid t\right)\right|\right|_2^2$$

$$\left(\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon\right)$$

Generation (DDPM) :



$\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

for $t = T \rightarrow 0$:

1. $\hat{\mathbf{x}}_0 = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t))/\sqrt{\bar{\alpha}_t}$
2. $p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \hat{\mathbf{x}}_0), \tilde{\beta}_t \mathbf{I})$
   $\mathbf{x}_{t-1} \sim p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0)$

arxiv.org/abs/2006.11239

Score (gradient of log density) :

$$q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\Rightarrow \log q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \text{const} - \frac{1}{2(1-\bar{\alpha}_t)}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^T(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)$$

$$\Rightarrow \text{(Score) } \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid \mathbf{x}_0) = -\frac{1}{1-\bar{\alpha}_t}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \qquad \text{(2) } \epsilon = \frac{1}{\sqrt{1-\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)$$

$$\Rightarrow \text{(Score) } \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid \mathbf{x}_0) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon$$

∴ Estimating ϵ is equivalent to estimating a scaled version of the **Score**!

1. Score-based Denoising Diffusion Models

2. MCVD: Masked Conditional Video Diffusion

arxiv.org/abs/2205.09853

MCVD: Masked Conditional Video Diffusion

Mila

p past frames: $\mathbf{p} = \{\mathbf{p^i}\}_{i=1}^{p}$ | k current frames: $\mathbf{x}_0 = \{\mathbf{x_0^i}\}_{i=1}^{k}$ | f future frames: $\mathbf{f} = \{\mathbf{f^i}\}_{i=1}^{f}$ | $(\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)$

- **Video Prediction:**

$$\mathbb{E}_{t,\ [\mathbf{p},\mathbf{x}_0]\sim p_{\text{data}},\ \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid \mathbf{p}, t)\right|\right|_2^2$$

- **Video Generation:**

$$\mathbb{E}_{t,\ \mathbf{x}_0\sim p_{\text{data}},\ \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid t)\right|\right|_2^2$$

- **Video Interpolation:**

$$\mathbb{E}_{t,\ [\mathbf{p},\mathbf{x}_0,\mathbf{f}]\sim p_{\text{data}},\ \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid \mathbf{p}, \mathbf{f}, t)\right|\right|_2^2$$

arxiv.org/abs/2205.09853

MCVD: Masked Conditional Video Diffusion

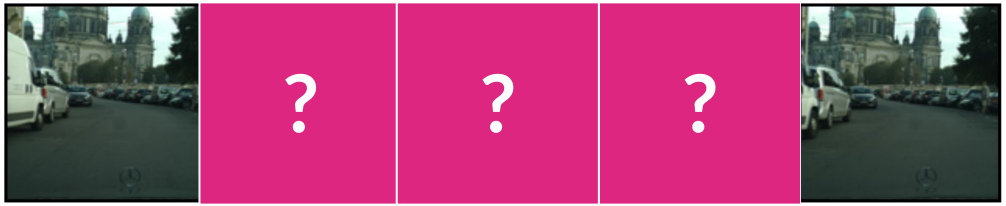
Mila

Real data

Video Prediction

Video Generation

Video Interpolation



arxiv.org/abs/2205.09853

MCVD: Masked Conditional Video Diffusion

Mila

p past frames: $\mathbf{p} = \{\mathbf{p^i}\}_{i=1}^{p}$ | k current frames: $\mathbf{x}_0 = \{\mathbf{x_0^i}\}_{i=1}^{k}$ | f future frames: $\mathbf{f} = \{\mathbf{f^i}\}_{i=1}^{f}$ | $(\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)$

- **Video Prediction:**

$$\mathbb{E}_{t,\ [\mathbf{p},\mathbf{x}_0]\sim p_{\text{data}},\quad \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid \mathbf{p}, t)\right|\right|_2^2$$

Random masking!

- **Video Prediction + Generation:**

$$\mathbb{E}_{t,\ [\mathbf{p},\mathbf{x}_0,\mathbf{f}]\sim p_{\text{data}},\ \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\ m_p\sim\mathcal{B}(p_{\text{mask}})} \left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid m_p\mathbf{p}, t)\right|\right|_2^2$$

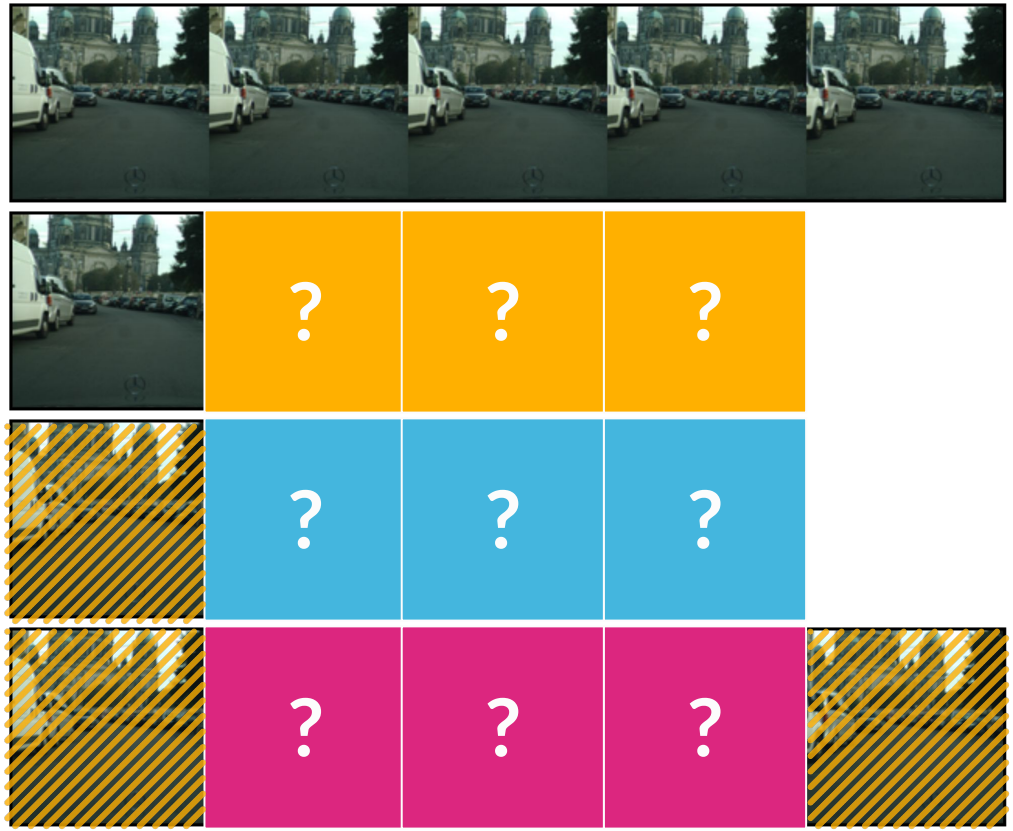
- **Video Prediction + Generation + Interpolation:**

$$\mathbb{E}_{t,\ [\mathbf{p},\mathbf{x}_0,\mathbf{f}]\sim p_{\text{data}},\ \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\ (m_p,m_f)\sim\mathcal{B}(p_{\text{mask}})} \left|\left|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid m_p\mathbf{p}, m_f\mathbf{f}, t)\right|\right|_2^2$$

arxiv.org/abs/2205.09853

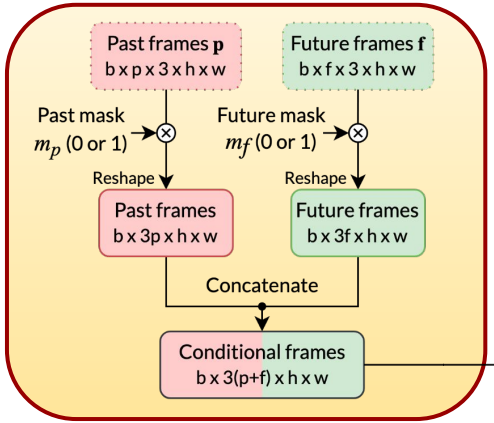
MCVD: Masked Conditional Video Diffusion

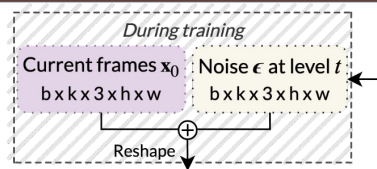
Mila

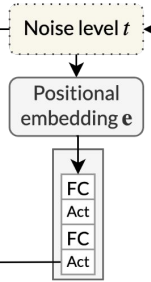


arxiv.org/abs/2205.09853

MCVD: Masked Conditional Video Diffusion





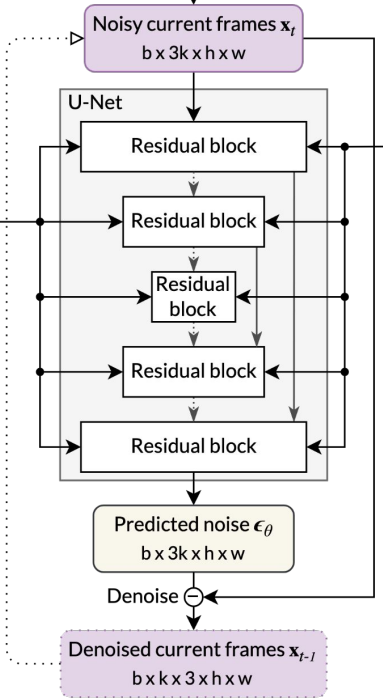


During sampling (T steps)

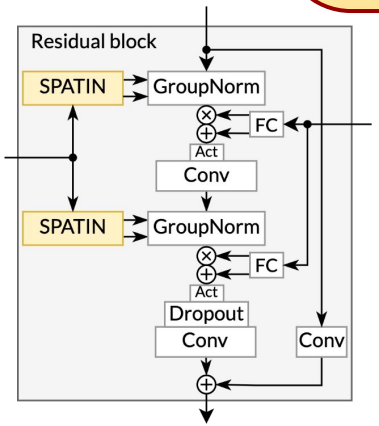
$t = 1, (T-1)/T, \dots, 1/T$

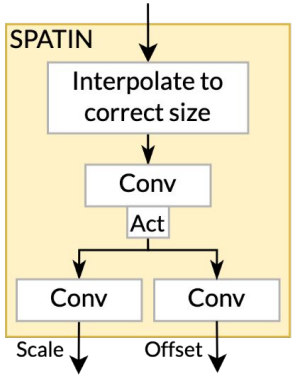
During training

$t \sim U(\{1, (T-1)/T, \dots, 1/T\})$



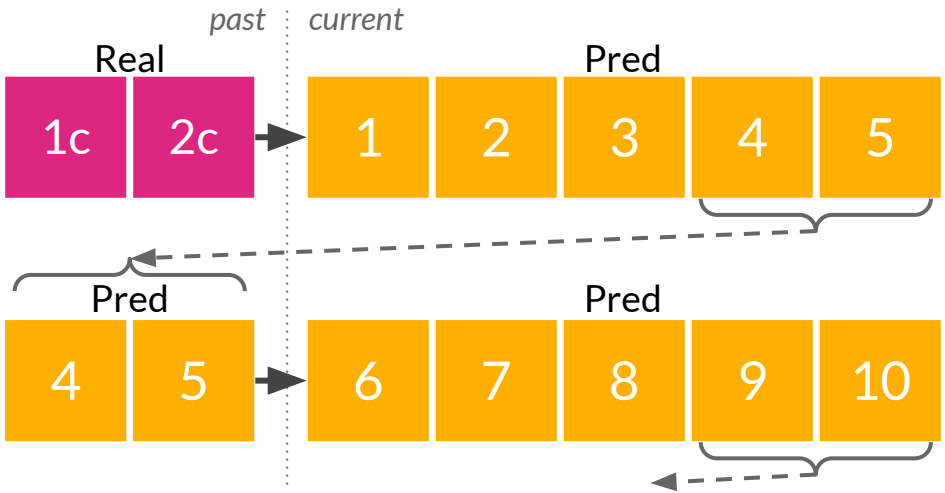
$$\mathbf{e}(t) = \left[\dots, \cos\left(tc^{\frac{-2d}{D}}\right), \sin\left(tc^{\frac{-2d}{D}}\right), \dots\right]^{\mathrm{T}}$$





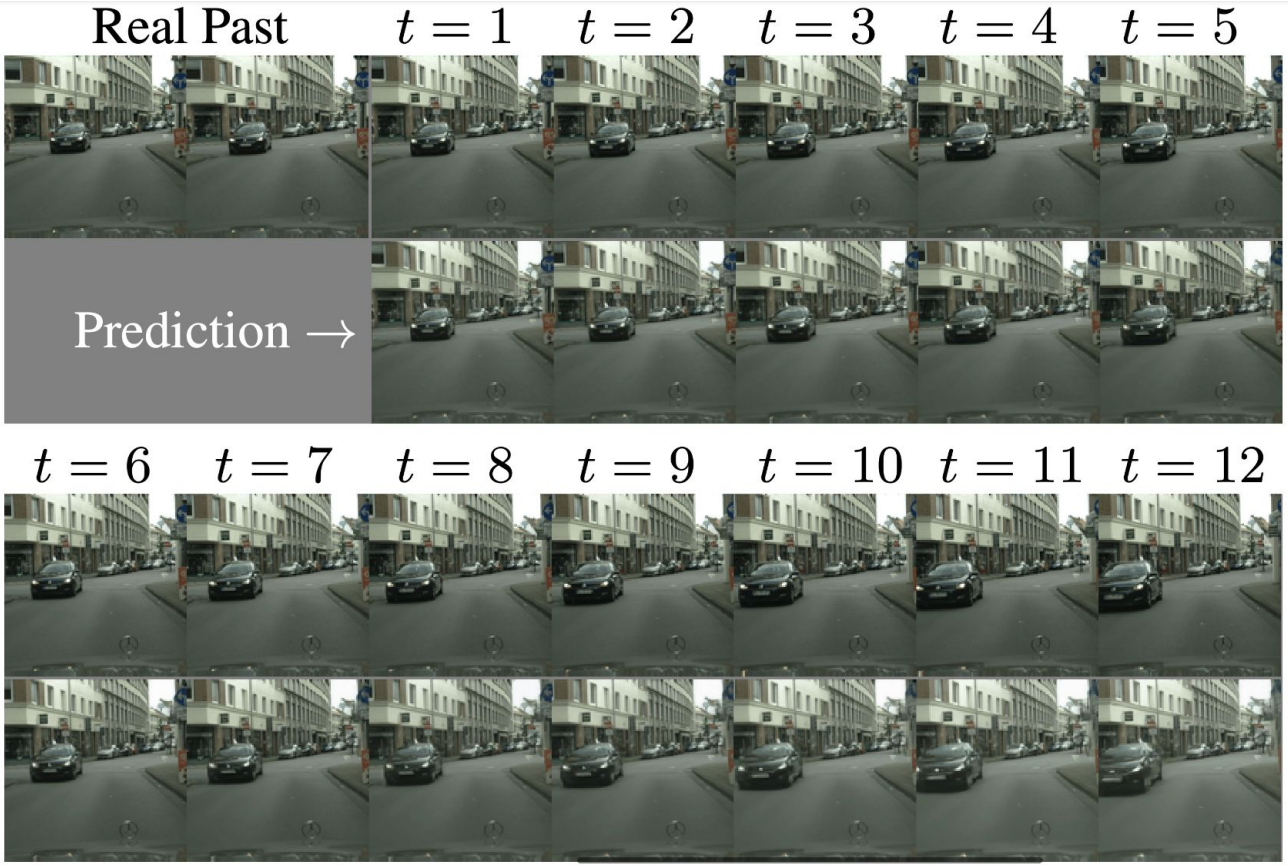
arxiv.org/abs/2205.09853

Block-autoregressive generation:



arxiv.org/abs/2205.09853

MCVD: Masked Conditional Video Diffusion



arxiv.org/abs/2205.09853

MCVD: Masked Conditional Video Diffusion

Mila

(128x128)

Cityscapes [2 → 28; trained on k]	k	FVD↓	LPIPS↓
SVG-LP Denton and Fergus [2018]	10	1300.26	0.549 ± 0.06
vRNN 1L Castrejón et al. [2019]	10	682.08	0.304 ± 0.10
Hier-vRNN Castrejón et al. [2019]	10	567.51	0.264 ± 0.07
GHVAE Wu et al. [2021]	10	418.00	0.193 ± 0.014
MCVD spatin (Ours)	**5**	184.81	0.121 ± 0.05
MCVD concat (Ours)	**5**	**141.31**	**0.112** ± 0.05

(64x64)

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
MCVD concat past-future-mask	**5**	**20.77**	0.753

(64x64)

KTH [10 → $pred$; trained on 10]	$pred$	FVD↓	PSNR↑	SSIM↑
SAVP [Lee et al., 2018]	30	374	26.5	0.756
MCVD spatin (Ours)	30	323	27.5	0.835
MCVD concat past-future-mask (Ours)	30	295	24.3	0.746
SLAMP [Akan et al., 2021]	30	228	29.4	0.865
SRVP [Franceschi et al., 2020]	30	**222**	**29.7**	**0.870**

arxiv.org/abs/2205.09853

(64x64)

BAIR [past p → $pred$; 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**
SVG-LP [Akan et al., 2021]	2	10	28	256.6	–	0.816
SLAMP [Akan et al., 2021]	2	10	28	245.0	19.7	0.818
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

mask-cond-video-diffusion.github.io

arxiv.org/abs/2205.09853

Thank you!

voletiv.github.io