

GODIVA: Generating Open-Domain Videos from nAtural Descriptions

Chenfei Wu^{1*} Lun Huang^{2*} Qianxi Zhang¹ Binyang Li¹
 Lei Ji¹ Fan Yang¹ Guillermo Sapiro² Nan Duan^{1†}

¹Microsoft Research Asia ²Duke University

{chewu, qianxi.zhang, binyang.li, leiji, fanyang, nanduan}@microsoft.com

{lun.huang, guillermo.sapiro}@duke.edu

Abstract

Generating videos from text is a challenging task due to its high computational requirements for training and infinite possible answers for evaluation. Existing works typically experiment on simple or small datasets, where the generalization ability is quite limited. In this work, we propose GODIVA, an open-domain text-to-video pretrained model that can generate videos from text in an auto-regressive manner using a three-dimensional sparse attention mechanism. We pretrain our model on Howto100M, a large-scale text-video dataset that contains more than 136 million text-video pairs. Experiments show that GODIVA not only can be fine-tuned on downstream video generation tasks, but also has a good zero-shot capability on unseen texts. We also propose a new metric called Relative Matching (RM) to automatically evaluate the video generation quality. Several challenges are listed and discussed as future work.

1 Introduction

“Creativity is a fundamental feature of human intelligence, and a challenge for AI.” [3]. Recent advances in image and text generation have shown great creativity of machines, including GANs [4, 32], VAEs [22, 26], RNNs [30, 16] and Self-Attentions [19]. However, it is still a challenge for the AI agent to create videos, especially for real-world diversity ones. Generating videos requires the machine to not only create a large number of pixels but also ensure semantic coherence among them.

We take up these challenges of generating videos from the text, namely the text-to-video generation (T2V) task. Given a natural description, T2V requires the machine to understand it and create a semantically consistent video. Although not much, there are still some works studying this topic using GANs. Firstly, [11] and [15] use GANs with 3D convolutions to generate fixed-length low-resolution videos. Then, [2] uses a conditional filter to generate videos of varying lengths. [6] integrates LSTM cells with 2D convolutional networks to model both frame quality and temporal coherence. However, these works conduct experiments on simple or small datasets, where generalization ability is limited.

Besides GAN-based methods, VQ-VAE is another promising research direction and has been shown great progress in generating images and videos, especially DALL-E [21] for text-to-image generation. It successfully generates high-quality images from text. In this paper, we turn to the more challenging text-to-video generation task, where both spatial and temporal coherence of the visual information must be taken into account. Some other recent works [20, 28, 33] apply VQ-VAE for the task of video prediction—forecasting future video frames given the past. Concurrently, we are the first to design a VQ-VAE pretrained model for the T2V task.

*Both authors contributed equally to this research.

†Corresponding author.

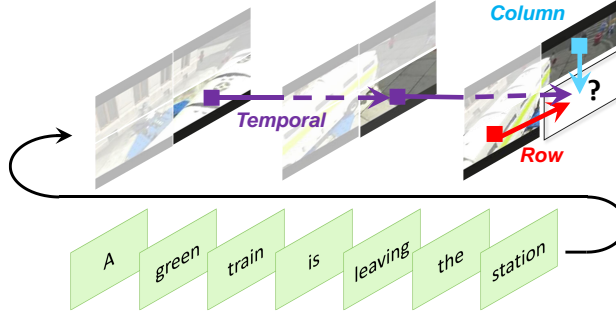


Figure 1: A simple illustration of our GODIVA model with three-dimensional sparse attention mechanism for text-to-video generation task. The video is auto-regressively predicted with the consideration of four aspects: the input text, same position of the previous generated frames, same rows on the same frame, same columns on the same column.

In this paper, we propose GODIVA to generate open-domain videos from text using VQ-VAE and three-dimensional sparse attention. Firstly, a VQ-VAE auto-encoder is trained to represent continuous video pixels with discrete video tokens. Then, a three-dimensional sparse attention model is trained using language as input and the discrete video tokens as labels to generate videos, considering temporal, column, and row information, as shown in Fig. 1.

Our contributions are three-fold: (1) We proposed an open-domain text-to-video pretrained model with a three-dimensional sparse attention mechanism, which can significantly reduce the computation cost; (2) We proposed a new Relative Matching (RM) Metric, which can evaluate both visual quality and semantic match for video generation; (3) We pretrained our proposed model on the HowTo100M dataset and demonstrated its video generation capabilities on both fine-tuning and zero-shot settings.

2 Related Works

In this section, we briefly review related works for video generation. We first review the video-to-video generation task, which has been widely studied in recent years. Then we review text-to-image and text-to-video generation. We also highlight the differences between previous models and ours.

2.1 Video-to-video generation

Most video generation studies focus on video prediction tasks. Input the first few frames of a video, the video prediction task predicts the following frames of a video. We call it video-to-video (V2V) generation for comparison with text-to-video (T2V) generation.

Existing video-to-video generation can be divided into three categories. Firstly, deterministic methods directly model the tractable density using RNNs and CNNs and exploit both spatial and temporal information of a video. [8] used ConvLSTM as the basic block to predict pixel motions instead of values. [12] proposed PredNet, which predicts future frames by incorporating previous predictions. Further, [29] proposed stacked ConvLSTM, which shares the hidden state among the layers in the stack. Recently, [5] proposed ContextVP, which aggregates contextual information for each pixel in all possible directions. Secondly, the GAN-based methods avoid explicit density function and use a generator to generate videos and a discriminator to judge if the video is generated. [27] proposed VGAN, which is the first model to generate videos using GANs. After that, [23] proposed TGAN, which separates a spatiotemporal generator into time-series and space models to generate videos. Then, [25] proposed MoCoGAN, which produces videos more efficiently by decomposing the latent space into the motion and the content subspaces. Recently, [24] proposed TGAN2, which trains each sub-generator with its specific discriminator. Thirdly, VAE methods model the approximate density by capturing a low-dimensional representation z and optimize a lower bound on the likelihood. [1] proposed SV2P to capture sequence uncertainty in a single set of latent variables kept fixed for each predicted sequence. Then, [7] proposed SVG. They used a per-step latent variable (SVG-FP) and a variant with a learned prior (SVG-LP), which makes the prior at a certain timestep a function of

previous frames. Recently, [20] proposed a Latent Video Transformer, which encodes each frame of a video and predicts the discrete video features. GAN-based models.

Our model can be categorized into the VAE-based models. Different from recent VQ-VAE based works such as Latent Video Transformer [20], our work focus on text-to-video generation task instead of video-to-video generation task. We also incorporate a three-dimensional sparse attention to model the sparse relations between visual tokens.

2.2 Text-to-image generation

Text-to-image generation has been widely researched in recent years [17]. The most similar work is DALL-E [21] which successfully generates high-quality images from text. In this paper, we turn to a more challenging text-to-video generation task, which considers both spatial and temporal coherence of the visual information.

2.3 Text-to-video generation

Different from video-to-video generation, text-to-video generation has been few studied. Firstly, [11] and [15] use GANs with 3D convolutions to generate fixed-length low-resolution videos. Then, [2] uses a conditional filter to generate videos of varying lengths. [6] integrates LSTM cells with 2D convolutional networks to model both frame quality and temporal coherence.

Most text-to-video generation methods use GAN-based methods while our model incorporates a VQ-VAE for this task. As far as we know, this is the first paper that uses VQ-VAE for this task.

3 The GODIVA Method

Let x be an observable video, and we use a discrete latent code z to represent it, which has a lower dimension. In the following, we show how to represent x using z with VQ-VAE [26] in section 3.1, and generate videos from the text by modeling $P(z|t)$ in section 3.2, where t denotes the given text.

3.1 Frame-wise video auto-encoder

For an input video $x \in \mathbb{R}^{L \times H \times W \times C}$ with L frames, the l th frame $x^{(l)}$ is encoded in Eq. (1).

$$y^{(l)} = E(x^{(l)}), \quad (1)$$

where $y^{(l)} \in \mathbb{R}^{(hw) \times d_B}$ is the latent variable with $h \times w$ regions. Then, $y^{(l)}$ is quantized to get a more compact latent representation, as denoted in Eq. (2).

$$z_i^{(l)} = \arg \min_j \|y_i^{(l)} - B_j\|^2, \quad (2)$$

where $B \in \mathbb{R}^{K \times D}$ is the codebook where the i th region of the latent variable $y_i^{(l)} \in \mathbb{R}^{d_B}$ is searched to find the nearest indexes $z^{(l)} \in \mathbb{R}^{hw}$. Then, $z^{(l)}$ is embedded by the codebook in Eq. (3).

$$b^{(l)} = B[z^{(l)}], \quad (3)$$

where $b^{(l)} \in \mathbb{R}^{(hw) \times d_B}$ is the embedding of $z^{(l)}$. Next, $b^{(l)}$ is sent to a decoder that reconstructs the original video frame, as shown in Eq. (4).

$$\hat{x}^{(l)} = D(b^{(l)}), \quad (4)$$

where $\hat{x}^{(l)} \in \mathbb{R}^{H \times W \times C}$ is the reconstructed frame. Finally, the VQ-VAE can be trained in the objective denoted in Eq. (5).

$$\mathcal{L}^{VQ-VAE} = \frac{1}{L} \sum_{l=1}^L \|x^{(l)} - \hat{x}^{(l)}\|_2^2 + \|sg[y^{(l)}] - b^{(l)}\|_2^2 + \beta \|y^{(l)} - sg[b^{(l)}]\|_2^2, \quad (5)$$

where the three items are reconstruction loss, codebook loss and commitment loss respectively. β is the weighting factor. sg denotes the stop gradient operator.

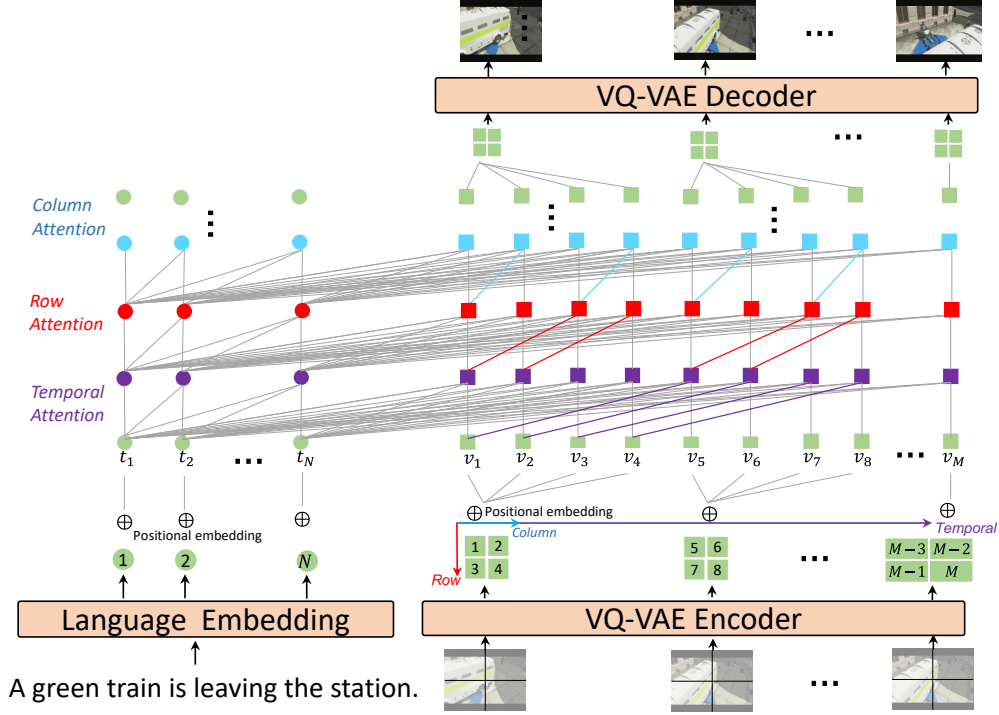


Figure 2: Illustration of GODIVA. To generate a video of $W \times H = 64 \times 64$ pixels and $L = 10$ frames, the size of the VQ-VAE discrete representation is $w \times h = 16 \times 16$. Thus the model needs to generate a total of $M = 2560$ tokens. When generating the 8th visual token, our model only pays attention to the same position token in the previous frame (4th visual token) or the previous row or column token in the same frame (7th and 6th visual token).

3.2 GODIVA video generator

In this section, we focus on generating videos from text by modeling the conditional probability $P(z|t)$. Given an input text $t \in \mathbb{R}^N$ with N tokens, the embeddings of the text are calculated with the consideration of positional information, as denoted in Eq. (6):

$$t^e = E^{(t)}[word_{idx}] + P^{(t)}[0, 1, \dots, N - 1], \quad (6)$$

where $E^{(t)} \in \mathbb{R}^{S \times D}$ is the text embedding matrix, S is the text vocabulary size, $P^{(t)} \in \mathbb{R}^{N \times D}$ is the text positional embedding matrix, and $t^e \in \mathbb{R}^{N \times D}$ is the final text embedding. We use the pretrained VQ-VAE Encoder (Eq. (1~3)) to encode each frame the ground-truth videos in Eq. (7):

$$b^{(l)} = B \left[\arg \min_j ||E(x^{(l)})_j - B_j||^2 \right] \quad (7)$$

where the ground-truth video sequence $x \in \mathbb{R}^{L \times H \times W \times C}$ is encoded into a sequence of discrete latent visual token embeddings $b \in \mathbb{R}^{M \times d_B}$. $M = L \times h \times w$ is the maximum of visual tokens. Then, the video embeddings are calculated with the consideration of positional information in Eq. (8):

$$v^e = \text{Linear}(b) + P^{(v)}[0, 1, \dots, M - 1], \quad (8)$$

where the Linear layer mapped z to $\text{Linear}(z) \in \mathbb{R}^{M \times D}$, which has the same dimension as t^e . $P^{(v)} \in \mathbb{R}^{M \times D}$ is the video positional embedding matrix. $v^e \in \mathbb{R}^{M \times D}$ is the final ground-truth video embedding. Now, a decoder can be trained to generate videos in an auto-regressive way, as denoted in Eq. (9):

$$v_m^e = \text{Decoder}(t^e, v_{<m}^e) \quad (9)$$

where $v_m^e \in \mathbb{R}^D$ is the transformed visual embeddings at step m . Note that M is a large number, especially for real-world videos. To reduce computation, We introduce a three-dimensional sparse

attention in Eq. (10):

$$\begin{aligned} h_{i,j,l}^{(T)} &= SA^{(T)}(v_{i,j,<l}^e), \\ h_{i,j,l}^{(R)} &= SA^{(R)}(v_{<i,j,l}^e), \\ h_{i,j,l}^{(C)} &= SA^{(C)}(v_{i,<j,l}^e). \end{aligned} \quad (10)$$

where SA denotes the self-attention layer. T, R, C denotes Temporal, Row, and Column respectively. $h_{i,j,l}^{(T)}, h_{i,j,l}^{(R)}, h_{i,j,l}^{(C)} \in \mathbb{R}^D$ are the hidden states at step (i, j, l) . Note that we change the notation of the step from m to (i, j, l) for a clearer expression of these three sparse attentions. As we can see from Eq. (10), the sparse attention for each axis only attends to the indexes in the previous axis, instead of the indexes in the global axis. Thus the computation complexity reduces from $O((Lhw)^2)$ to $O(Lhw(L+h+w))$. Then, the three attention layers are stacked alternately, as denoted in Eq. (11).

$$h_{ijl} = \underbrace{\left[SA^{(T)}, SA^{(R)}, SA^{(C)}, SA^{(T)}, \dots, SA^{(C)} \right]}_{R \text{ layers}} (h_{<=i,<=j,<=l}), \quad (11)$$

where $h \in \mathbb{R}^{M \times D}$ is the output hidden states of these stacked attention layers. Then, h is fed to a Linear layer to get the logits of the predicted visual tokens, as denoted in Eq. (12).

$$P(\hat{z}|t) = \text{softmax}(\text{Linear}(h)) \quad (12)$$

where the linear layer maps the dimension of h into the VQ-VAE vocabulary size $\text{Linear}(h) \in \mathbb{R}^{M \times K}$. $\hat{z} \in \mathbb{R}^M$ is the predicted visual tokens. Finally, the model is trained using cross-entropy loss, as denoted in Eq. (13).

$$\mathcal{L} = -\frac{1}{M} \sum_{i=1}^M z_i \log(P(\hat{z}|t)) \quad (13)$$

4 Experiments

4.1 Datasets

We pretrain GODIVA on Howto100M dataset [13], which consists of more than 136 million text-video pairs. We then evaluate our model on the MSR-VTT dataset [31], which consists of 10000 video clips with 20 human-annotated captions for each of them. We also train GODIVA from scratch on the Moving Mnist dataset [14] and Double Moving Mnist dataset, both were automatically generated from the Mnist dataset [10]. The original Moving Mnist dataset has two motions: up-down and left-right. In this paper, we follow [6] and add four more directions: move left then right, move right then left, move up then down and move down then up.

4.2 Evaluation Metrics

It is challenging to quantitatively evaluate the performance of text-to-video generation models. This is mainly due to two reasons: Firstly, given a piece of text, there are countless corresponding videos. It is hard to objectively judge which one is better. Secondly, an evaluation metric should consider both visual quality and semantic matching of the generated video. To handle these challenges, we introduce two kind of metrics: A CLIP Similarity (SIM) metric and a Relative Matching (RM) metric for automatic evaluation in Sec. 4.2.1, A Visual Realisticity (VR) and Semantic Consistency (SC) metric for human evaluation in Sec. 4.2.2.

4.2.1 Automatic Evaluation Metrics

The key factor for judging the quality of the generated video is whether it matches the text. Using a pretrained visual-language matching model will inevitably introduce the bias of its domain data. Thanks to recent zero-shot work CLIP [18], which provides a strong zero-shot ability for visual-text matching and thus reduced those data biases. Since CLIP is pretrained between image and text, we

calculate the similarities between text and each frame of the video and then take the average value as the semantic matching in Eq. (14).

$$SIM(t, \hat{v}) = \frac{1}{L} \sum_{l=1}^L CLIP(t, \hat{v}^{(l)}), \quad (14)$$

where t denotes the input text. \hat{v} is the predicted video with L frames. Note that SIM only provides the absolute score of the semantic match. To further reduce the influence of the CLIP model, we divide SIM by the similarity between text and the ground-truth video to get a relative matching score, which we call Relative Matching (RM) metric, as denoted in Eq. (15).

$$RM(t, \hat{v}) = \frac{SIM(t, \hat{v})}{SIM(t, v)}, \quad (15)$$

where v is the ground-truth video with L frames. The RM metric reveals the domain-independent generation quality since if the generated video is more relevant to the text, it will obviously have a higher RM value. If the generated video is not relevant to the text or has a low quality, the RM value will be lower.

4.2.2 Human Evaluation Metrics

To conduct a human evaluation, we invite 200 evaluators as testees and conduct a human evaluation. Let $\{M_1, M_2, \dots, M_N\}$ be a set of models to evaluate, T be the number of samples in the test set. To reduce the subjective biases, we ask the testees to compare the Visual Realisticity (VR) and Semantic Consistency (SC) of two videos (v_i, v_j) generated from two models (M_i, M_j) with the same query q respectively, as denoted in Eq. (16)~(17).

$$VR(M_i) = \frac{1}{NT} \sum_{t=1}^{N,T} r_{ij}^{(t)}, \quad r_{ij}^{(t)} = \begin{cases} 1, & v_i \text{ is more realistic than } v_j \text{ for sample } t, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

$$SC(M_i) = \frac{1}{NT} \sum_{t=1}^{N,T} c_{ij}^{(t)}, \quad c_{ij}^{(t)} = \begin{cases} 1, & v_i \text{ is more consistent with } q \text{ than } v_j \text{ for sample } t, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

4.3 Implementation details

In Sec. 3.1, the size of the input video is $L = 10, H = 64, W = 64, C = 3$. Both the encoder E in Eq. (1) and the decoder D in Eq. (4) are implemented with two CNN layers. The kernel size is 4 and stride is 2. Thus the latent variable has the size of $h \times w = 16 \times 16$. The latent variable dimension $d_B = 128$. The VQ-VAE codebook has a total of $K = 10000$ tokens. The VQ-VAE model is pretrained on ImageNet with a learning rate of $1e-3$ and batch size 32. Note that when we conduct experiments on Moving Mnist Dataset, we train another VQ-VAE on this dataset. We found this will lead to better generation performance.

In Sec. 3.2, the input text has a maximum length of $N = 35$. The dimension $D = 1024$. The maximum size of the visual tokens is $M = 2560$. The Self-Att in Eq. (10) uses 16 attention heads. The GODIVA has a total of $R = 12$ layers in Eq. (11). The GODIVA model is pretrained on the Howto100M dataset with 64 V100 GPUs. It is finetuned on the MSR-VTT dataset with 8 V100 GPUs. Both settings have the same batch size of 32 and a learning rate of $5e-4$. More details, including the source code, will come soon in Github.

4.4 Qualitative Results

We qualitatively evaluate our model from two aspects. Firstly, we evaluate the zero-shot ability of our model by comparing GODIVA to two prior approaches: T2V [11] and TFGAN [2]. Both approaches are trained on the real-world dataset created from a clean-up of Kinetics [9] and Youtube videos. As shown in Fig. (3), for the same query "Play golf on grass", T2V successfully generates the grass and action of "playing golf" in a resolution of 64×64 , but the result looks blurry (see the first row). TFGAN was successfully trained on a resolution of 128×128 and generates a higher quality result



Figure 3: Comparison of samples from our model to prior approaches on real-word dataset. $n \times n$ represents the number of pixels of the video frame.

Model	Input Sentence: Digit 9 is moving down then up.	Input Sentence: Digit 7 moves right then left while digit 3 moves down then up.
VGAN[27]		
SyncDraw[14]		
TGANs[15]		
MocoGAN[25]		
IRC-GAN[6]		
GODIVA(ours)		

Figure 4: Comparison of samples from our model to prior approaches on Moving Mnist and Double Moving Mnist dataset. Note that VGAN, TGANs and MocoGAN are a modified version by [6] to support text-to-video generation.

(see the second row). Both T2V and TFGAN generate text-related videos, but the generated frames are in a single scene and the difference between frames is not significant. This limits the creativity of neural models. Interestingly, GODIVA not only generates text-related videos but also changing scenes (see the third and fourth row). For example, GODIVA(64×64) first shows the grass field, then it gives the athlete a close-up shot, and finally the action of hitting the golf ball. Note that GODIVA(64×64) and GODIVA(128×128) are different models, thus they generate totally different videos. The last row gives another (128×128) resolution results generated by GODIVA. In total, GODIVA is able to generate videos with clear frames and coherent semantics.

Secondly, we evaluate the unseen video generation ability by comparing GODIVA to several GAN-based approaches. The models in Fig. (4) are trained on the Moving MNIST dataset [14] and Double Moving MNIST dataset [14] respectively. Note that there is no video in the training set that shows "Digit 9 is moving down than up", but there are some variant samples such as "Digit 9 is moving left and right" or "Digit 3 is moving down than up". GODIVA successfully generates semantic correct results(see the last row of the left part). This shows GODIVA learns to capture the semantic

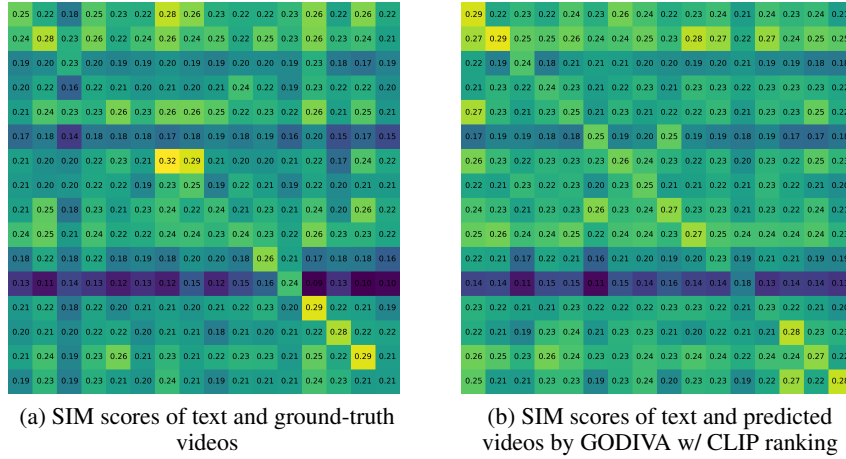


Figure 5: SIM scores calculated in 16 random samples. In sub figure (a) and (b), row denotes the query and the column denotes the videos.

Table 1: Qualitative Results on MSR-VTT dataset. All values reported are multiplied by 100.

Model	Automatic Evaluation		Human Evaluation	
	SIM	RM	VR	SC
GT	24.23	100	-	-
GODIVA (6 layer)	21.45	86.94	9.38	9.38
GODIVA w/o Row Attention	21.23	85.95	31.25	40.63
GODIVA w/o Temporal Attention	21.52	87.44	40.63	43.75
GODIVA w/o Column Attention	22.08	89.20	46.88	46.88
GODIVA	22.82	93.48	81.25	78.13
GODIVA w/ CLIP ranking	24.02	98.34	88.12	81.25

alignment between text and video, rather than just search videos in the training set to find the one most similar to the input sentence. Besides, GODIVA generates high-quality videos, even compared with the state-of-the-art IRC-GAN [14] approach. The digit "9" is both spatially clear and temporally consistent. Another example in Double Moving MNIST on the right shows a similar phenomenon.

4.5 Quantitative Results

We quantitatively evaluate our model through both automatic and human metrics. To validate the effectiveness of RM metric, we first draw the SIM scores between text and ground-truth videos in Fig. 5(a). It can be seen from the diagonal that SIM is basically able to distinguish semantically similar videos from other videos. Tab. 1 shows the ablation experiments of different settings of GODIVA. We pretrain GODIVA on Howto100M dataset and finetune it on MSR-VTT dataset. We find that SIM and RM have the same trend with the human evaluation metrics. The first row shows the results between input text and ground-truth videos. The second row shows that sufficient scale for GODIVA is crucial. GODIVA (6 layer) shows worse performance than the default GODIVA setting (12 layer). The next three rows show the effectiveness of the three dimensional attentions. We found that the Row Attention is the most important. Following DALL-E [21], we randomly sample 32 times in the top 10 probabilities in Eq. (12) during inference and using CLIP ranking to find the best generated video. The performance is then significantly improved to 98.34 in RM metric.

5 Conclusions

In this paper, we propose a three-dimensional sparse attention to generate open-domain videos from natural descriptions using VQ-VAE discrete visual tokens. We also propose a new Relative Matching

metric to automatically evaluate generation quality. Experiments show that our model not only can be fine-tuned on downstream video generation tasks, but also has a good zero-shot capability on unseen texts. However, there are still several challenges: Firstly, it is still a great challenge to generate long videos with high resolution. When generating only 64×64 resolution videos with 10 frames, the total of visual tokens M already becomes 2560. Secondly, automatically evaluating text-to-video generation task remains a challenge. In the future, video-based CLIP metric may give more accurate results for semantic consistency for text and videos. Thirdly, GAN-based methods show a great potential for text-to-video generation (see Fig. (4)), their generative abilities for open-domain dataset remain a good research direction.

References

- [1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.
- [2] Yogesh Balaji, Martin Renqiang Min, Bing Bai, Rama Chellappa, and Hans Peter Graf. Conditional GAN with Discriminative Filter Generation for Text-to-Video Synthesis. In *IJCAI*, pages 1995–2001, 2019.
- [3] Margaret A. Boden. Creativity and artificial intelligence. *Artificial intelligence*, 103(1-2):347–356, 1998.
- [4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv:1809.11096 [cs, stat]*, February 2019.
- [5] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 753–769, 2018.
- [6] Kangle Deng, Tianyi Fei, Xin Huang, and Yuxin Peng. IRC-GAN: Introspective Recurrent Convolutional GAN for Text-to-video Generation. In *IJCAI*, pages 2216–2222, 2019.
- [7] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183. PMLR, 2018.
- [8] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *arXiv preprint arXiv:1605.07157*, 2016.
- [9] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, and Paul Natsev. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [10] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [11] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [12] William Lotter, Gabriel Kreiman, and David Cox. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. *arXiv:1605.08104 [cs, q-bio]*, February 2017.
- [13] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2630–2640, 2019.
- [14] Gaurav Mittal, Tanya Marwah, and Vineeth N. Balasubramanian. Sync-draw: Automatic video generation using deep recurrent attentive architectures. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1096–1104, 2017.
- [15] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1789–1798, 2017.
- [16] Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*, 2016.

- [17] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. MirrorGAN: Learning Text-to-image Generation by Redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019.
- [18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.
- [19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.
- [20] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent Video Transformer. *arXiv preprint arXiv:2006.10704*, 2020.
- [21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv:2102.12092 [cs]*, February 2021.
- [22] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.
- [23] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2830–2839, 2017.
- [24] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train Sparsely, Generate Densely: Memory-Efficient Unsupervised Training of High-Resolution Temporal GAN. *International Journal of Computer Vision*, 128:2586–2606, 2020.
- [25] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018.
- [26] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.
- [27] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *arXiv preprint arXiv:1609.02612*, 2016.
- [28] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting Video with VQVAE. *arXiv preprint arXiv:2103.01950*, 2021.
- [29] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. In *NIPS*, January 2017.
- [30] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.
- [31] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016.
- [32] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.
- [33] Yunzhi Zhang, Wilson Yan, Pieter Abbeel, and Aravind Srinivas. VideoGen: Generative Modeling of Videos using VQ-VAE and Transformers. September 2020.

6 Appendix

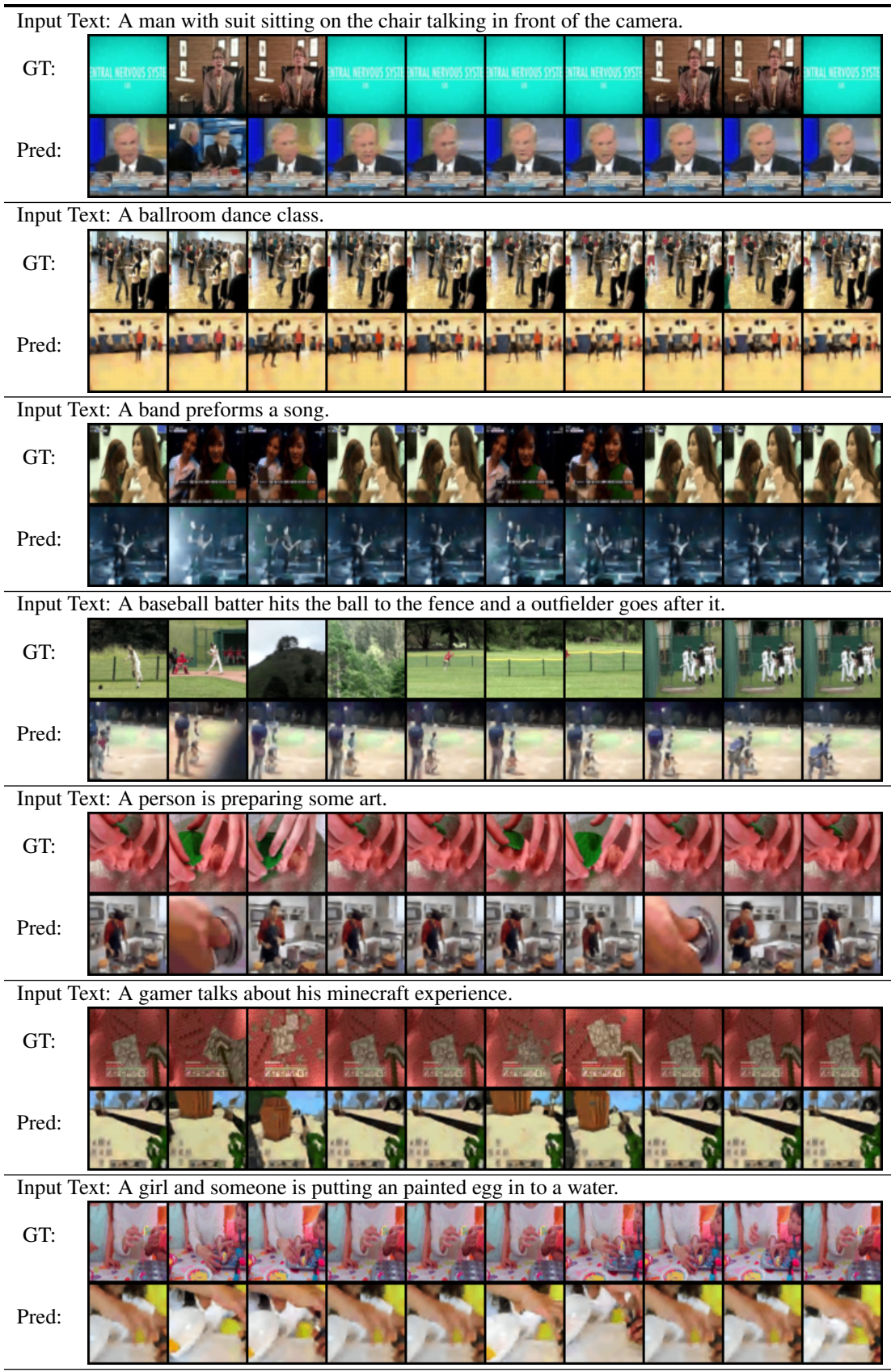


Figure 6: More samples generated by GODIVIA.



Figure 7: More samples generated by GODIVIA.