VQ-VDM: Video Diffusion Models with 3D VQGAN

Ryota Kaji
The University of Electro-Communications
Tokyo, Japan
kaji-r@mm.inf.uec.ac.jp

Keiji Yanai
The University of Electro-Communications
Tokyo, Japan
yanai@cs.uec.ac.jp

ABSTRACT

In recent years, deep generative models have achieved impressive performance such as realizing image generation that is indistinguishable from real images. Particularly, Latent Diffusion Models, one of the image generation models, have had a significant impact on society. Therefore, video generation is attracting attention as the next modality. However, video generation is more challenging than image generation due to the consideration of temporal consistency and the increase in computational complexity, since a video is a sequence of multiple frames. In this study, we propose a video generation model based on diffusion models employing 3D VQGAN, which is called VQ-VDM. The proposed model is about nine times faster than the Video Diffusion Models which directly generate videos, since our model generates a latent representation which is decoded into a video by a VQGAN decoder. Moreover, our model can generate higher quality video than prior video generation methods exclude state-of-the-art method.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**.

KEYWORDS

video generation, diffusion models, 3D VQGAN

ACM Reference Format:
Ryota Kaji and Keiji Yanai. 2023. VQ-VDM: Video Diffusion Models with 3D VQGAN. In *ACM Multimedia Asia 2023 (MMAsia '23), December 06–08, 2023, Tainan, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3595916.3626363

1 INTRODUCTION

In recent years, deep generative models have achieved impressive performance such as realizing image generation that is indistinguishable from real images. Particularly, Latent Diffusion Models [12], one of the image generation models, have had a significant impact on society. Therefore, video generation is attracting attention as the next modality. However, video generation is more challenging than image generation due to the consideration of temporal consistency and the increase in computational complexity, since a video is a sequence of multiple frames.

MMAsia '23, December 06–08, 2023, Tainan, Taiwan

ACM ISBN 979-8-4007-0205-1/23/12…$15.00
https://doi.org/10.1145/3595916.3626363

Although numerous methods using Transformers for autoregressive generation of video frames [11, 27] have been proposed, there is a problem of accumulating error every time generation occurs. In contrast, Video Diffusion Models (VDM) [6] can generate high-quality videos without accumulating error by simultaneously generating all frames. However, due to the architecture of diffusion models, it has very high computational complexity, resulting in slower generation time compared to other methods.

Therefore, in this study, we propose a video generation model based on Latent Diffusion Models [12] with 3D VQGAN. By learning about the latent representation encoded by 3D VQGAN, it is possible to reduce computational costs compared to VDM [6] which directly generate videos. Our proposed method is about nine times faster than VDM, although our method generates higher resolution videos. Moreover, our model can generate higher quality video than prior video generation methods exclude state-of-the-art method such as TATS [18].

2 RELATED WORKS

The video generation task is to generate high quality videos that do not exist in the training data by modeling the distribution of real world videos with a generative model. The VAE-based method of He *et al.* [8] is based on the idea that the video is governed by two factors: temporal invariance and scene dynamics. MoCoGAN [16], a GAN-based method, also considers that video can be divided into motion and content, and generates video from different sampling noises. DVD-GAN [1] consists of a Spatial Discriminator and a Temporal Discriminator. DIGAN [17] uses Implicit Neural Representations (INR) for video generation. By manipulating spatial and temporal coordinates, respectively, the dynamics of the generated video is improved. These GAN-based methods are the mainstream approach in the previous studies and have advantages such as fast video generation. However, due to the characteristics of GANs, there are issues such as unstable learning and mode collapse. The proposed method based on diffusion models solves these problems.

VideoGPT [27], an autoregressive-based method, is an autoregressive video generation model using VQ-VAE [24] and Transformer [25]. In TATS [18], VQ-VAE is replaced by VQGAN [3] to achieve higher quality and longer video generation with larger codebook size and hierarchical generation structure. These autoregressive-based methods have the problem of accumulating losses each time a video frame is generated. In contrast, the proposed method generates all frames simultaneously, so there is no loss accumulation.

Video Diffusion Models (VDM) [6] is a video generation method that uses diffusion models. The architecture of the 3D U-Net used in VDM is spatio-temporally decomposed, and only Temporal Attention is added to the 2D U-Net to support 3D. This method is a straightforward extension of diffusion models for image generation to the video domain, and is capable of generating very

high-quality video. However, in general, diffusion models requires huge computational resources and sampling time. In contrast, our proposed method is trained to generate latent variables decoded by 3D VQGAN instead of generating videos directly, which enables fast sampling with fewer computational resources.

2.1 Method

The method is divided into two steps: sampling latent vectors using diffusion models and encoding video frames into latent vectors using 3D VQGAN. First, Gaussian noise of the same size as the latent variable is sampled, and latent variables are generated by diffusion models. Then, the trained 3D VQGAN is used to generate the video.

2.2 Video Compression with 3D VQGAN

Since learning videos directly with Diffusion Models is computationally expensive, we use 3D VQGAN to compress videos into low-dimensional latent variables. A schematic diagram of the 3D VQGAN is shown in Figure 1.

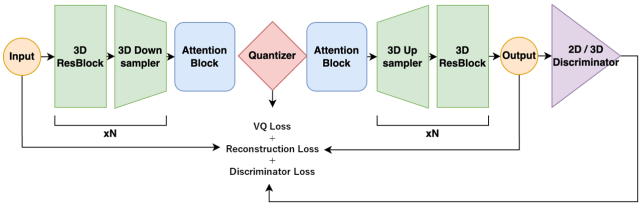

Figure 1: 3D VQGAN

The VQGAN encoder $\mathcal{E}$ downsamples an input video $x \in \mathbb{R}^{3\times T\times H\times W}$ one-fourth times in the temporal direction and one-eighth times in the spatial direction. Thus, a 3 channels × 16 frames × 128px height × 128px width video is encoded into a 4 channels × 4 frames × 16px height × 16px width latent vector $z \in \mathbb{R}^{4\times(T/4)\times(H/8)\times(W/8)}$. The encoded latent vector is replaced by the codebook embedding vector e by the quantization module $\mathbf{q}$, yielding the quantized latent vector $z_q = \mathbf{q}(z_e, e)$. The VQGAN decoder $\mathcal{D}$ generates a video by upsampling the latent vector. Both of the encoder and the decoder are composed of 3D convolution layers.

The loss function consists of Reconstruction Loss, VQ Loss, Discriminator Loss and Auxiliary Loss. The Reconstruction Loss is represented in the following equation:

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

$\mathcal{L}_{\text{LPIPS}}$ is the Perceptual Loss [28] using VGG19.

Reconstruction Loss is composed of L2 Loss and Perceptual Loss of video x and reconstructed video $\hat{x}$. The VQ Loss is represented as follows:

$$\mathcal{L}_{\text{vq}} = ||sg[z_e(x)] - e||_2^2 + \beta||z_e(x) - sg[e]||_2^2 \tag{2}$$

where z_e is the output of the Encoder and e is the codebook embedding. The first item of VQ Loss is Codebook Loss and the second item is Commitment Loss. The part enclosed by $sg[\ ]$ (stop gradient) does not back propagate the gradient. Therefore, this loss function is a loss that brings the codebook embedding and the output of E closer to each other.

Next, the Discriminator Loss is expressed by

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

where D is the Discriminator; the Discriminator is a lightweight design, consisting of a 3D convolution that downsamples in all layers.

In addition, the following Discriminator auxiliary losses are added to stabilize the learning according to [21].

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

where $D^{(i)}$ is the intermediate features in layer i of the Discriminator. This loss can be used to make learning more stable by minimizing the L2 loss in the intermediate features as well as in the final layer of the Discriminator.

Thus, the final loss function is the following with coefficient λ.

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

where $\lambda_1 = 1.0$, $\lambda_2 = 0.5$, and $\lambda_3 = 1.0$. The parameter β for VQ Loss was set to 0.25. In learning 3D VQGAN, $\mathcal{L}_{\text{disc}}$ does not include the first 10,000 iterations in the loss function.

We use replication padding for the padding of Conv3D in the temporal direction. It is a copy of the real frame rather than zero padding, by following the method of Songwei *et al.* [18].

2.3 Video Generation with Diffusion Models

We train video diffusion models which output latent variables decoded into a video by the 3D VQGAN decoder. Since video's pixels are correspond to nearby pixels in spatio-temporal directions, we use a 3D U-Net in the inverse process of diffusion models as shown in Figure 2.

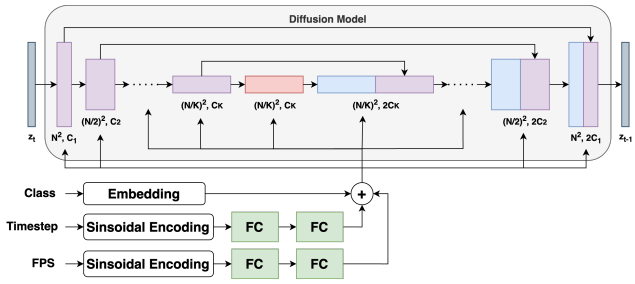

Figure 2: 3D U-Net

Each block of the 3D U-Net consists of Residual Block, Spatial Attention, and Temporal Attention. In the downsampling process, the video frame size is compressed to half the size and the number of channels is increased. In the upsampling process, the frame size is upsampled by a factor of two and the number of the channels is decresed. The intermediate outputs of the blocks in all downsampling processes are skip-connected and concatenated in module block units in the upsampling process.

The composition within the Residual Block was implemented in the order of GroupNormalize, SiLU, and Conv3D. Embeddings such as time steps conditioned by 3D U-Net were conditioned by adding them in the middle of the Residual Block. Conv3D uses replication

padding, which is a copy of the real frame, instead of zero padding as in 3D VQGAN.

Temporal Attention performs axis swapping on the input variable $h \in \mathbb{R}^{B\times C\times T\times H\times W}$ to form $h' \in \mathbb{R}^{B\times H\times W\times T\times C}$, and treats all axes in the spatial direction as batch axes and calculates an Attention Map. Causal Attention mask is applied to the Attention Map so that frames after the self-frame cannot be referenced.

Since some training datasets used for video generation are not provided with fixed FPS values, it is necessary to provide a method that enables uniform training on the model side. Therefore, we also add the FPS embedding calculated by Eq. 6 to the embedding used for time step conditioning.

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

where PE means positional encoding with trigonometric-based periodic functions proposed by Vaswani *et al.* [25]. Eq. 6 has the same form as time step embedding, but is computed using Linear modules with different weights.

2.4 Training and Sampling

In order to generate class-conditional videos with classifier guidance, an additional classifier models must be trained. Therefore, diffusion models jointly train class-conditional and unconditional training models in order to use classifier-free guidance [5]. Since we use the standard DDPM formulation [4] for training diffusion models, the loss functions of the proposed method, including FPS embedding, are represented in Eq. 7 and Eq. 8.

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

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

where z_t is the latent variable z at the time step t, f is the FPS value, and c is the class. At the training time, unconditional learning is performed with probability ρ in Eq. 7 and class conditional learning is performed with probability $1-\rho$ in Eq. 8.

In the testing time, the flow of video generation by our model is shown in Figure 3. We do not need the VQGAN encoder for video generation.

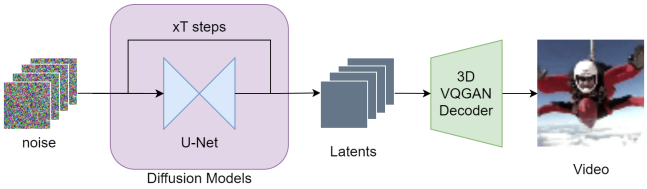

Figure 3: Video generation with 3D VQGAN and Diffusion Model

The class c to be generated during class conditional learning is also prepared and generated by performing classifier free guidance in a step within the diffusion model. The classifier-free guidance is expressed using the guidance scale w as in Eq. 9.

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

3 EXPERIMENTS

3.1 Settings

For the experiments, we utilized the UCF-101 dataset [19] and Sky Time-lapse dataset [26]. We randomly extracted continuous sequences of 16 frames from these datasets and resized them to 128x128 frame size for training 3D VQGAN and the diffusion models. Note that the training of our VQ-VDM consists two steps where we perform training of the 3D VQGAN encoder/decoder first, and training of the latent diffusion models with frozen 3D VQGAN later.

For the measurement of Inception score (IS) [15], Fréchet video distance (FVD) [23] and Kernel Video Distance (KVD) [23], which are evaluation indices, 10000, 2048 and 2048 samples were generated and evaluated, respectively. For IS measurements, we utilized a C3D model [22] trained on Sports-1M dataset [9] and fine-tuned on UCF-101. As for FVD and KVD measurements, we utilized an I3D model [2] trained on Kinetics-400 dataset [10].

UCF-101 is a dataset consisting of 13320 short videos of people performing 101 different actions; we trained the VQ-VDM with the Class and FPS conditions. The parameter ρ in the joint learning was set to 0.5.

Sky Time-lapse is a dataset consisting of 5000 videos of dynamic sky scenes, such as the cloudy sky with moving clouds, and the starry sky with moving stars. We trained the VQ-VDM without FPS or class conditions. Therefore, The parameter ρ in the joint learning was set to 1.

Figure 4: Generated videos on UCF-101.

Figure 5: Generated videos on the Sky Time-Lapse dataset.

Figure 6: Comparison between TATS (top) and Ours (bottom) on UCF-101 dataset.

Table 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

Table 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

Table 3: Sampling time

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

3.2 The Quality of Video Generation

Figure 4 and Figure 5 show photorealistic video generation for UCF-101 and Sky Time-lapse, respectively. Both are temporally consistent and plausibly generated at a resolution of 128x128.

Figure 6 shows a comparison video between TATS and Ours. Although VQ-VDM is inferior to TATS in quantitative evaluation, it can be seen that in some of the generation results, the quality of the two are visually equivalent. Moreover, while TATS frequently experiences temporal oscillations for the generation of 16 frames, our method can consistently generate video frames over time.

Table 1 shows a comparison with other class-conditional generation methods. Although the quantitative evaluation of the proposed method scores poorly against TATS, it shows competitive results, outperforming all the other GAN-based methods and CogVideo [7].

Table 2 shows the quantitative evaluation in Sky Time-lapse. Our method archives state-of-the-art FVD over all the baselines including TATS on the Sky Time-lapse.

3.3 Sampling Efficiency

We compared the sampling rates of the proposed method and VDM [6] in units of 100 time steps. Since the original implementation of VDM is not publicly available, we performed the measurements using a reproduced implementation. Ten measurements were taken for each method, and the mean and standard deviation are shown in Table 3.

Table 3 shows that the sampling time of VDM at 100 time steps is 35.26 seconds, while the proposed method takes 3.95 seconds. Therefore, the proposed method is 8.92 times faster than VDM. It should also be noted that the proposed method generates images with a larger resolution. The proposed method can generate videos with higher resolution about nine times faster than VDM.

4 CONCLUSION

In this study, we proposed a video generation model VQ-VDM based on diffusion models with 3D VQGAN. By learning about the latent variables encoded by 3D VQGAN, it was able to reduce computational costs compared to Video Diffusion Models which directly generate videos. So, our proposed method VQ-VDM generated high-quality video about nine times faster than VDM [6]. Also, the higher quality videos were able to be generated by using higher resolution against VDM, and by being based on diffusion models against the other video generation methods.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers, 21H05812, 22H00540, 22H00548, and 22K19808.

REFERENCES

[1] Clark Aidan, Donahue Jeff, and Simonyan Karen. 2019. Adversarial Video Generation on Complex Datasets. *arXiv preprint arXiv:1907.06571* (2019).

[2] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 6299–6308.

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 12873–12883.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proc. of Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

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

[6] Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. [n. d.]. Video Diffusion Models. In *Proc. of Advances in Neural Information Processing Systems*.

[7] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2022. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868* (2022).

[8] He Jiawei, Lehrmann Andreas, Marino Joseph, Mori Greg, and Sigal Leonid. 2018. Probabilistic Video Generation using Holistic Attribute Control. In *Proc.of European Conference on Computer Vision*. 452–467.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 1725–1732.

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).

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

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 10684–10695.

[13] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. 2017. Temporal generative adversarial nets with singular value clipping. In *Proc.of IEEE International Conference on Computer Vision*. 2830–2839.

[14] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. 2020. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision* 128, 10-11 (2020), 2586–2606.

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. *Proc. of Advances in Neural Information Processing Systems* 29 (2016).

[16] Tulyakov Sergey, Liu Ming-Yu, Yang Xiaodong, and Kautz Jan. 2018. MoCoGAN: Decomposing Motion and Content for Video Generation. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 1526–1535.

[17] Yu Sihyun, Tack Jihoon, Mo Sangwoo, Kim Hyunsu, Kima Junho, Ha Jung-Woo, and Shin Jinwoo. 2022. Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks. In *Proc. of International Conference on Learning Representation*.

[18] Ge Songwei, Hayes Thomas, Yang Harry, Yin Xi, Pang Guan, Jacobs David, Huang Jia-Bin, and Parikh Devi. 2022. Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer. In *Proc.of European Conference on Computer Vision*.

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

[20] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. 2021. A Good Image Generator Is What You Need for High-Resolution Video Synthesis. In *Proc. of International Conference on Learning Representation*.

[21] Wang Ting-Chun, Liu Ming-Yu, Zhu Jun-Yan, Tao Andrew, Kautz Jan, and Catanzaro Bryan. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proc.of IEEE Computer Vision and Pattern Recognition*.

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3D convolutional networks. In *Proc.of IEEE International Conference on Computer Vision*. 4489–4497.

[23] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717* (2018).

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

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Proc. of Advances in Neural Information Processing Systems* 30 (2017).

[26] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 2364–2373.

[27] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157* (2021).

[28] Liu Yifan, Chen Hao, Chen Yu, Yin Wei, and Chunhua Shen. 2021. Generic Perceptual Loss for Modeling Structured Output Dependencies. In *Proc.of IEEE Computer Vision and Pattern Recognition*. 5424–5432.