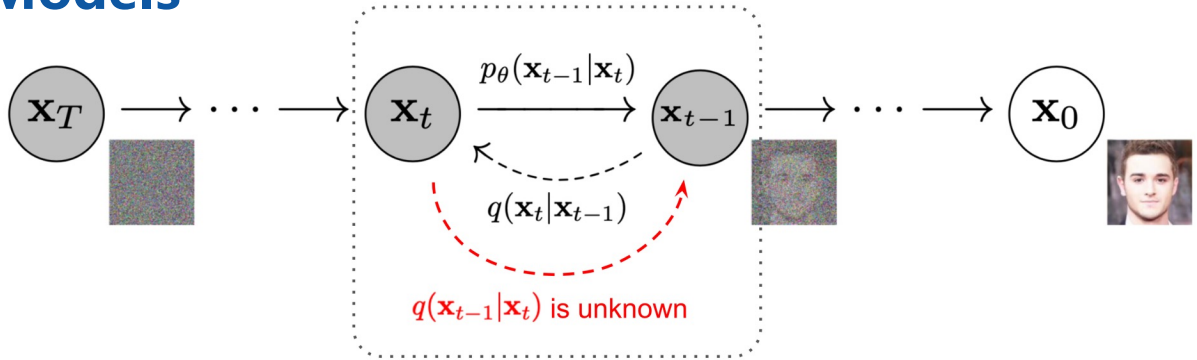
JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation

*Zhengxiong Luo[1,2,4,5] Dayou Chen[2] Yingya Zhang[2]
†Yan Huang[4,5] Liang Wang[4,5] Yujun Shen[3] Deli Zhao[2] Jingren Zhou[2] Tieniu Tan[4,5,6]
[1]University of Chinese Academy of Sciences (UCAS) [2]Alibaba Group
[3]Ant Group [4]Center for Research on Intelligent Perception and Computing (CRIPAC)
[5]Institute of Automation, Chinese Academy of Sciences (CASIA) [6]Nanjing University

*Work done at Alibaba DAMO Academy.
†Corresponding author.

Diffusion Models



- Encoding:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\,\epsilon_{t-1}$$

$$s.t. \quad \epsilon_{t-1} \sim N(0,1)$$

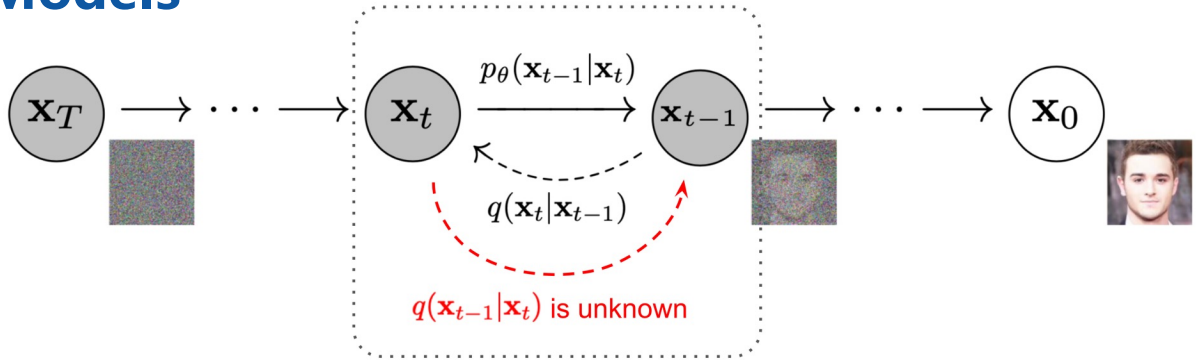
$$p(x_t|x_{t-1}) \sim N(\sqrt{1-\beta_t}x_{t-1}, \beta_t)$$

$$x_t = \sqrt{\hat{a}_t}x_0 + \sqrt{1-\hat{a}_t}\,\hat{\epsilon}_t$$

$$s.t. \begin{cases} \hat{\alpha}_t = \prod_1^t \alpha_t \\ \hat{\epsilon}_t \sim N(0,1) \\ \alpha_t = 1-\beta_t \end{cases}$$

$$p(x_t|x_0) \sim N(\sqrt{\hat{a}_t}x_0, 1-\hat{a}_t)$$

Diffusion Models



- Decoding:

$$q(x_{t-1}, x_t|x_0) = q(x_{t-1}|x_t, x_0)q(x_t|x_0)$$
$$= q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)$$

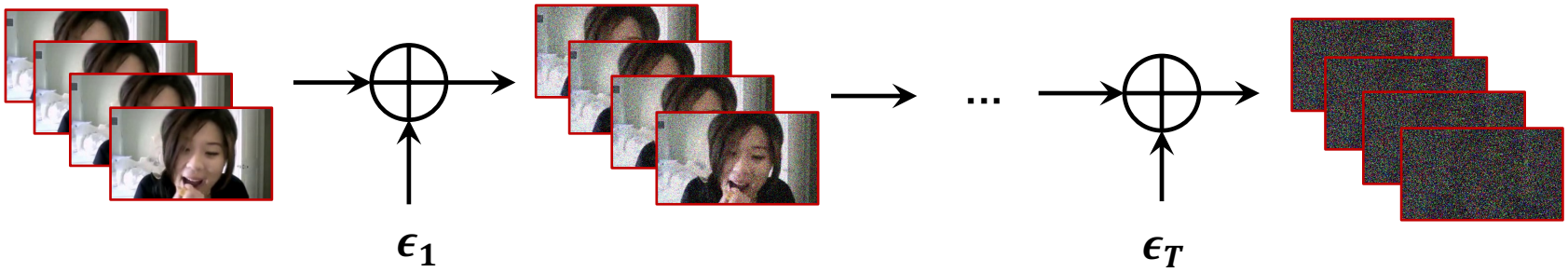
$$\Rightarrow \quad q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0)q(x_t|x_0)}{q(x_{t-1}|x_0)}$$

$$\Rightarrow \quad x_{t-1} = \frac{x_t}{\sqrt{\alpha_t}} - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\sqrt{1-\eta}z_\theta - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\sqrt{\eta}\epsilon'_t \qquad s.t. \begin{cases} \hat{\epsilon}_t \coloneqq z_\theta(\mathrm{x_t}, \mathrm{t}) \\ \epsilon'_t \sim N(0,1) \end{cases}$$

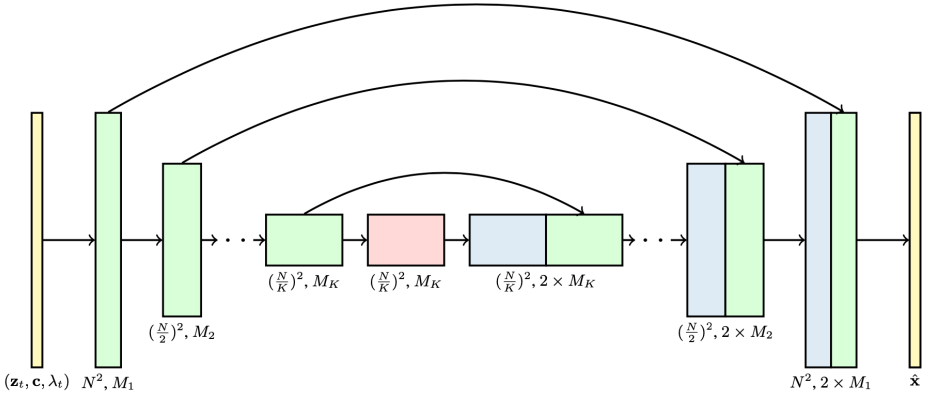
$\eta = 0 \longrightarrow$ DDIM Sampling $\qquad$ $\eta = \frac{1-\hat{\alpha}_{t-1}}{1-\hat{\alpha}_t}\alpha_t \longrightarrow$ DDPM Sampling

Related Works

- Encoding:



- Decoding:



- Each frame is individually encoded, ignoring the temporal correlation and redundancy.
- The coherence of the generated videos relies only on the temporal attention module in the denoising network.

Decomposed Diffusion Models

$$x^i = \sqrt{\lambda^i} x^0 + \sqrt{1-\lambda^i}\Delta x^i, \quad i = 1, 2, ..., N$$

$$\downarrow$$

$$z_t^i = \sqrt{\hat{\alpha}_t}(\sqrt{\lambda^i} x^0 + \sqrt{1-\lambda^i}\Delta x^i) + \sqrt{1-\hat{\alpha}_t}\epsilon_t^i$$

$$\downarrow$$

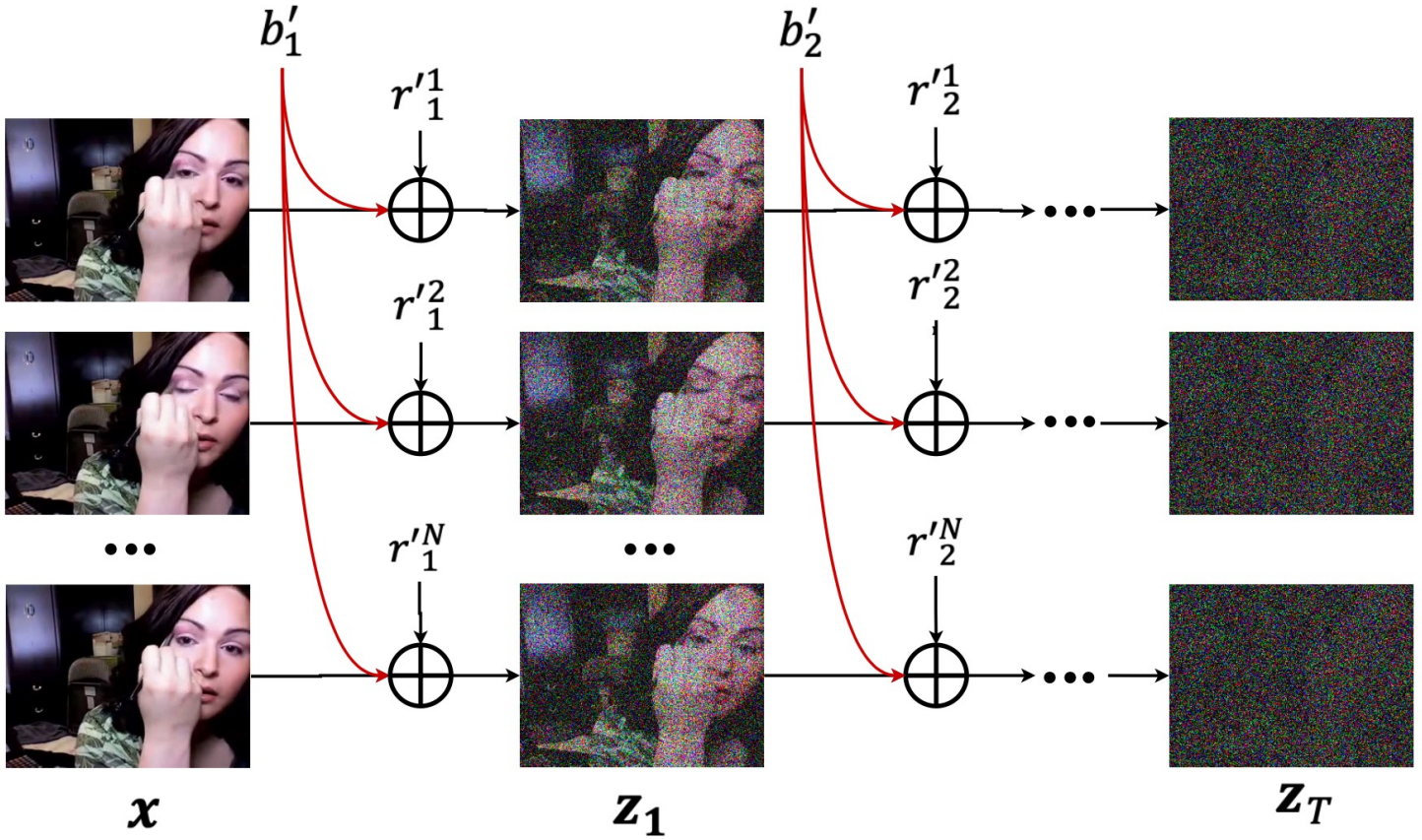
$$\epsilon_t^i = \sqrt{\lambda^i} b_t^i + \sqrt{1-\lambda^i} r_t^i \quad b_t^i, r_t^i \sim \mathcal{N}(0,1)$$

$$\downarrow$$

$$z_t^i = \sqrt{\lambda^i}(\underbrace{\sqrt{\hat{\alpha}_t} x^0 + \sqrt{1-\hat{\alpha}_t} b_t^i}_{\text{diffusion of } x^0}) +$$

$$\sqrt{1-\lambda^i}(\underbrace{\sqrt{\hat{\alpha}_t}\Delta x^i + \sqrt{1-\hat{\alpha}_t} r_t^i}_{\text{diffusion of } \Delta x^i}).$$

Decomposed Diffusion Models



Decomposed Diffusion Models

Figure 2. Comparisons between images generated from (a) independent noises; (b) noises with a shared base noise. Images of the same row are generated by the decoder of DALLE-2 [24] with the same condition.

Decomposed Diffusion Models

$$z_t^i = \begin{cases} \sqrt{\hat{\alpha}_t} x^i + \sqrt{1-\hat{\alpha}_t} b_t & i = \lfloor N/2 \rfloor \\ \sqrt{\hat{\alpha}_t} x^i + \sqrt{1-\hat{\alpha}_t}(\sqrt{\lambda^i} b_t + \sqrt{1-\lambda^i} r_t^i) & i \neq \lfloor N/2 \rfloor \end{cases}$$

$$\mathcal{L}_t = \begin{cases} \|\epsilon_t^i - \mathbf{z}_\phi^b(z_t^{\lfloor N/2 \rfloor}, t)\|^2 & i = \lfloor N/2 \rfloor \\ \|\epsilon_t^i - \sqrt{\lambda^i}[\mathbf{z}_\theta^b(z_t^{\lfloor N/2 \rfloor}, t)]_{sg} - \sqrt{1-\lambda^i}\mathbf{z}_\psi^r(z_t'^i, t, i)\|^2 & i \neq \lfloor N/2 \rfloor \end{cases}$$

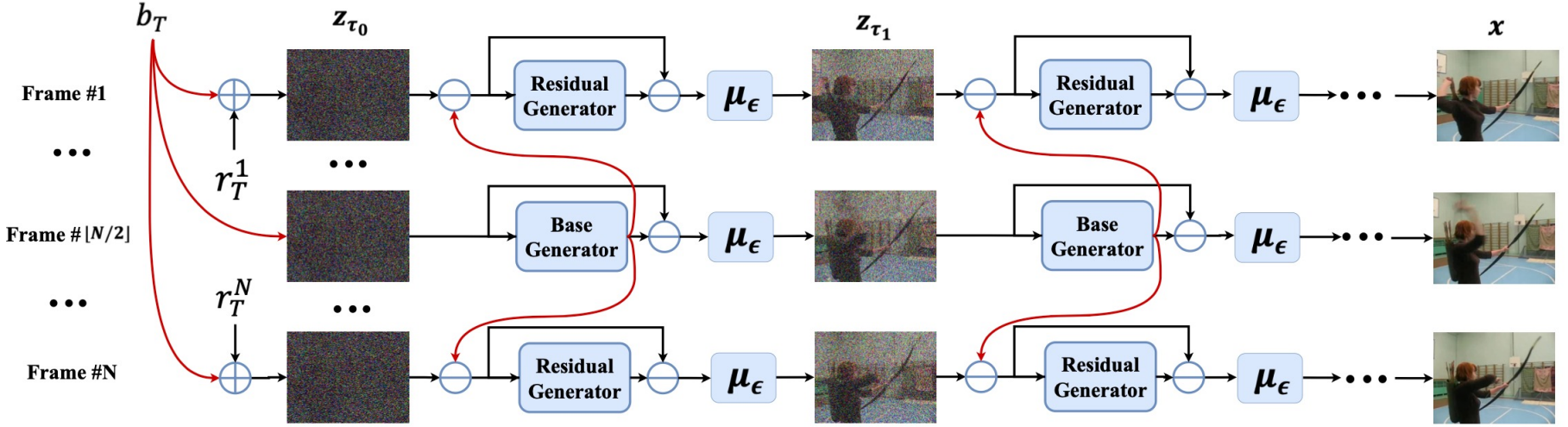


Figure 4. Visualization of DDIM [35] sampling process of DecDPM. In each sampling step, we first remove the base noise with the base generator and then estimate the remaining residual noise via the residual generator. τ_i denotes the DDIM sampling steps. μ_ϵ denotes mean-value predicted function of DDIM in ϵ-prediction formulation. We omit the coefficients and conditions in the figure for simplicity.

Decomposed Diffusion Models

Algorithm 2 DDIM Sampling

Sampling $b \sim \mathcal{N}(0,1)$, $z^{\lfloor N/2 \rfloor} \leftarrow b$
for $i = 1$ to N and $i \neq \lfloor N/2 \rfloor$ do
  Sampling $r^i \sim \mathcal{N}(0,1)$
  $z^i = \sqrt{\lambda^i} b + \sqrt{1-\lambda^i} r^i$
end for
for $t = T$ to 1 do
  $b \leftarrow \mathbf{z}^b_\phi(z^{\lfloor N/2 \rfloor}, t)$; $\epsilon^{\lfloor N/2 \rfloor} \leftarrow b$
  for $i = 1$ to N and $i \neq \lfloor N/2 \rfloor$ do
    $z'^i \leftarrow z^i - \sqrt{\lambda^i}\sqrt{1-\hat{\alpha}} b$
    $r^i \leftarrow \mathbf{z}^r_\psi(z'^i, t, i)$
    $\epsilon^i \leftarrow \sqrt{\lambda^i} b + \sqrt{1-\lambda^i} r^i$
  end for
  for $i = 1$ to N do
    $z^i \leftarrow \sqrt{\hat{\alpha}_{t-1}}\left(\frac{z^i - \sqrt{1-\hat{\alpha}_t}\epsilon^i}{\sqrt{\hat{\alpha}_t}}\right) + \sqrt{1-\hat{\alpha}_{t-1}}\epsilon^i$
  end for
end for
return $\{z^i \mid i = 1, 2, \ldots, N\}$

Algorithm 1 DDPM Sampling

Sampling $b \sim \mathcal{N}(0,1)$, $z^{\lfloor N/2 \rfloor} \leftarrow b$
for $i = 1$ to N **and** $i \neq \lfloor N/2 \rfloor$ **do**
  Sampling $r^i \sim \mathcal{N}(0,1)$
  $z^i = \sqrt{\lambda^i} b + \sqrt{1-\lambda^i} r^i$
end for
for $t = T$ to 1 **do**
  $b \leftarrow \mathbf{z}^b_\phi(z^{\lfloor N/2 \rfloor}, t)$; $\epsilon^{\lfloor N/2 \rfloor} \leftarrow b$
  for $i = 1$ to N **and** $i \neq \lfloor N/2 \rfloor$ **do**
    $z'^i \leftarrow z^i - \sqrt{\lambda^i}\sqrt{1-\hat{\alpha}} b$; $r^i \leftarrow \mathbf{z}^r_\psi(z'^i, t, i)$
    $\epsilon^i \leftarrow \sqrt{\lambda^i} b + \sqrt{1-\lambda^i} r^i$
  end for
  for $i = 1$ to N **do**
    $\mu^i \leftarrow \frac{1}{\sqrt{\alpha_t}} z^i - \frac{1-\alpha_t}{\sqrt{1-\hat{\alpha}_t}\sqrt{\alpha_t}} \epsilon^i$; $\sigma \leftarrow \frac{1-\hat{\alpha}_{t-1}}{1-\hat{\alpha}_t}(1-\alpha_t)$
  end for
  Sampling $b \sim \mathcal{N}(0,1)$
  $z^{\lfloor N/2 \rfloor} \leftarrow \sigma b + \mu^{\lfloor N/2 \rfloor}$
  for $i = 1$ to N **and** $i \neq \lfloor N/2 \rfloor$ **do**
    Sampling $r^i \sim \mathcal{N}(0,1)$
    $z^i \leftarrow \sigma(\sqrt{\lambda^i} b + \sqrt{1-\lambda^i} r^i) + \mu^i$
  end for
end for
return $\{z^i \mid i = 1, 2, \ldots, N\}$

Experiments

Table 1. Quantitative comparisons on UCF101.↓ denotes the lower the better. ↑ denotes the higher the better. The best results are denoted in bold.

Method	Resolution	IS ↑	FVD↓
Unconditional			
TGAN [29]	16 × 64 × 64	11.85	–
MoCoGAN-HD [40]	16 × 128 × 128	32.36	838
DIGAN [50]	16 × 128 × 128	32.70	577
StyleGAN-V [34]	16 × 256 × 256	23.94	–
VideoGPT [47]	16 × 128 × 128	24.69	–
TATS [9]	16 × 128 × 128	57.63	420
VDM [16]	16 × 64 × 64	57.00	295
VideoFusion	16 × 64 × 64	71.67	**139**
VideoFusion	16 × 128 × 128	**72.22**	220
Class-conditioned			
VGAN [45]	16 × 64 × 64	8.31	–
TGAN [29]	16 × 64 × 64	15.83	–
TGANv2 [30]	16 × 128 × 128	28.87	1209
MoCoGAN [40]	16 × 64 × 64	12.42	–
DVD-GAN [4]	16 × 128 × 128	32.97	–
CogVideo [17]	16 × 160 × 160	50.46	626
TATS [9]	16 × 128 × 128	79.28	332
VideoFusion	16 × 128 × 128	**80.03**	**173**

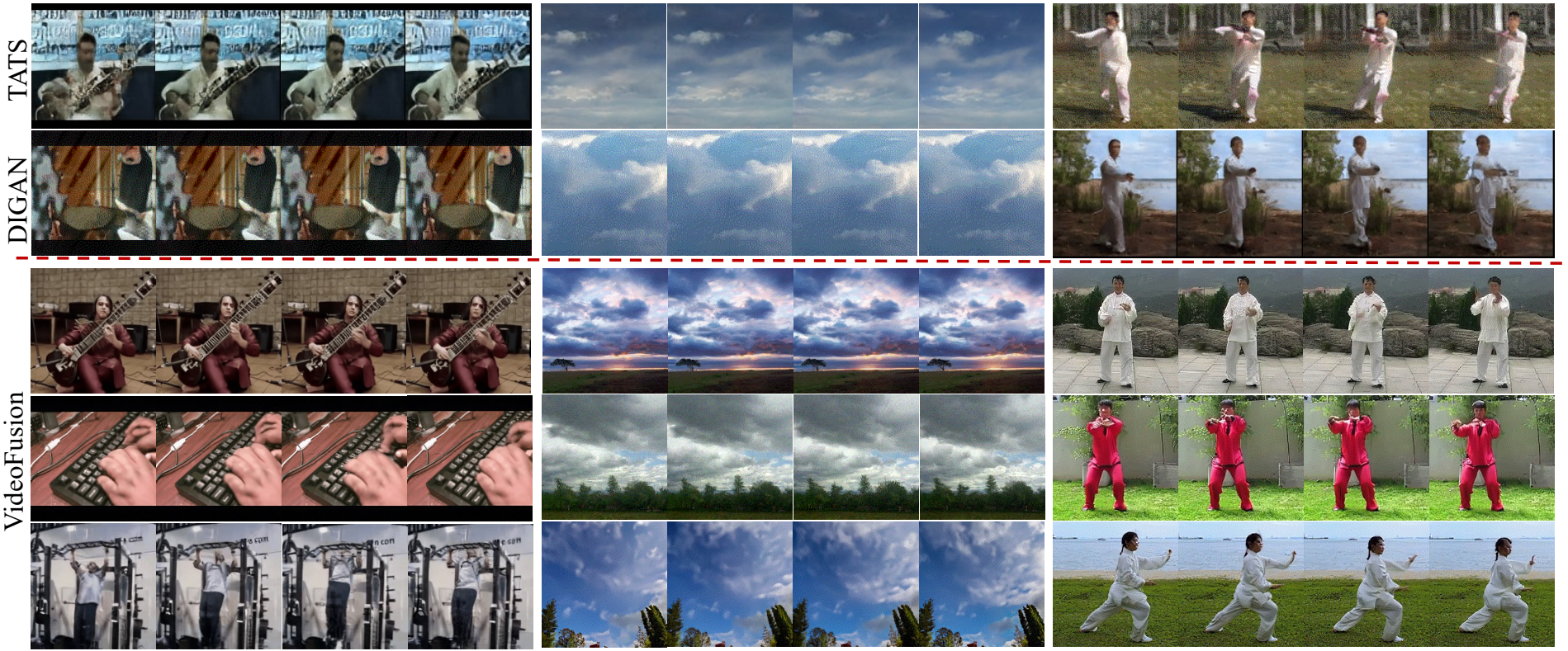
Table 2. Quantitative comparisons on Sky Time-lapse [46]. ↓ denotes the lower the better. The best results are denoted in bold.

Method	FVD (↓)	KVD (↓)
MoCoGAN-HD [40]	183.6	13.9
DIGAN [50]	114.6	6.8
TATS [9]	132.6	5.7
VideoFusion	**47.0**	**5.3**

Table 3. Quantitative comparisons on TaiChi-HD [33]. ↓ denotes the lower the better. The best results are denoted in bold.

Method	FVD (↓)	KVD (↓)
MoCoGAN-HD [40]	144.7	25.4
DIGAN [50]	128.1	20.6
TATS [9]	94.6	9.8
VideoFusion	**56.4**	**6.9**

Experiments



(a) UCF101 (b) Sky Time-Lapse (c) TaiChi-HD

Experiments

Table 4. We re-implement VDM [16] (denoted as VDM*) based on the base generator of VideoFusion. The efficiency comparisons are shown below.

Method	Memory (GB)	Latency (s)
VDM*	63.82	0.40
VideoFusion	49.85(↓ 21.8%)	0.17(↓ 57.5%)

Table 5. Study on λ^i. Unconditional generation results on UCF101 [39].

λ^i	0.10	0.25	0.50	0.75
IS ↑	67.23	69.16	71.67	69.56
FVD ↓	149	122	139	181

Table 6. Study on pretraining. Unconditional generation results on UCF101 [39].

Method	IS↑	FVD ↓
VDM [16]	57.00	295
VideoFusion w/o pretrain	65.29	183
VideoFusion w/ pretrain	71.67	139

Table 7. Study on joint training. Unconditional generation results on UCF101 [39].

Training method	IS↑	FVD ↓
Fixed	65.06	187
w/o stop gradient	67.86	168
w stop gradient	71.67	139

Experiments Results

Experiments Results

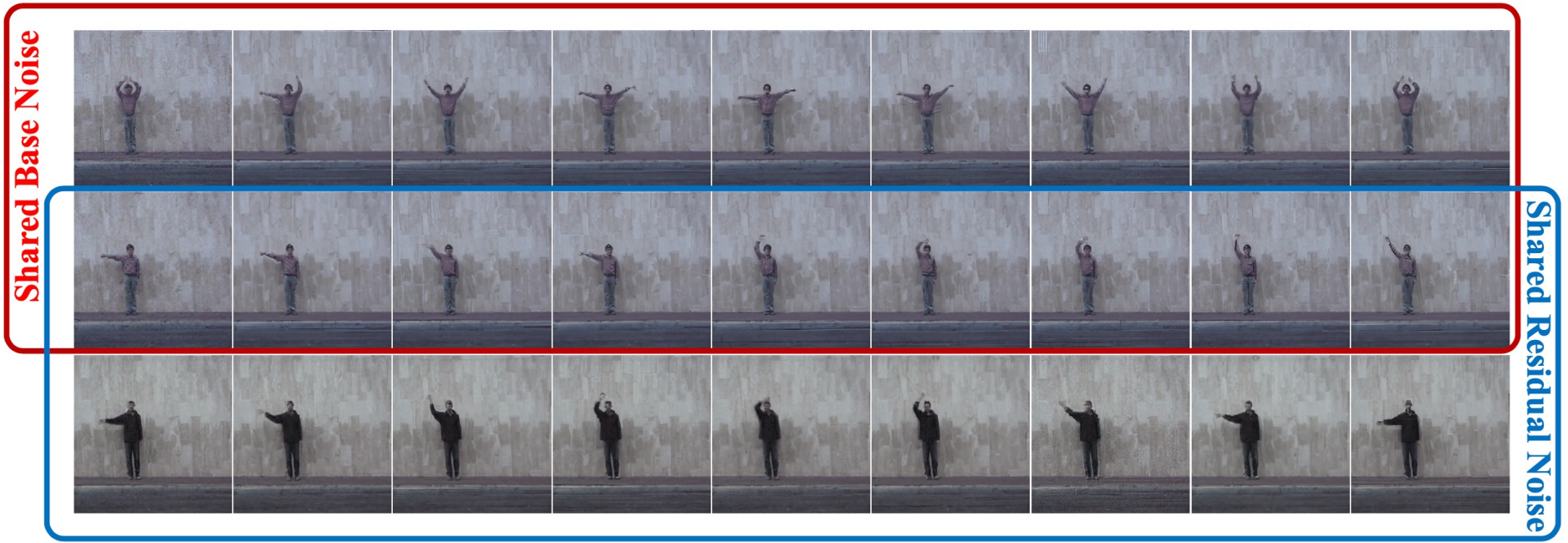


Figure 1. Unconditional generation results on the Weizmann Action datasets [11]. Videos of the top-two rows share the same base noise but have different residual noises. Videos of the bottom-two rows share the same residual noise but have different base noises.

Additional Information

https://www.modelscope.cn/models/damo/cv_diffusion_text-to-video-synthesis/summary

https://www.modelscope.cn/models/damo/text-to-video-synthesis/summary

Q&A