Paying Attention to Video Generation

Rishika Bhagwatkar
VNIT, Nagpur
rishika.vnit@gmail.com

Khurshed Fitter
VNIT, Nagpur
khurshedpf@gmail.com

Saketh Bachu
VNIT, Nagpur
saketh7000@gmail.com

Akshay Kulkarni
Indian Institute of Science
akshayk.vnit@gmail.com

Shital Chiddarwar
VNIT, Nagpur
s.chiddarwar@gmail.com

Abstract

Video generation is a challenging research topic which has been tackled by a variety of methods including Generative Adversarial Networks (GANs), Variational Autoencoders (VAE), optical flow and autoregressive models. However, most of the existing works model the task as image manipulation and learn pixel-level transforms. In contrast, we propose a latent vector manipulation approach using sequential models, particularly the Generative Pre-trained Transformer (GPT). Further, we propose a novel Attention-based Discretized Autoencoder (ADAE) which learns a finite-sized codebook that serves as a basis for latent space representations of frames, to be modelled by the sequential model. To tackle the reduced resolution or the diversity bottleneck caused by the finite codebook, we propose attention-based soft-alignment instead of a hard distance-based choice for sampling the latent vectors. We also intend to extensively evaluate the proposed approach on the BAIR Robot Pushing, Sky Time-lapse and Dinosaur Game datasets and compare with state-of-the-art (SOTA) approaches.

1 Introduction

Machine Learning (ML) paradigms and model architectures are designed and optimized, keeping in mind the task that is intended to be solved or tackled [3, 27]. This, in turn introduces passive yet influential notions for domain-specific conventions and methodologies for modelling ML systems. While some notions like the direct dependence of a model's performance and the amount of relevant training data [3, 46] hold true for most cases, other factors like model architectures and evaluation metrics differ drastically across tasks. A prime example of this is the stark difference between models that deal with images and those that deal with temporal data like speech and text. We propose an attempt to harness the best of both worlds for video generation tasks.

Generating videos from initial frame(s), on the face of it, seems to be an image manipulation task which could be tackled satisfactorily using models based on or employing generative paradigms, mainly Generative Adversarial Networks (GANs) [56, 17, 8, 28]. Such techniques usually focus on learning pixel-level transforms [25, 34, 52, 56] and try to generate succeeding frames using learnable motion or pixel flow characteristics [33, 35]. However, an alternative stance could be to formulate video generation as a sequence modelling task [32, 9, 18] on a finite set of latent vectors. The recent advancements in Natural Language Processing (NLP) [5, 13, 37, 38, 51] reflect the success of sequence modelling on latent space representations of language tokens.

Most of the current literature on video generation use the former approach. However, we choose the latter as our initial stance to leverage the advancements in sequential modelling. Further, most video generation approaches use pixel-level predictions and model and generate images as a series of pixels

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

[49, 7]. We focus on latent vector manipulation using sequential models, namely the Generative Pre-trained Transformer (GPT) [5, 37, 38]. Sequence models perform well at learning the plausible neighbours or successors of a given set of language tokens. We propose to extend the idea such that the model learns to predict the latent space representation of a frame, by conditioning over a set of already available (predicted or provided) latent vectors of previous frames.

We propose a novel Attention-based Discretized Autoencoder (ADAE) which learns a discretized finite set of vectors called the codebook. The codebook is then used to generate frame embeddings, analogous to word embeddings [2, 31] in NLP. We use a discretized Autoencoder (AE) to ensure a finite latent basis size, analogous to vocabularies in NLP. We intend to validate our approach on the BAIR Robot Pushing [14], Sky Time-lapse [55] and Dinosaur Game datasets and compare the performance with state-of-the-art (SOTA) approaches.

The main contributions of our work are:
1. A novel ADAE to represent an entire frame as an encoded vector referred to as a frame embedding.
2. Utilizing the sequential modelling prowess of the GPT to sequentially model frame embeddings.

2 Related work

2.1 Video generation

Videos can be generated from text, initial frames, complete videos and even pixel transformations on existing frames [25, 52, 34, 6, 15]. Video generation started off as a deterministic modelling task and later shifted towards approaches involving GANs [16, 56, 17, 29, 53, 43, 47, 1, 42, 8, 28], Variational Autoencoders (VAEs) [10, 17, 50, 41, 12, 11, 24, 21], optical flow[33, 35] and autoregressive models[40, 45, 44, 23, 54, 20]. Using only VAEs does not provide the best of results, and using GANs usually makes the models computationally expensive and difficult to train.

The current state of the art approaches like TRIVD-GAN-FP [28] and autoregressive models like PixelSnail [7, 49] deliver great quality frames but come at a heavy price of training and deployment overheads and do not seem to perform well on resolutions beyond 64×64 pixels. A similar issue arises with the attention and subscaling-based Video Transformer [54, 30].

The latest work, Latent Video Transformer [39] generates videos by generating latent space representations of frames, by coupling a Vector Quantized Variational Autoencoder (VQ-VAE) [50] and an entire Transformer [51]. We use only the decoder blocks of the transformer, stacked on top of each other to form a GPT-based sequential model along with our ADAE. The choice of decoder blocks of the Transformer, over the encoder blocks, is due to the presence of masked self-attention which forces the model to condition on the previously available (provided or generated) frame embeddings. While masking off random samples (BERT-based paradigm [13]) may enrich the robustness of the generated embeddings, we prefer the GPT-based approach to learn sequential modelling.

2.2 Latent space representation

The robustness and effectiveness of most NLP models depend heavily on the robustness and contextual capacity of the token embeddings used [2]. Word embeddings are the most commonly used latent space or embedded representations of tokens. The GPT and BERT-based approaches are prime examples of successful NLP models banking on robust embeddings. Other approaches like ELMo [36], further decompose words into characters and hence do not need to worry about a fixed vocabulary space as opposed to models like GPT and BERT which use a fixed size embedding space and outliers are usually assigned a special token, resulting in a single referencing index for each token.

Latent space representations for images are usually continuous d-dimensional vector spaces (d is the bottleneck or encoding dimension) and are learned using autoencoder-based models. The VQ-VAE [50] and VQ-VAE-2 [41] models, learn a discretized, finite subset of the d-dimensional latent space called the codebook, the size of which is fixed a priori. However, the discrete vectors in the codebook, contain mainly positional information about the latent space representation of an image.

We propose a novel Attention-based Discretized Autoencoder (ADAE) that discretizes the continuous latent space $\mathbb{R}^d$ into a finite codebook. However, unlike the VQ-VAE models [50, 41], the codebook in our model forms a basis that is used to encode an entire frame into a single vector using an attention-based soft-aligned linear combination [51, 4] of the codebook vectors.

3 Approach

3.1 Attention-based discretized autoencoder

In typical reconstruction AEs, the encoder $E(.;\theta)$ learns to map an input $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ to a vector belonging to a continuous latent space $\mathbb{R}^d$, i.e. $E(\mathbf{x};\theta) = \mathbf{z} \in \mathbb{R}^d$ where d is the bottleneck dimension. The decoder jointly learns a mapping from $\mathbf{z} \in \mathbb{R}^d$ to $D(\mathbf{z};\phi) = \mathbf{x}' \in \mathbb{R}^{H \times W \times C}$. The loss criterion is simply the Mean Squared Error (MSE) loss between the output of the decoder and the input.

$$\mathcal{L}(\mathbf{x}, \mathbf{x}'; \theta, \phi) = ||\mathbf{x} - \mathbf{x}'||_2^2 \tag{1}$$

Such AEs can not be directly paired with language models due to the latter's requirement of finite vocabulary sizes. We discretize the latent space of the AE into a finite set $\xi = \{\mathbf{e}_i\}\ \forall\ i \in [1, N]$, forming a codebook of N vectors with each vector $\mathbf{e}_i \in \mathbb{R}^d$. The latent vector $\mathbf{z}$ is now a linear combination of the codebook vectors.

$$\mathbf{z} = \sum_{i \in I} a_i \mathbf{e}_i \tag{2}$$

Where, I is the set of top-k indices, ranked in decreasing order of normalized attention scores $a_i = A(E(\mathbf{x};\theta), \mathbf{e}_i)$ as per the attention metric $A(.,.)$. This introduces a soft-alignment metric as opposed to a choice based on strict minimal distance. If we replace the linear combination by

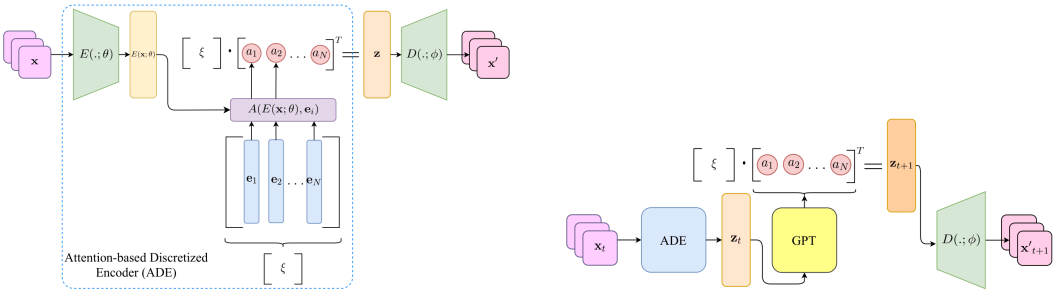
$$\mathbf{z} = \mathbf{e}_j \text{ such that } ||E(\mathbf{x};\theta) - \mathbf{e}_j||_2^2 < ||E(\mathbf{x};\theta) - \mathbf{e}_i||_2^2\ \forall\ i \in [1, N] - \{j\} \tag{3}$$

then similar inputs may produce identical outputs leading to a diversity bottleneck. We tackle this by introducing an attention-based soft-alignment [4, 51]. We formulate our loss function as

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}; \theta, \phi, \mathbf{e}_i) = ||\mathbf{y} - \hat{\mathbf{y}}||_2^2 + ||\text{sg}[E(\mathbf{x};\theta)] - \mathbf{e}_i||_2^2 + \beta||E(\mathbf{x};\theta) - \text{sg}[\mathbf{e}_i]||_2^2\ \forall\ i \in I \tag{4}$$

The loss function is similar to that proposed in [41] and the first term is simply the MSE loss between the label and decoded output. The stop-gradient operation is denoted by sg[.] and β is a hyperparameter which incorporates the relative reluctance of changing the codebook corresponding to the encoder's output.

The fundamental difference between our Attention-based Discretized Autoencoder (ADAE) (shown in Fig. 1a) and the VQ-VAE [50, 41] models, is that the latter focuses on learning a grid-like latent space representation for an input, with the latent vectors being sampled from a discrete codebook and the reconstruction is done pixel by pixel. Our model on the other hand, learns to encode an entire frame as a linear combination of discrete vectors from a learnable codebook. The reconstruction is not autoregressive (as opposed to [50, 41]) and the decoder learns to model the distribution $p(\mathbf{x} \mid \mathbf{z})$.



(a) (b)

Figure 1: (a) A block diagram of our ADAE model. The attention metric $A(.,.)$ calculates attention scores between with the encoder's output and codebook vectors. The normalized scores are used as coefficients of the corresponding codebook vectors to get the frame embedding ($\mathbf{z}$). The frame embedding is then passed to the decoder for reconstruction. (b) A block diagram of our approach. The GPT is given t frame embeddings ($\mathbf{z}_{\{1,2,...t\}}$) and it predicts t probability distributions, each corresponding to the model's prediction for the succeeding frame embedding, for each frame in the input set. **Note:** In both the figures, we have shown only 1 RGB frame, with top-k filtering disabled.

3.2 Generative pre-trained transformer

A GPT [37, 38, 5] is a large, generative, attention-based sequential model, based on the Transformer model [51]. However, the model utilizes only the decoder blocks of the Transformer model and employs masked self-attention to learn sequential generation of tokens.

We propose to interpose a GPT-based model between the encoder and decoder parts of our proposed ADAE (as shown in Fig. 1b) to learn sequential modelling over the latent vectors. The output of such models is usually a probability distribution vector of length equal to the number of elements in the vocabulary. The indices of top-k probability scores from this output distribution are used to form the set I. This set contains indices of the attention scores to be used as coefficients of the codebook vectors to form a linear combination as per Eq 2. The result of this linear combination is then passed to the ADAE's decoder for reconstruction.

4 Experiments

4.1 Datasets

- **BAIR Robot Pushing Dataset:** This dataset [14] contains about 59,000 videos (≈ 1.5 million video frames) of robots interacting with objects mainly via pushing motions. It is divided into a training set (57,000 videos), an unseen test set and a seen test set (1,250 videos each).
- **Sky Time-lapse Dataset:** This dataset [55] contains about 38,000 videos of 32 frames each. The videos are time lapse shots of the sky with clouds and stars moving across the frames and are sourced mainly from YouTube.
- **Dinosaur Game Dataset** We have curated over 5 hours of video, containing clips of Google Chrome's "Dinosaur Game". The frames can be binarized into black and white frames, hence providing us with a comfortably learnable dataset to test prototypes.

We resize all the frames to $64 \times 64 \times 3$ pixels for a fair comparison with previous works, although we shall further experiment using higher resolutions to evaluate the scalability. Further, we introduce a downsampling parameter m_d to downsample or skip frames from videos, in cases where consecutive frames are too similar. This helps ensure that the model does not end up learning an approximate identity map for some cases.

4.2 Teacher forcing

Teacher forcing ratio (r_{tf}) is the probability of using the ground truth as an input to a sequential model, instead of its own prediction at a given time stamp. Usually, it is set to 1 while training and 0 while testing. However, we set it to 1 for a few initial frames (up to one fourth of video length) and then set it as a hyperparameter which determines the trade-off between ease of convergence while training and robustness while sampling.

4.3 Model architectures

Attention-based Discretized Autoencoder: We propose an ADAE which discretizes the continuous latent space $\mathbb{R}^d$ into a fixed size codebook. This makes it almost intuitive to use its codebook as a basis for the vocabulary or embedding space of a language model, like the GPT employed by us.

We intend to implement broadly two classes of autoencoder architectures as a part of our ablation study. The encoders are based on the architectures employed in VQ-VAE [50, 41] and ResNet [19] along with their respective symmetrical (mirror image) inversions as the decoder networks.

Further we propose varying the codebook size N, from about 1,000 to 65,000 (in steps of $\approx$ 8,000) and encoding/bottleneck dimension d, from around 200 to more than 2,000 (in steps of $\approx$ 250). We intend to compare amongst and choose from a few attention metrics, namely, dot product attention [4], energy based attention and query-key-value-based attention [51] and shall be tuning the value of k for choosing the top-k attention scores, for each metric.

Generative Pre-trained Transformer: The GPT model forms the sequential kernel of our model and is interposed between the encoder and the decoder of our ADAE. It takes in attended frame

embeddings ($\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_{t-1} \in \mathbb{R}^d$) and outputs the attention scores to be used as coefficients for the linear combinations of the codebook vectors, to get the next frame embedding ($\mathbf{z}_t \in \mathbb{R}^d$) as per Eq 2.

We shall vary the depth of the model from about 10 to 40 stacked, modified Transformer [51] decoder blocks, similar to the architectures followed in [37, 38]. Further, we will be using learnable positional embeddings as opposed to sinusoidal ones used in [51].

4.4 Training and evaluation metrics

As a part of our ablation experiments, we shall train our ADAE and GPT models jointly as well as separately. When training separately, we shall train the ADAE as a reconstruction autoencoder ($\mathbf{x} = \mathbf{y}$) first and then fine-tune it while training the GPT model. While training jointly, we would train both the networks from scratch. However, while training jointly, the labels would be the succeeding frame as opposed to the same frame in the disjoint training case.

Finally, we intend to evaluate our model on BAIR Robot Pushing and Sky Time-lapse datasets and compare it with other models as shown in Tables 1 and 2.

Table 1: Frechet Video Distance (FVD) [48] scores of various models, referred from [39]

Method	FVD (↓)
SVP-FP[12]	315.5
CDNA[14]	296.5
LVT[39]	125.8 ± 2.9
SV2P[12]	262.5
SAVP[24]	116.4
DVD-GAN-FP[8]	109.8
TriVD-GAN-FP[28]	103.3
Video Transformer[54]	94 ± 2
Ours	-

Table 2: Peak Signal to Noise Ratio (PSNR), Strucutral Similarity Index (SSIM) [58] and Flow Mean Squared Error (FlowMSE) [26] scores of various models, referred from [57]

Method	PSNR (↑)	SSIM (↑)	FlowMSE (↓)
MoCoGAN[47]	23.867	0.849	1.365
MDGAN[22]	23.042	0.822	1.406
DTVNet[57]	29.917	0.916	1.275
Ours	-	-	-

5 Discussions and conclusion

We consider potential issues that could arise in the proposed approach. One of them could be mode-collapse where our model generates sequences of almost indistinguishable (as per human standards) frames. In such a case, we would increase the downsampling rate in order to ensure enough diversity between consecutive frames. On the other hand, if our model is unable to converge, then we may consider reducing the downsampling rate in an attempt to make variations more subtle.

Another plausible issue could be unwanted blurry regions or artifacts appearing after reconstruction, owing to the fact that the latent space representation focuses more on the entire frame rather than local groups of pixels. We could tackle this by dividing frames into grids and learning multiple codebooks.

In this work, we propose a latent vector manipulation approach to video generation. Specifically, we introduce a novel Attention-based Discretized Autoencoder (ADAE) to learn a finite set of vectors. This set acts as the basis for the frame embeddings, which are sequentially modelled using a GPT.

References

[1] D. Acharya, Z. Huang, D. P. Paudel, and L. V. Gool. Towards high resolution video generation with progressive growing of sliced wasserstein gans, 2018.

[2] F. Almeida and G. Xexéo. Word embeddings: A survey, 2019.

[3] M. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Essen, A. Awwal, and V. Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8:292, 03 2019. doi: 10.3390/electronics8030292.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

[6] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang. Deep video generation, prediction and completion of human action sequences. *Lecture Notes in Computer Science*, page 374–390, 2018. ISSN 1611-3349. doi: 10.1007/978-3-030-01216-8_23. URL `http://dx.doi.org/10.1007/978-3-030-01216-8_23`.

[7] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel. Pixelsnail: An improved autoregressive generative model, 2017.

[8] A. Clark, J. Donahue, and K. Simonyan. Adversarial video generation on complex datasets, 2019.

[9] Y. Dandi, A. Das, S. Singhal, V. P. Namboodiri, and P. Rai. Jointly trained image and video generation using residual vectors, 2019.

[10] Y. Dandi, A. Das, S. Singhal, V. P. Namboodiri, and P. Rai. Jointly trained image and video generation using residual vectors, 2019.

[11] E. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. 2017.

[12] E. Denton and R. Fergus. Stochastic video generation with a learned prior. 2018.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[14] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction, 2016.

[15] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network, 2019.

[16] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. 2014.

[17] S. Gur, S. Benaim, and L. Wolf. Hierarchical patch vae-gan: Generating diverse videos from a single sample, 2020.

[18] J. He, A. Lehrmann, J. Marino, G. Mori, and L. Sigal. Probabilistic video generation using holistic attribute control, 2018.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[20] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans. Axial attention in multidimensional transformers, 2019.

[21] J.-T. Hsieh, B. Liu, D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. 2018.

[22] Y. Intrator, G. Katz, and A. Shabtai. Mdgan: Boosting anomaly detection using multi-discriminator generative adversarial networks, 2018.

[23] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. 2016.

[24] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction, 2018.

[25] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin. Video generation from text, 2017.

[26] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang. Flow-grounded spatial-temporal video prediction from still images. *CoRR*, abs/1807.09755, 2018. URL `http://arxiv.org/abs/1807.09755`.

[27] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48 – 56, 2017. ISSN 2468-502X. doi: https://doi.org/10.1016/j.visinf.2017.01.006. URL `http://www.sciencedirect.com/science/article/pii/S2468502X17300086`.

[28] P. Luc, A. Clark, S. Dieleman, D. de Las Casas, Y. Doron, A. Cassirer, and K. Simonyan. Transformation-based adversarial video prediction on large-scale data, 2020.

[29] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. 2016.

[30] J. Menick and N. Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling, 2018.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[32] G. Mittal, T. Marwah, and V. N. Balasubramanian. Sync-draw. *Proceedings of the 2017 ACM on Multimedia Conference - MM '17*, 2017. doi: 10.1145/3123266.3123309. URL `http://dx.doi.org/10.1145/3123266.3123309`.

[33] K. Ohnishi, S. Yamamoto, Y. Ushiku, and T. Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. 2017.

[34] N. Parmar, A. Vaswani, J. Uszkoreit, Łukasz Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer, 2018.

[35] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. 2016.

[36] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations, 2018.

[37] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training.

[38] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.

[39] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev. Latent video transformer, 2020.

[40] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. 2016.

[41] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019.

[42] M. Saito and S. Saito. Tganv2: Efficient training of large models for video generation with multiple subsampling layers. 11 2018.

[43] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping, 2017.

[44] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. 2015.

[45] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. 2016.

[46] H. SUG. Performance of machine learning algorithms and diversity in data. *MATEC Web of Conferences*, 210:04019, 01 2018. doi: 10.1051/matecconf/201821004019.

[47] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation, 2017.

[48] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, abs/1812.01717, 2018. URL `http://arxiv.org/abs/1812.01717`.

[49] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016.

[50] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning, 2018.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[52] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2992–3000, 2017.

[53] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics, 2016.

[54] D. Weissenborn, O. Täckström, and J. Uszkoreit. Scaling autoregressive video models. 2020.

[55] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks, 2018.

[56] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. volume 97 of *Proceedings of Machine Learning Research*, pages 7354–7363. PMLR, 2019. URL `http://proceedings.mlr.press/v97/zhang19d.html`.

[57] J. Zhang, C. Xu, L. Liu, M. Wang, X. Wu, Y. Liu, and Y. Jiang. Dtvnet: Dynamic time-lapse video generation via single still image, 2020.

[58] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.