VQ-VDM: ベクトル量子化変分オートエンコーダと拡散モデルを用いた動画生成モデル

梶 凌太[1,a)]　柳井 啓司[1,b)]

概要

近年，深層学習を用いた画像生成モデルは優れた性能を達成しており，次のモダリティとして動画生成に対しても注目が集まっている。しかしながら，動画は画像を複数フレーム重ねたものであるため，時系列情報の考慮，計算量の増加という 2 つの点で画像生成と比較して困難である。本研究では 3D VQGAN と diffusion models を用いた動画生成モデル VQ-VDM を提案する。VQ-VDM は，動画を直接生成する Video Diffusion Models と比較して約 9 倍の高速化を達成した。また，Video Diffusion Models に対してはより高解像度な生成により，他の動画生成手法に対しては diffusion models を用いた生成のため，より高品質な動画生成が可能となった。

1. はじめに

近年，Deep Learning を用いた生成モデルは優れた性能を達成しており，画像生成においては実画像と見分けがつかないような生成を実現している。特に最近の Latent Diffusion Models (LDM) [12] を始めとする画像生成モデルは，社会に大きなインパクトを与えている。そして，次のモダリティとして動画生成に対しても注目が集まっている。しかしながら，動画は画像を複数フレーム重ねたものであるため，時系列情報の考慮，計算量の増加という 2 つの点で画像生成と比較して困難である。Transformer を用いた手法 [11], [27] は数多く提案されているが，自己回帰手法であるため生成する度に損失が蓄積してしまう問題が存在する。これに対し，Video Diffusion Models (VDM) [6] は全てのフレームを同時に生成するため，損失を蓄積することなく高品質な動画生成を実現した。しかしながら，diffusion models の性質上計算量が多く，他の手法と比較して生成時間が非常に遅い。

そこで本研究では，3D VQGAN を用いた LDM [12] に基づく動画生成モデルを提案する。3D VQGAN で符号化された潜在変数を学習することで，動画を直接生成する VDM [6] と比較して計算量を削減することができる。提案手法は，より高解像度の動画を生成するにもかかわらず，VDM と比較して約 9 倍の高速化を実現した。また，拡散モデルを用いているため他の動画生成手法と比較して，より高品質な動画生成を可能にした。

[1] 電気通信大学
[a)] kaji-r@mm.inf.uec.ac.jp
[b)] yanai@cs.uec.ac.jp

2. 関連研究

動画生成タスクは，実世界動画の分布を生成モデルによって近似し，学習データには存在しない高品質な動画を生成することを目的としている。He ら [8] の VAE ベースの手法は，動画は時間不変性とシーンダイナミクスの 2 つの要素に支配されているという考えに基づいている。

GAN ベースの手法である MoCoGAN [16] も，映像は動きとコンテンツに分けられると考え，異なるサンプリングノイズから動画を生成している。DVD-GAN [1] は，Spatial Discriminator と Temporal Discriminator を用いて写実性を向上している。DIGAN [17] は，動画生成に Implicit Neural Representations (INR) を用いており，空間座標と時間座標をそれぞれ操作することで，生成動画におけるダイナミクスを改善している。これら GAN ベースの手法は先行研究では主流のアプローチであり，動画の生成速度が速いなどの利点がある。しかし，GAN の学習構造上，学習の不安定さやモード崩壊などの問題がある。それに対して diffusion models ベースの手法では，これらの問題は起こらない。

自己回帰ベースの手法である VideoGPT [27] は，VQ-VAE [24] と Transformer [25] を用いた自己回帰動画生成モデルである。TATS [18] では，VQ-VAE を VQGAN [3] に置き換え，より大きなコードブックサイズと階層的な生成構造により，より高品質でより長い動画生成を実現している。これらの自己回帰ベースの手法では，動画フレームを生成するたびに損失が蓄積されるという問題がある。これに対し，diffusion models ベースの手法では全てのフレームを同時に生成するため，損失の蓄積はない。

Video Diffusion Models (VDM) [6] は，diffusion models を用いた動画生成方法である。VDM で用いられる 3D

U-Net のアーキテクチャは時空間分解されており，2D U-Net に Temporal Attention のみを追加して 3D に対応している。この手法は，画像生成に用いられている diffusion models を素直に動画領域へ拡張したものであり，非常に高品質な動画を生成することが可能である。しかしながら，一般に diffusion models は膨大な計算資源と先行研究より長いサンプリング時間を必要とする。これに対し，提案手法である VQ-VDM は動画を直接生成するのではなく，3D VQGAN で符号化された潜在変数を生成するように学習するため，少ない計算資源で高速なサンプリングが可能である。

3. 手法

本手法は，diffusion model を用いた潜在変数のサンプリングと，3D VQGAN を用いた動画から潜在変数への圧縮の 2 ステップに分けられる。まず，潜在変数と同じ大きさのガウスノイズをサンプリングし，diffusion model により潜在変数を生成する。その後，学習させた 3D VQGAN を用いて動画を生成する。

3.1 3D VQGAN による動画圧縮

diffusion model を用いて動画を直接学習するのは計算量が膨大になるため，3D VQGAN を用いて動画を低次元の潜在変数に圧縮する。圧縮に用いる 3D VQGAN の概要図を図 1 に示す。

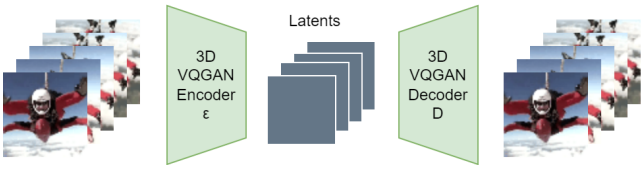

図 1 3D VQGAN

3D VQGAN Encoder $\mathcal{E}$ は，入力された動画 $\mathbf{x} \in \mathbb{R}^{3\times T\times H\times W}$ を時間方向に 4 倍，空間方向に 8 倍ダウンサンプルする。したがって，3 チャネル × 16 フレーム × 高さ 128 ピクセル × 幅 128 ピクセルの動画は 4 チャネル × 4 フレーム × 高さ 16 ピクセル × 幅 16 ピクセルの潜在ベクトル $\mathbf{z} \in \mathbb{R}^{4\times(T/4)\times(H/8)\times(W/8)}$ へエンコードされる。エンコードされた潜在ベクトルは量子化モジュールによって最も L2 距離が小さいコードブック埋め込みベクトル $\mathbf{e}$ に置き換えられ，量子化された潜在ベクトル $\mathbf{z}_q$ になる。量子化潜在ベクトルは 3D VQGAN Decoder $\mathcal{D}$ によってアップサンプリングされ，動画 $\hat{\mathbf{x}}$ を再構成する。これら $\mathcal{E}$ と $\mathcal{D}$ はどちらも 3 次元畳込みを用いて構成される。

損失関数は，Reconstruction Loss，VQ Loss，Discriminator Loss，Auxiliary Loss で構成される。Reconstruction Loss は，以下の式で表される。

$$\mathcal{L}_{\text{recon}} = ||\mathbf{x} - \hat{\mathbf{x}}||^2 + \mathcal{L}_{\text{LPIPS}}(\mathbf{x}, \hat{\mathbf{x}}) \tag{1}$$

$\mathcal{L}_{\text{LPIPS}}$ は VGG19 を用いた Perceptual Loss [28] である。Reconstruction Loss は入力動画と再構成動画の L2 Loss と Perceptual Loss から構成される。VQ Loss は以下の式で表される。

$$\mathcal{L}_{\text{vq}} = ||sg[\mathbf{z}_e] - \mathbf{e}||_2^2 + \beta||\mathbf{z}_e - sg[\mathbf{e}]||_2^2 \tag{2}$$

ここで $\mathbf{z}_e$ はエンコーダの出力，$\mathbf{e}$ はコードブック埋め込みである。VQ Loss の第 1 項は Codebook Loss，2 項目は Commitment Loss である。式 2 中の $sg[\,]$ (stop gradient) で囲まれた部分は勾配を逆伝播しないため，この損失関数はコードブック埋め込みのベクトルと Encoder $\mathcal{E}$ の出力を近づける損失になっている。Discriminator Loss は以下の式で表される。

$$\mathcal{L}_{\text{disc}} = logD(\mathbf{x}) + log(1 - D(\hat{\mathbf{x}})) \tag{3}$$

ここで D は Discriminator である。Discriminator は軽量な設計となっており，全ての層でダウンサンプルする 3 次元畳込みで構築されている。また，敵対的な学習を安定させるため [21] に従って以下の補助損失を追加している。

$$\mathcal{L}_{\text{disc\_aux}} = \sum_i ||D^{(i)}(\mathbf{x}) - D^{(i)}(\hat{\mathbf{x}})||^2 \tag{4}$$

ここで，$D^{(i)}$ は Discriminator の i 層における中間特徴量である。この損失により，Discriminator の最終層だけでなく中間出力に対して L2 Loss を最小化することで，学習をより安定させることができる。

したがって，最終的な損失関数は係数 λ によって次のようになる。

$$\mathcal{L} = \min_{\mathcal{E},\mathbf{q},\mathcal{D}} \max_{D} (\mathcal{L}_{\text{recon}} + \lambda_1\mathcal{L}_{\text{vq}} + \lambda_2\mathcal{L}_{\text{disc}} + \lambda_3\mathcal{L}_{\text{disc\_aux}}) \tag{5}$$

ここで，係数はそれぞれ $\lambda_1 = 1.0, \lambda_2 = 0.5, \lambda_3 = 1.0$ に設定した。また，VQ Loss のパラメータ β は 0.25 に設定した。3D VQGAN の学習時，$\mathcal{L}_{\text{disc}}$ は最初の 10000 イテレーションには含めていない。

3 次元畳込みのパディングには，Songwei ら [18] に従い，レプリケーションパディングを用いる。

3.2 Diffusion Models による動画生成

潜在変数の事前分布を diffusion model を用いて学習する。逆過程のパラメータ化に用いる 3D U-Net を図 2 に示す。

3D U-Net の各ブロックは，Residual Block，Spatial Attention，Temporal Attention で構成される。ダウンサンプリング処理では，潜在変数のフレームサイズを半分に圧縮し，チャンネル数を倍に増やす。アップサンプリング処理では，フレームサイズを倍にアップサンプリングし，チャンネル数を半分にする。すべてのダウンサンプリング側に

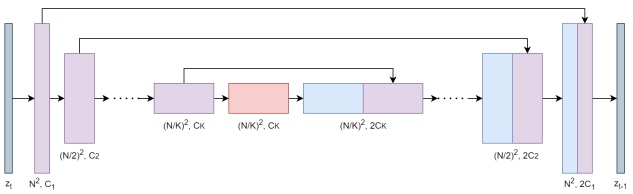

図 2 3D U-Net

おけるブロックの中間出力はスキップ接続され，アップサンプリング側のブロックに対してモジュール単位でチャネル方向に連結される。

Residual Block は，Group Normalize，SiLU，Conv3D によって構成されている。3D U-Net で条件付けされるタイムステップなどの Embedding は，Residual Block 中で加算することにより条件付けした。Conv3D では，3D VQGAN と同様である，実フレームのコピーであるレプリケーションパディングを使用した。

Temporal Attention は入力変数 $\mathbf{h} \in \mathbb{R}^{B\times C\times T\times H\times W}$ に対して軸の入れ替えを行い，$\mathbf{h}' \in \mathbb{R}^{B\times H\times W\times T\times C}$ とし，空間方向の軸を全てバッチ軸として扱い Attention Map を算出する。Attention Map には，自己フレーム以降のフレームを参照できないように，Causal Attention mask が適用される。

動画生成に用いる学習データセットの中には，FPS の値が固定で提供されていないものもあるため，モデル側で統一的に学習できる方法を用意する必要がある。そこで，タイムステップの条件付けに用いる埋め込みに，式 6 で計算した FPS の埋め込みを加算する。

$$emb_{\text{fps}} = \text{Linear}(\text{SiLU}(\text{Linear}(\text{PE}(fps)))) \tag{6}$$

ここで，PE は Vaswani ら [25] が提案した三角関数に基づく位置エンコーディングを用いている。式 6 はタイムステップ埋め込みと同じ形をとるが，別々の重みをもった線形層により計算される。

3.3 学習・生成

diffusion models の生成には，分類器フリーガイダンス [5] を用いる為，クラス条件付き生成モデルと無条件生成モデルを共同学習する必要がある。学習・生成には標準的な DDPM の定式化 [4] を用いるため，FPS 埋め込みを含む VQ-VDM の損失関数は以下となる。

$$\mathcal{L}_{uncondition}(\theta) := \mathbb{E}_{t,z_0,\epsilon}[||\epsilon - \epsilon_\theta(z_t,t,f)||^2] \tag{7}$$

$$\mathcal{L}_{class\_condition}(\theta) := \mathbb{E}_{t,z_0,\epsilon}[||\epsilon - \epsilon_\theta(z_t,t,f,c)||^2] \tag{8}$$

ここで，z_t はタイムステップ t における潜在変数 z，f は FPS 埋め込み，c はクラスである。学習では，ρ の割合で式 7 の無条件生成モデルを学習し，$1-\rho$ で式 8 のクラス条件付き生成モデルを学習する。

動画生成時の概要を図 3 に示す。学習済みの VQGAN Decoder と diffusion models を用いてガウスノイズから動画を生成する。

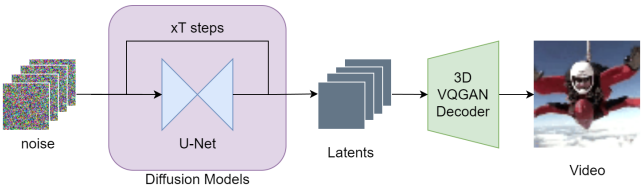

図 3 VQ-VDM

また，クラス条件付き生成時に条件付けるクラス c は，DDPM サンプリング時の各ステップにおいて分類器フリーガイダンスを用いて条件づける。分類器フリーガイダンスは，ガイダンススケール w を用いて以下のように表される。

$$\hat{\epsilon_\theta}(\mathbf{z}_t,t,f,c) = w\cdot(\epsilon_\theta(\mathbf{z}_t,t,f,c) - \epsilon_\theta(\mathbf{z}_t,t,f)) + \epsilon_\theta(\mathbf{z}_t,t,f) \tag{9}$$

4. 実験

4.1 設定

実験には，UCF-101 データセット [19] と Sky Time-lapse データセット [26] を用いた。これらのデータセットから 16 フレームの連続したシーケンスをクリップし，各フレームを 128x128 にリサイズして 3D VQGAN 及び diffusion model の学習に用いた。なお，VQ-VDM の学習は 3D VQGAN の学習を先に行った後，diffusion model の学習を行った。

評価指標には Inception score（IS）[15]，Fréchet video distance（FVD）[23]，Kernel Video Distance（KVD [23] を用いた。評価時には，それぞれ 10000，2048，2048 サンプルを生成し評価した。IS の測定には，Sports-1M データセット [9] で学習し，UCF-101 でファインチューニングされたた C3D モデル [22] を利用した。FVD と KVD の測定に関しては，Kinetics-400 データセット [10] で学習した I3D モデル [2] を利用した。

UCF-101 は 101 種類の行動クラスを含む 13320 本の動画からなるデータセットである。VQ-VDM の学習ではクラスと FPS 両方について条件付けを行った。また，共同学習のパラメータは $\rho = 0.5$ に設定した。

Sky Time-lapse は 5000 本の動画から構成されるデータセットであり，流れる雲や星空などのタイムラプス動画を含んでいる。画像群から構成される動画データセットであるため，クラスや FPS の条件付けは行わない。したがって，共同学習のパラメータは $\rho = 1.0$ に設定した。

4.2 定性・定量評価

Fig. 4 は UCF-101，Fig. 5 は Sky Time-lapse の写実的な動画生成結果である。どちらも時間的な整合性があり，128x128 の解像度で高品質に生成できていることがわかる。

表 1　UCF-101

Method	Resolution	Class	IS(↑)	FVD(↓)
TGAN [13]ICCV2017	64x64	Yes	15.83	-
MoCoGAN [16]CVPR2018	64x64	Yes	12.42	-
DVD-GAN [1]arXiv2019	128x128	Yes	27.38	-
TGANv2 [14]IJCV2020	128x128	Yes	28.87	1209
DIGAN [17]ICLR2022	128x128	No	32.70	577
CogVideo [7]arXiv2022	160x160	Yes	50.46	626
VDM [6]NIPS2022	64x64	No	57.00	-
TATS [18]ECCV2022	128x128	Yes	79.28	332
Ours	128x128	Yes	64.13	425

表 2　Sky Time-lapse

Method	Resolution	FVD(↓)	KVD(↓)
MoCoGAN-HD [20] ICLR2021	128x128	183.6	13.9
DIGAN [17] ICLR2022	128x128	114.6	6.8
TATS [18] ECCV2022	128x128	132.6	5.7
Ours	128x128	109.4	5.9

表 3　Sampling time

Method	Resolution	100 step time [s]
VDM [6]NIPS2022	16x64x64	35.26±2.43
Ours	16x128x128	3.95±0.01

図 6 は TATS と VQ-VDM の比較動画を示している。VQ-VDM は定量評価では TATS に劣っているが，生成結果の一部では両者の品質は視覚的に同等であることが分かる。また，TATS が 16 フレーム生成に対して度々時間的振動が起きているのに対し，我々の手法では時間に一貫性を持って生成できている。

条件付き生成についてベースラインとの比較を表 1 に示す。提案手法の定量評価は TATS に対して劣るものの，他の GAN ベースの手法や CogVideo [7] を上回るなど，競争力のある結果を示している。

表 2 は，Sky Time-lapse での定量的な評価を示している。本手法は，Sky Time-lapse において，TATS を含む全てのベースラインに対して，最先端の FVD を達成した。

図 4　UCF-101　動画生成結果

図 5　Sky Time-lapse 動画生成結果

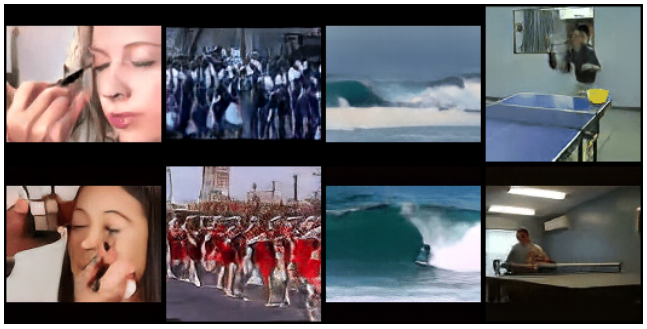

図 6　UCF-101 における TATS（上）と提案手法（下）のクラス条件付き動画生成結果。クラスは左から順に，”Apply eye makeup”, ”Band marching”, ”Surfing”, ”Table tennis shot”が指定されている。(注：Acrobat で本図は動画として再生される)

4.3　サンプリング効率

提案手法と VDM [6] のサンプリング時間を 100 タイムステップ単位で比較した。VDM の実装は公開されていないため，独自に再現実装を行い計測した。各手法とも 10 回ずつ測定し，その平均値と標準偏差を表 3 に示す。

表 3 によると，100 タイムステップにおける VDM のサンプリング時間は 35.26 秒であり，提案手法は 3.95 秒であった。したがって，提案手法は VDM に比べて 8.92 倍高速であることがわかる。また，提案手法はより大きな解像度の動画を生成することに注意されたい。提案手法は，VDM の約 9 倍の速さで，より高解像度の動画を生成することができる。

5.　まとめ

本研究では，3D VQGAN と diffusion models を用いた動画生成モデル，VQ-VDM を提案した。3D VQGAN で符号化された潜在変数を学習することで，動画を直接生成する VDM と比較して計算コストを大きく削減した。それにより，提案手法の VQ-VDM は，VDM [6] と比較して約 9 倍の速度で動画を生成することができる。また，VDM に対しては高解像度化により，他の動画生成手法に対しては diffusion model を用いることでより高品質な動画生成を実現した。

参考文献

[1] Aidan, C., Jeff, D. and Karen, S.: Adversarial Video Generation on Complex Datasets, *arXiv preprint arXiv:1907.06571* (2019).

[2] Carreira, J. and Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 6299–6308 (2017).

[3] Esser, P., Rombach, R. and Ommer, B.: Taming transformers for high-resolution image synthesis, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 12873–12883 (2021).

[4] Ho, J., Jain, A. and Abbeel, P.: Denoising diffusion probabilistic models, *Proc. of Advances in Neural Information Processing Systems*, Vol. 33, pp. 6840–6851 (2020).

[5] Ho, J. and Salimans, T.: Classifier-free diffusion guidance, *arXiv preprint arXiv:2207.12598* (2022).

[6] Ho, J., Salimans, T., Gritsenko, A. A., Chan, W., Norouzi, M. and Fleet, D. J.: Video Diffusion Models, *Proc. of Advances in Neural Information Processing Systems*.

[7] Hong, W., Ding, M., Zheng, W., Liu, X. and Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers, *arXiv preprint arXiv:2205.15868* (2022).

[8] Jiawei, H., Andreas, L., Joseph, M., Greg, M. and Leonid, S.: Probabilistic Video Generation using Holistic Attribute Control, *Proc.of European Conference on Computer Vision*, pp. 452–467 (2018).

[9] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L.: Large-scale video classification with convolutional neural networks, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 1725–1732 (2014).

[10] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P. et al.: The kinetics human action video dataset, *arXiv preprint arXiv:1705.06950* (2017).

[11] Le Moing, G., Ponce, J. and Schmid, C.: CCVS: Context-aware Controllable Video Synthesis, *Proc. of Advances in Neural Information Processing Systems*, Vol. 34, pp. 14042–14055 (2021).

[12] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B.: High-resolution image synthesis with latent diffusion models, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 10684–10695 (2022).

[13] Saito, M., Matsumoto, E. and Saito, S.: Temporal generative adversarial nets with singular value clipping, *Proc.of IEEE International Conference on Computer Vision*, pp. 2830–2839 (2017).

[14] Saito, M., Saito, S., Koyama, M. and Kobayashi, S.: Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan, Vol. 128, No. 10-11, pp. 2586–2606 (2020).

[15] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X.: Improved techniques for training GANs, *Proc. of Advances in Neural Information Processing Systems*, Vol. 29 (2016).

[16] Sergey, T., Ming-Yu, L., Xiaodong, Y. and Jan, K.: MoCoGAN: Decomposing Motion and Content for Video Generation, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 1526–1535 (2018).

[17] Sihyun, Y., Jihoon, T., Sangwoo, M., Hyunsu, K., Junho, K., Jung-Woo, H. and Jinwoo, S.: Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks, *Proc. of International Conference on Learning Representation* (2022).

[18] Songwei, G., Thomas, H., Harry, Y., Xi, Y., Guan, P., David, J., Jia-Bin, H. and Devi, P.: Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer, *Proc.of European Conference on Computer Vision* (2022).

[19] Soomro, K., Zamir, A. R. and Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild, *arXiv preprint arXiv:1212.0402* (2012).

[20] Tian, Y., Ren, J., Chai, M., Olszewski, K., Peng, X., Metaxas, D. N. and Tulyakov, S.: A Good Image Generator Is What You Need for High-Resolution Video Synthesis, *Proc. of International Conference on Learning Representation* (2021).

[21] Ting-Chun, W., Ming-Yu, L., Jun-Yan, Z., Andrew, T., Jan, K. and Bryan, C.: High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, *Proc.of IEEE Computer Vision and Pattern Recognition* (2018).

[22] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M.: Learning spatiotemporal features with 3D convolutional networks, *Proc.of IEEE International Conference on Computer Vision*, pp. 4489–4497 (2015).

[23] Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M. and Gelly, S.: Towards accurate generative models of video: A new metric & challenges, *arXiv preprint arXiv:1812.01717* (2018).

[24] Van Den Oord, A., Vinyals, O. et al.: Neural discrete representation learning, *Proc. of Advances in Neural Information Processing Systems*, Vol. 30 (2017).

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Proc. of Advances in Neural Information Processing Systems*, Vol. 30 (2017).

[26] Xiong, W., Luo, W., Ma, L., Liu, W. and Luo, J.: Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 2364–2373 (2018).

[27] Yan, W., Zhang, Y., Abbeel, P. and Srinivas, A.: VideoGPT: Video generation using VQ-VAE and transformers, *arXiv preprint arXiv:2104.10157* (2021).

[28] Yifan, L., Hao, C., Yu, C., Wei, Y. and Shen, C.: Generic Perceptual Loss for Modeling Structured Output Dependencies, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 5424–5432 (2021).