

August 12, 2022



# Solving Video Tasks using Denoising Diffusion Models

[arxiv.org/abs/2205.09853](https://arxiv.org/abs/2205.09853)

[mask-cond-video-diffusion.github.io](https://mask-cond-video-diffusion.github.io)

**Vikram Voleti**

PhD candidate, Mila, University of Montreal

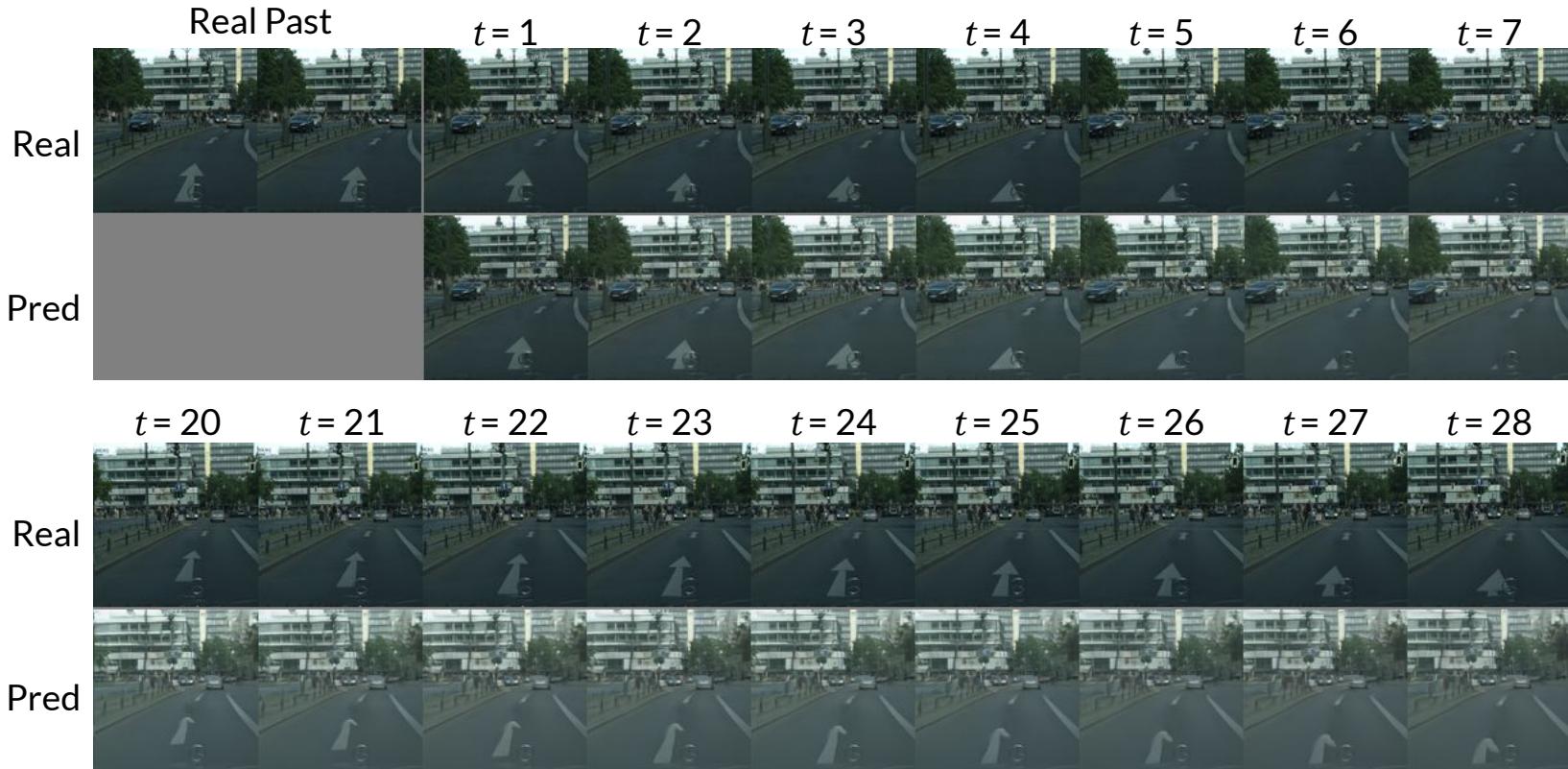
**Supervisor: Prof. Christopher Pal**

[voletiv.github.io](https://voletiv.github.io)

@(virtual) Samsung AI Center, Toronto, Canada



# MCVD: Masked Conditional Video Diffusion

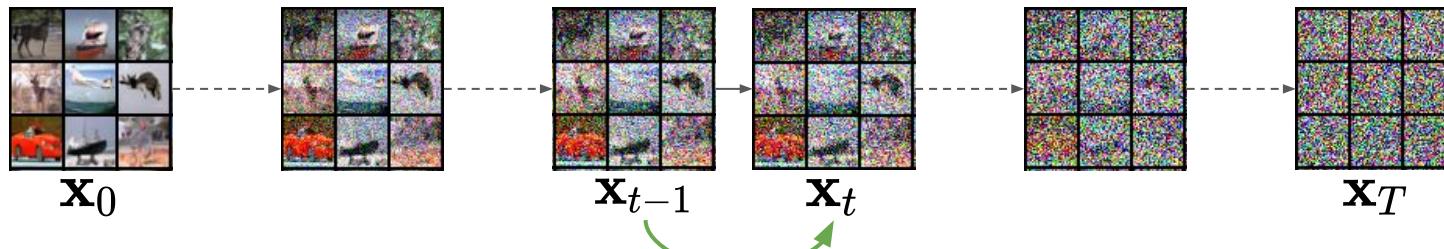


1. Score-based Denoising Diffusion Models
2. MCVD: Masked Conditional Video Diffusion

# Score-based Denoising Diffusion Models



## Forward Process (DDPM)



$$q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \quad ; \bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$$
$$\implies \mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon} \quad ; \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$$

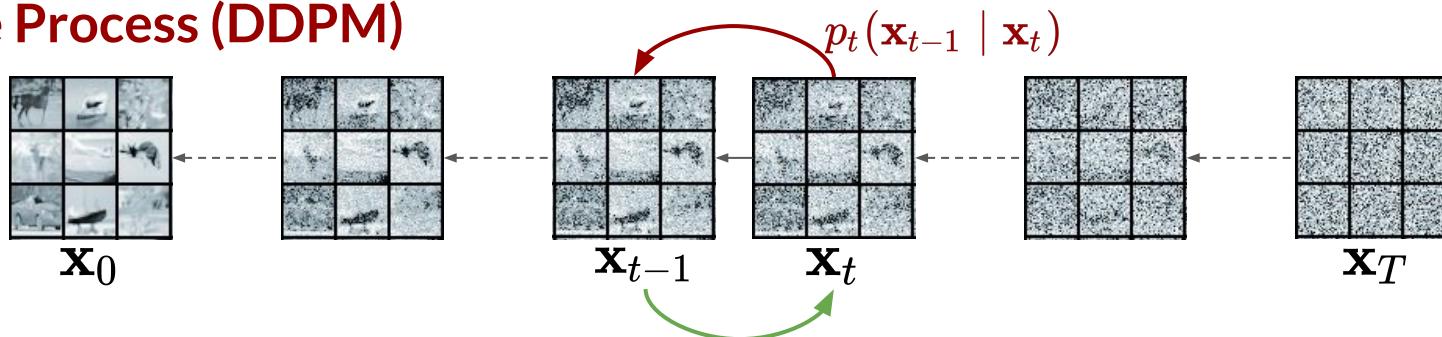
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

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

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

# Score-based Denoising Diffusion Models

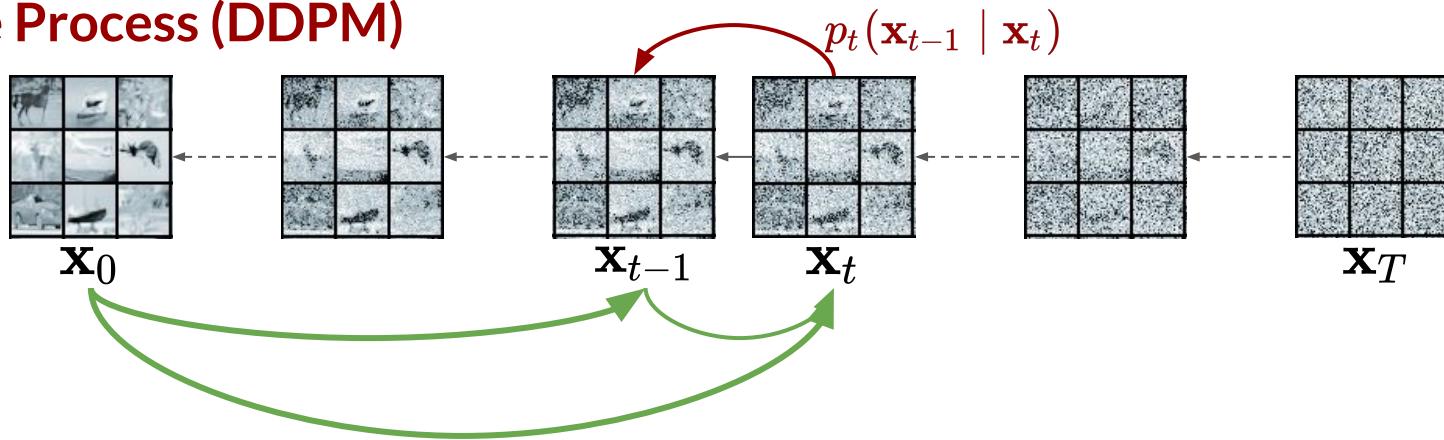
## Reverse Process (DDPM)



$$p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \frac{q_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \checkmark p(\mathbf{x}_{t-1})?}{p(\mathbf{x}_t)?} \quad (\text{Bayes' theorem})$$

# Score-based Denoising Diffusion Models

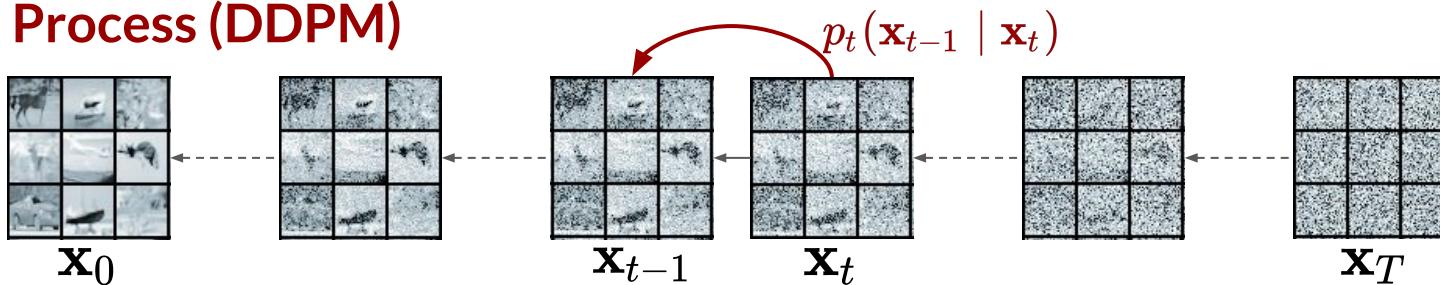
## Reverse Process (DDPM)



$$p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \frac{q_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \checkmark q_t(\mathbf{x}_{t-1} \mid \mathbf{x}_0) \checkmark}{q_t(\mathbf{x}_t \mid \mathbf{x}_0) \checkmark} \quad (\text{Condition on } \mathbf{x}_0)$$

# Score-based Denoising Diffusion Models

## Reverse Process (DDPM)



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

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

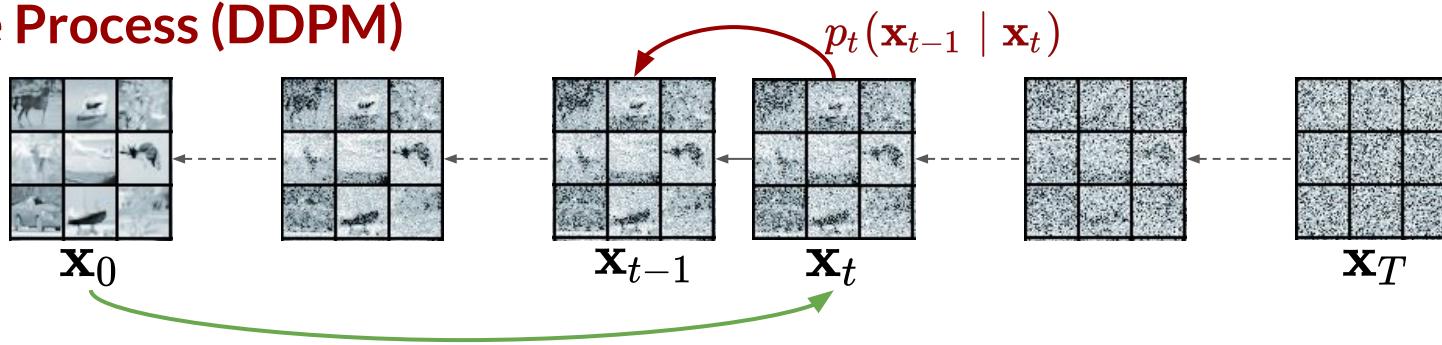
$$\begin{aligned} p(\mathbf{u}) &= \mathcal{N}(\mathbf{u} \mid \mathbf{m}\mathbf{u}, \mathbf{\Lambda}^{-1}), \\ p(\mathbf{v} \mid \mathbf{u}) &= \mathcal{N}(\mathbf{v} \mid \mathbf{A}\mathbf{u} + \mathbf{b}, \mathbf{L}^{-1}) \\ \Rightarrow p(\mathbf{v}) &= \mathcal{N}(\mathbf{v} \mid \mathbf{A}\mu + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{A}^T), \\ \Rightarrow p(\mathbf{u} \mid \mathbf{v}) &= \mathcal{N}(\mathbf{u} \mid \mathbf{C}(\mathbf{A}^T\mathbf{L}(\mathbf{v} - \mathbf{b}) + \mathbf{\Lambda}\mathbf{m}\mathbf{u}), \mathbf{C}) \\ [\mathbf{C} &= (\mathbf{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}] \end{aligned}$$

where  $\mathbf{u} = \mathbf{x}_{t-1} \mid \mathbf{x}_0$ ,  $\mathbf{v} = \mathbf{x}_t$

[arxiv.org/abs/2006.11239](https://arxiv.org/abs/2006.11239)

# Score-based Denoising Diffusion Models

## Reverse Process (DDPM)



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

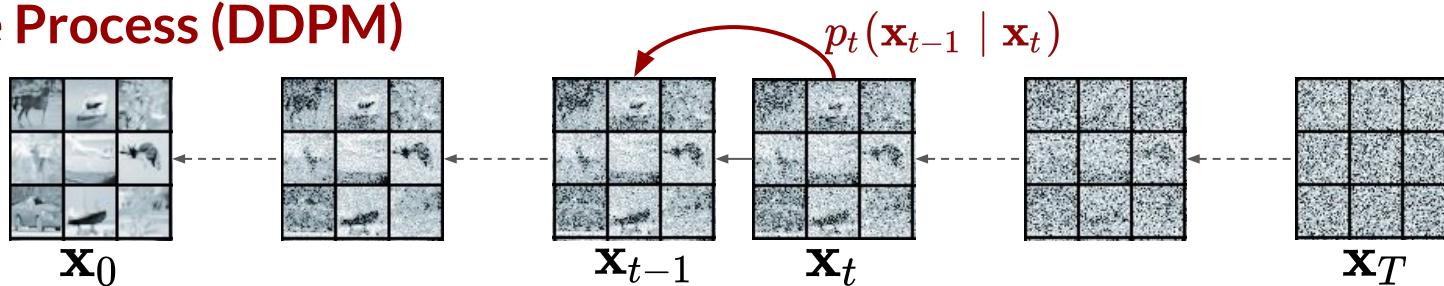
$$2 \quad p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

From (1) :

$$\begin{aligned} q_t(\mathbf{x}_t \mid \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \\ \mathbf{x}_t &= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \\ (1) \quad \mathbf{x}_0 &= (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon) / \sqrt{\bar{\alpha}_t} \end{aligned}$$

# Score-based Denoising Diffusion Models

## Reverse Process (DDPM)



Deep neural network!

$$1 \quad \hat{\mathbf{x}}_0 = (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t, t)) / \sqrt{\bar{\alpha}_t}$$

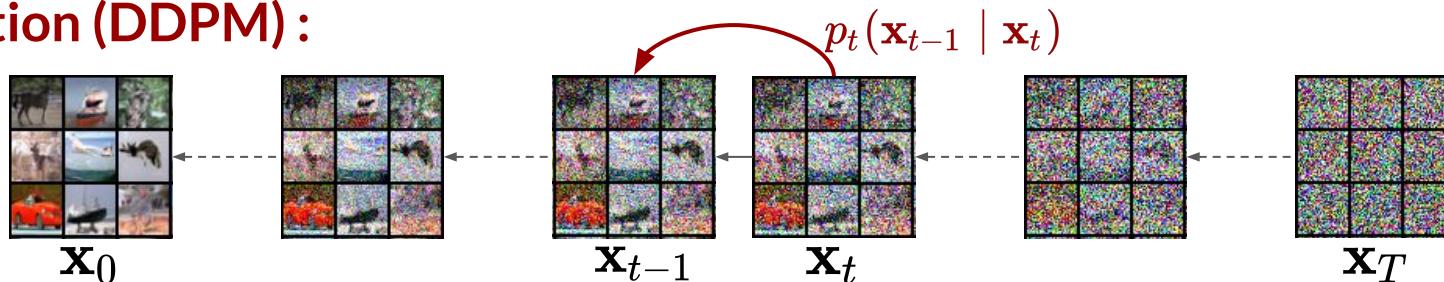
$$2 \quad p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \hat{\mathbf{x}}_0), \tilde{\beta}_t \mathbf{I})$$

Objective function:

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

# Score-based Denoising Diffusion Models

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

$$\mathbf{x}_{t-1} \sim p_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0)$$

# Score-based Denoising Diffusion Models



Score (gradient of log density) :

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

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

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

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

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

• Estimating  $\epsilon$  is equivalent to estimating a scaled version of the Score!

1. Score-based Denoising Diffusion Models
2. **MCVD: Masked Conditional Video Diffusion**

# MCVD: Masked Conditional Video Diffusion



$p$  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:**

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

- **Video Generation:**

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

- **Video Interpolation:**

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

[arxiv.org/abs/2205.09853](https://arxiv.org/abs/2205.09853)

# MCVD: Masked Conditional Video Diffusion



# MCVD: Masked Conditional Video Diffusion



$p$  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:**

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

Random masking!

- **Video Prediction + Generation:**

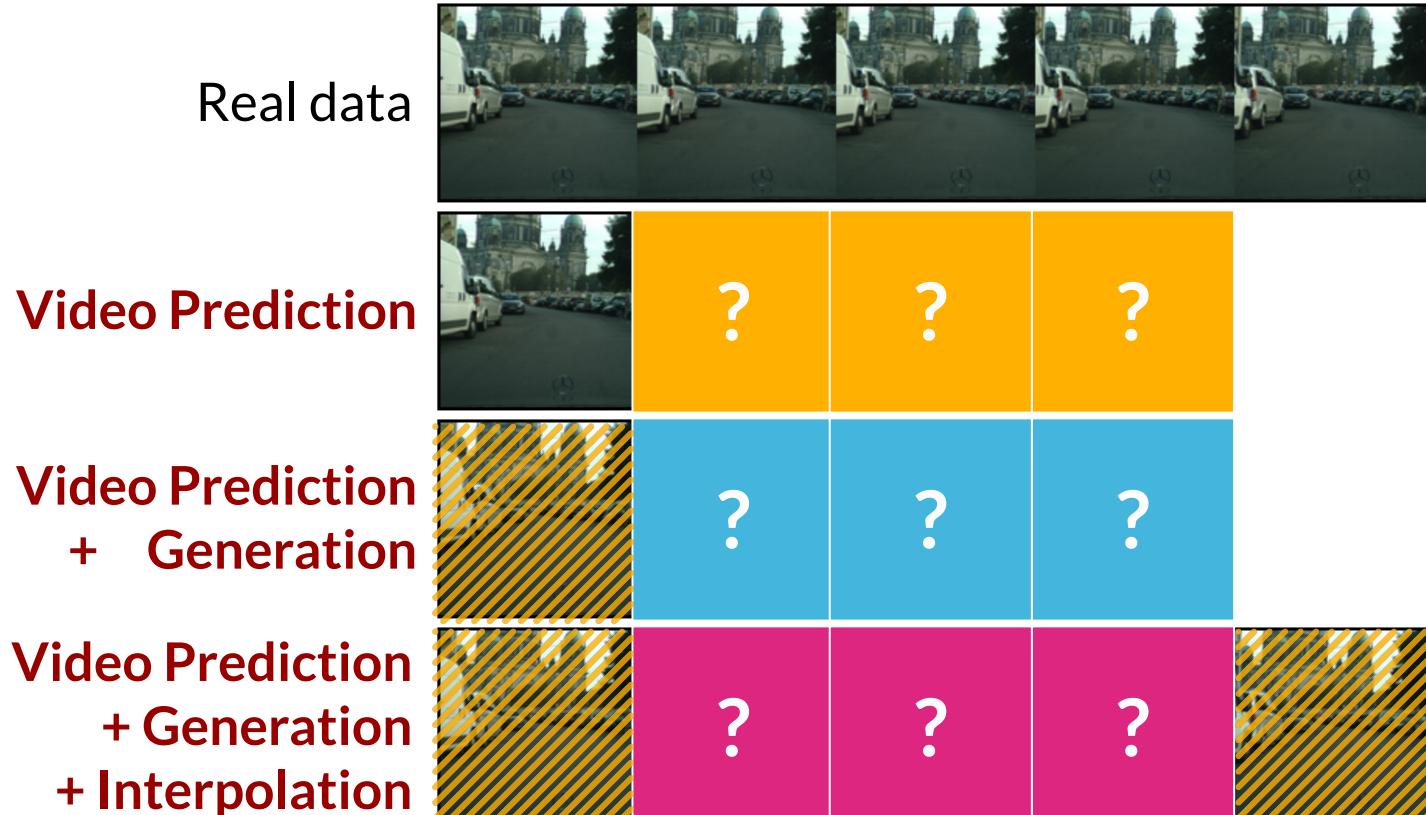
$$\mathbb{E}_{t, [\mathbf{p}, \mathbf{x}_0, \mathbf{f}] \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), m_p \sim \mathcal{B}(p_{\text{mask}})} \|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid m_p \mathbf{p}, t)\|_2^2$$

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

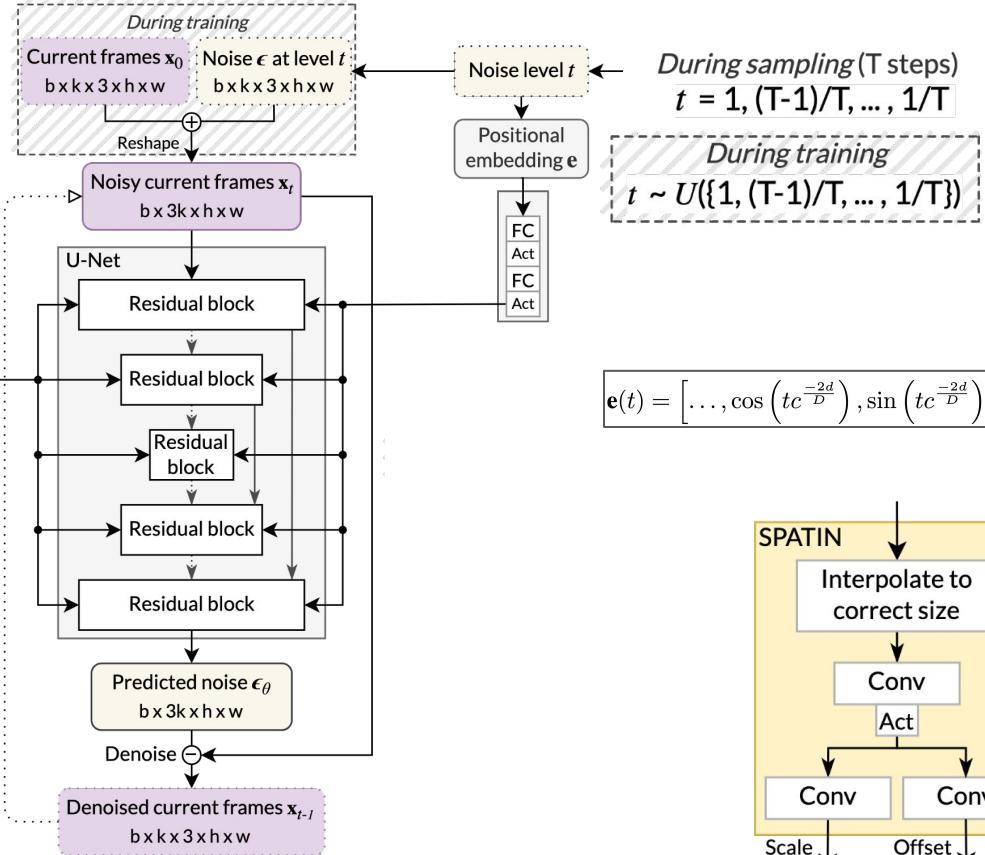
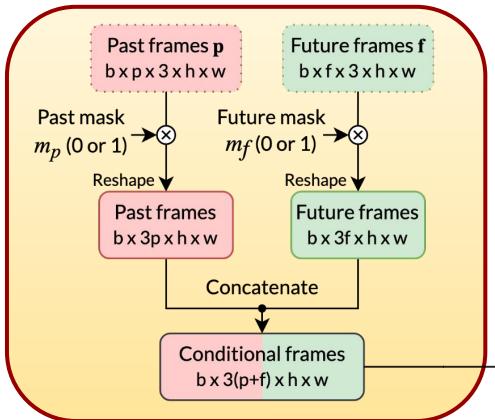
$$\mathbb{E}_{t, [\mathbf{p}, \mathbf{x}_0, \mathbf{f}] \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), (m_p, m_f) \sim \mathcal{B}(p_{\text{mask}})} \|\epsilon - \epsilon_\theta(\mathbf{x}_t \mid m_p \mathbf{p}, m_f \mathbf{f}, t)\|_2^2$$

[arxiv.org/abs/2205.09853](https://arxiv.org/abs/2205.09853)

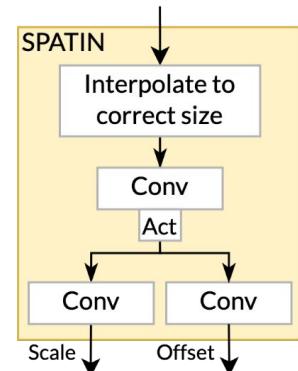
# MCVD: Masked Conditional Video Diffusion



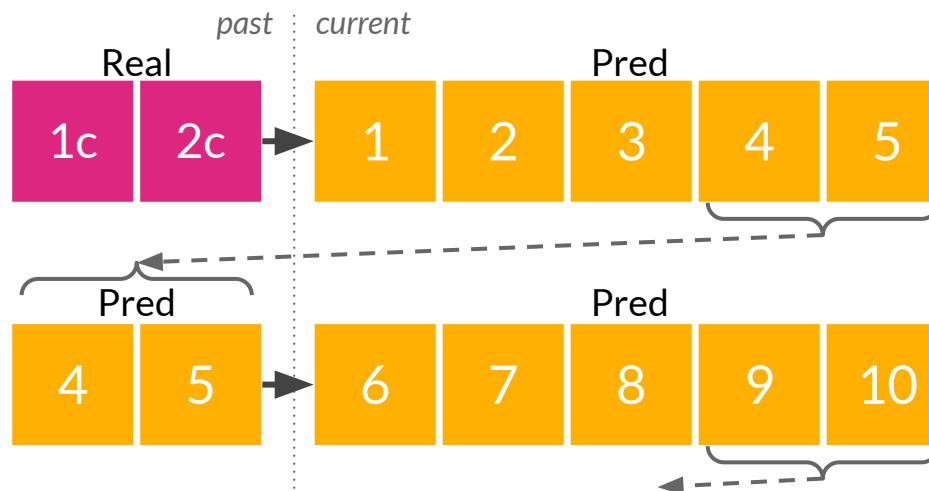
# MCVD: Masked Conditional Video Diffusion



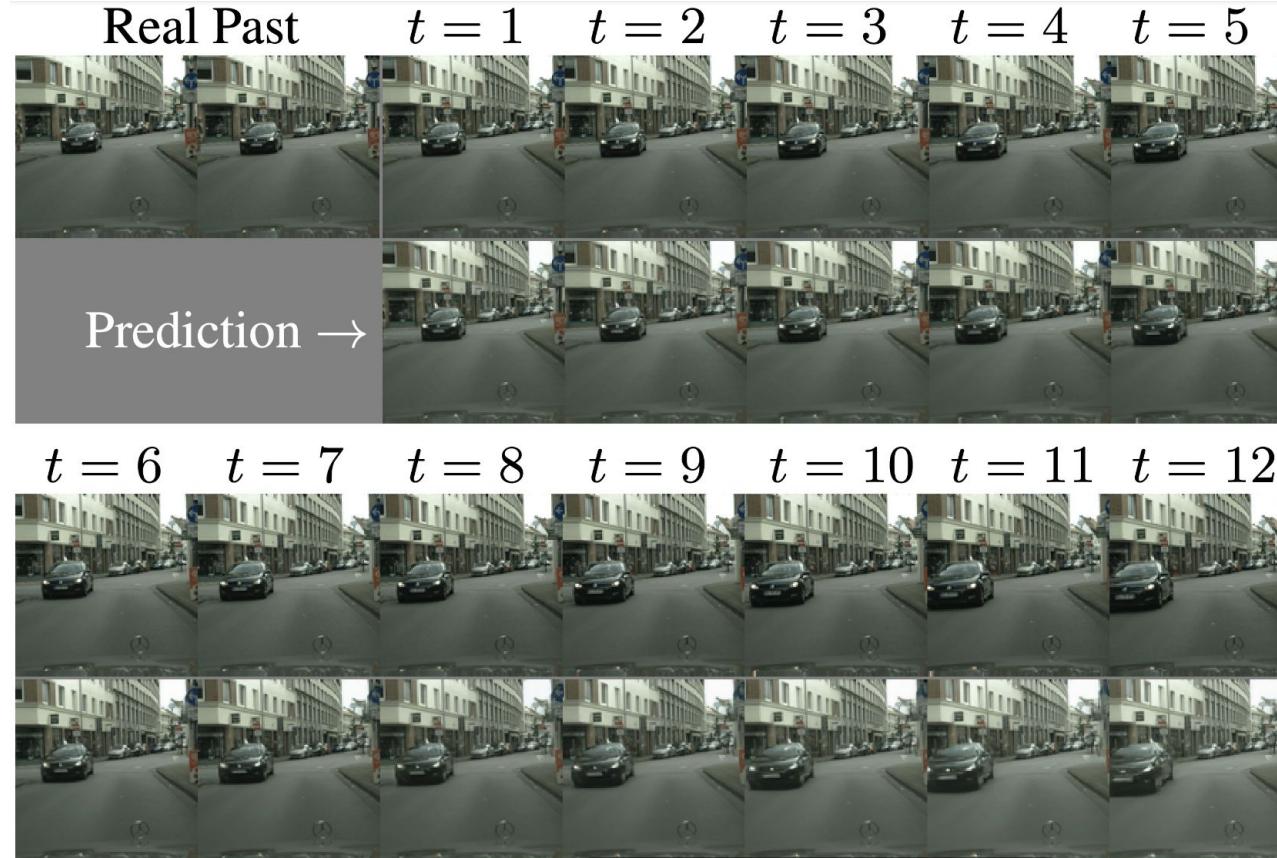
$$\mathbf{e}(t) = [\dots, \cos\left(tc^{\frac{-2d}{D}}\right), \sin\left(tc^{\frac{-2d}{D}}\right), \dots]^T$$



## Block-autoregressive generation:



# MCVD: Masked Conditional Video Diffusion



# MCVD: Masked Conditional Video Diffusion



(128x128)

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

(64x64)

BAIR [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	–	–
<b>MCVD</b> spatin (Ours)	1	<b>5</b>	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	–	–
<b>MCVD</b> concat (Ours)	1	<b>5</b>	15	98.8	18.8	0.829
<b>MCVD</b> spatin past-mask (Ours)	1	<b>5</b>	15	96.5	18.8	0.828
<b>MCVD</b> concat past-mask (Ours)	1	<b>5</b>	15	<b>95.6</b>	<b>18.8</b>	<b>0.832</b>
Video Transformer [Weissenborn et al., 2019]	1	15	15	94-96 <sup>a</sup>	–	–
FitVid [Babaeizadeh et al., 2021]	1	15	15	93.6	–	–
<b>MCVD</b> concat past-future-mask (Ours)	1	<b>5</b>	15	<b>89.5</b>	16.9	0.780
SAVP [Lee et al., 2018]	2	14	14	116.4	–	–
<b>MCVD</b> spatin (Ours)	2	<b>5</b>	14	94.1	19.1	0.836
<b>MCVD</b> spatin past-mask (Ours)	2	<b>5</b>	14	90.5	<b>19.2</b>	0.837
<b>MCVD</b> concat (Ours)	2	<b>5</b>	14	90.5	19.1	0.834
<b>MCVD</b> concat past-future-mask (Ours)	2	<b>5</b>	14	89.6	17.1	0.787
<b>MCVD</b> concat past-mask (Ours)	2	<b>5</b>	14	<b>87.9</b>	19.1	<b>0.838</b>
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	–	<b>0.822</b>
<b>MCVD</b> spatin (Ours)	2	<b>5</b>	28	132.1	17.5	0.779
<b>MCVD</b> spatin past-mask (Ours)	2	<b>5</b>	28	127.9	17.7	0.789
<b>MCVD</b> concat (Ours)	2	<b>5</b>	28	120.6	17.6	0.785
<b>MCVD</b> concat past-mask (Ours)	2	<b>5</b>	28	119.0	<b>17.7</b>	0.797
<b>MCVD</b> concat past-future-mask (Ours)	2	<b>5</b>	28	<b>118.4</b>	16.2	0.745

[arxiv.org/abs/2205.09853](https://arxiv.org/abs/2205.09853)

[mask-cond-video-diffusion.github.io](https://mask-cond-video-diffusion.github.io)



# Thank you!