Int. j. inf. tecnol. (January 2024) 16(1):221–236
https://doi.org/10.1007/s41870-023-01468-4

ORIGINAL RESEARCH

VTM-GAN: video-text matcher based generative adversarial network for generating videos from textual description

Rayeesa Mehmood[1] · Rumaan Bashir[1] · Kaiser J. Giri[1]

Received: 19 March 2023 / Accepted: 28 August 2023 / Published online: 16 September 2023

Abstract Text-to-video synthesis has garnered significant attention as a challenging task in the domain of vision computing. With the advent of unsupervised learning techniques, text-to-video synthesis has become more feasible. In this context, Generative Adversarial Network (GAN)-based training networks have emerged as the leading unsupervised deep learning methods, exhibiting promising results. However, achieving visual quality, temporal coherence, and semantic consistency between the generated video and textual descriptions remains a considerable challenge. In this paper, we propose a novel approach called Video-Text Matcher (VTM) based GAN for text-to-video synthesis. The proposed VTM is based on Contrastive Language-Image Pre-training (CLIP) but with modifications. It incorporates both global sentence-level and fine-grained word-level information to calculate the similarity between the generated video and the provided textual descriptions. Unlike CLIP, which focuses on matching losses at the global sentence-image level only, our VTM includes a word-region level loss to enhance the fine granularity consistency between the text and video. We evaluate our proposed approach using the Single Digit Bouncing MNIST GIFs (SBMG) dataset and conduct both qualitative and quantitative analyses. The results demonstrate that our proposed method generates appealing videos that align well with the given textual descriptions, showcasing the effectiveness of our approach for text-to-video synthesis.

Keywords Contrastive language–image pre-training · Text-to-video generation · Generative adversarial network · Video-text matcher

1 Introduction

In the current landscape of text-to-video generation with deep learning techniques, GAN-based methods have garnered widespread recognition. These algorithms have shown impressive results in various tasks, including stock price prediction [1], intrusion detection [2], spam detection [3], data augmentation [4] and many more. The generator network in GANs imparts it with an ability of automatic visual content generation. This network learns the probabilistic density estimation from the training data distribution, allowing it to generate samples within the boundaries of the data it was trained on. Through adversarial learning between the generator and discriminator networks, GANs can produce photo-realistic samples that are virtually indistinguishable from real samples. In computer vision, GANs find applications in diverse fields, such as image super-resolution [5, 6], image deblurring [7], human face synthesis [8, 9], high-resolution human faces [10, 11], face sketch synthesis [12], and image in-painting [13, 14]. Their versatility and capability to handle complex visual data make GANs a valuable asset in a wide range of tasks. Further advancements in this field has led to the expansion of the unconditioned GAN towards the conditional generative models. In the original GANs [15], there is no control over the type of samples to be generated. It uses the training data distribution to create data samples

✉ Rumaan Bashir
rumaan.bashir@islamicuniversity.edu.in

Rayeesa Mehmood
rayeesa.mehmood@iust.ac.in

Kaiser J. Giri
kaiser.giri@islamicuniversity.edu.in

[1] Department of Computer Science, Islamic University of Science and Technology, Kashmir, Jammu and Kashmir 192122, India

for any category. Furthermore, throughout the sampling process, the resulting data samples might not fully reflect all potential changes in the training set. In contrast, conditional GANs (CGAN) [16] introduced the concept of using additional conditional information in the model to exert control over the generated sample output. This supplementary information could take the form of class labels, text, sketches, and more. By incorporating this conditioning, the process of image generation becomes intention-based, allowing for the creation of variational data samples. Conditional image generation has become increasingly popular, particularly in scenarios that involve latent embedding. Notable applications include text-to-image generation [17–19], image translation [20, 21], and manipulation based on linguistic instructions [22]. These techniques empower the model to produce images that align with specific conditions, adding a new dimension of control and flexibility to the generative process.

GAN-based methods have also demonstrated superior performance in video synthesis. As videos are essentially collections of images, GAN-based video synthesis has become feasible. However, the fact that a video has multiple images as opposed to just one, presents the main challenge in video synthesis. Video synthesis is a multi-modal data generation challenge that includes elements of motion, speed, sound, and picture. The temporal aspect and interdependence between frames further increase the complexity of video generation. The connectedness between frames is crucial for producing high-quality videos. Merely ensuring that each individual frame looks realistic is insufficient; the temporal dependency between consecutive frames must be preserved for effective video synthesis. Despite the increased complexity of video-GANs compared to image-GANs, they have been utilized in various research studies involving video datasets [23]. In recent years, several generative adversarial networks (GANs) have been developed specifically for video generation. The primary objective behind these advancements is to generate high-resolution, indistinguishably natural videos with extended temporal range. However, while there has been significant research on producing conditional output for various inputs in the image domain, conditional video generation, especially from diverse inputs like audio signals, textual data, semantic maps, images, or videos, has received less attention and requires further investigation [24]. One of the most complex tasks in conditional video generation is generating videos based on textual descriptions, where semantic matching between the provided text and the synthesized video, visual frame quality, and frame coherence all need careful consideration. It is an extremely challenging yet crucial area of research with numerous potential applications, including multimedia special effects, synthetic data generation for reinforcement learning systems and domain adaptation, among others. In this study, we focus on the less explored domain of generating videos from text and propose Video-Text Matcher-Generative adversarial GAN (VTM-GAN). Our approach aims to tackle the challenges posed by text-conditioned video synthesis and contributes to advancing this important research area.

2 Related work

GANs have proven to be highly effective in generating high-quality images that closely align with given text descriptions. Unlike traditional image generation methods that rely solely on noise, text-to-image (T2I) generation operates by conditioning on concatenated noise and text conditions as input. This approach allows text descriptions to play a guiding role in the generation process, resulting in visually appealing conditional image outputs. The use of text descriptions, as opposed to mere labels, allows for the incorporation of a wealth of semantic information about the depicted objects, their attributes, and spatial arrangements, enabling the portrayal of diverse and intricate scenarios with fine details.

Text-to-image GANs: Reed et al. [17]. were the first to take the initiative and extended the conditional GANs, employing them to produce convincing visuals that correspond to the input texts. They demonstrated that their model could produce convincing visuals of birds and flowers from textual descriptions. Reed et al. [25] developed the "Generative Adversarial What-Where Network" (GAWWN) that generates images taking into consideration both "what" and "where" aspects. Besides focusing on what is to be drawn, it also determines at which location the object is to be drawn. In order to improve the resolution of the produced samples and to retain the semantic relationship between the textual and visual data, Zhang et al. [18] introduced StackGAN, which consists of two stages, with stage 1 generating low-resolution images and stage 2 generating high resolution images. Xu et al. [19] introduced a layered GAN referred to as Attentional Generative Adversarial Network (AttnGAN) for successfully generating fine-grained images with better quality using both word level information as well as the sentence level information. As the name suggests, it is the attention driven approach which means that more attention is given to the words of importance in a sentence. The demand for generating images at various scales, ensuring semantic consistency, and achieving high resolution has driven the adoption of complex stacked generative adversarial networks with multiple generators and discriminators. However, the increasing complexity of these networks has led to slower and less efficient GAN training processes. To address these challenges, Tao et al. [26] proposed the Deep Fusion Generative Adversarial Network (DF-GAN). Unlike other approaches, DF-GAN utilizes a single

generator-discriminator model and introduces a novel regularization method called "Matching-Aware zero-centered Gradient Penalty" to generate images from text without the need for additional networks. This innovative architecture aims to streamline the training process while still achieving high-quality image synthesis from textual descriptions.

Video-GANs: While both conditional and unconditional image generation are well-studied problems, different studies have also leveraged GANs in video generation. Vondrick et al. [27] were the pioneers in utilizing Generative Adversarial Networks (GANs) for video generation. The authors introduced video-GAN (VGAN), which emphasizes the importance of capturing scene dynamics by splitting it into dynamic and static components. Saito et al. [28] introduced Temporal Generative Adversarial Nets (TGAN), an approach capable of generating unlabeled videos by learning their semantic descriptions. Unlike its predecessor, VGAN, TGAN uses the approach of decoupling temporal-level features and frame-level features, rather than assuming a division into background and foreground streams. This model is equipped with a generator for image generation and a generator for temporal coherence. Tulyakov et al. [29] introduced a video generation model called Motion and Content decomposed Generative Adversarial Network (MoCoGAN). This model is designed to generate videos from a noise vector, similar to TGAN, and it also utilizes a generator for image generation and a generator for ensuring temporal coherence. However, what sets MoCoGAN apart is its generator for temporal coherence, which is constructed using a Recurrent Neural Network (RNN). Building on MoCoGAN's foundation, Saito et al. [30] further expanded it and introduced Temporal GAN-version 2 (TGAN-V2). TGAN-V2 employs various sub-discriminators within the discriminator, as well as multiple sub-generators within the generator. Through extensive training, this model has demonstrated superior video quality.

Dual video discriminator GAN (DVD-GAN), introduced by Clark et al. [31], extends the capabilities of BigGAN for generating videos. It adopts an efficient spatial–temporal division in its discriminator. Unlike previous approaches, the generator of DVD-GAN does not depend on predefined assumptions for static, dynamic and temporal features. Ohnishi et al. [32] presented a hierarchical method for generating appearance-and-motion-realistic videos, consisting of distinct FlowGAN and TextureGAN. The initial GAN is responsible for generating optical flow, capturing edge and motion information whereas TextureGAN adds texture to the generated optical flow. While this model successfully produces motion-realistic videos, the output lacks the visual clarity and finer details seen in higher-resolution videos. To improve the understanding of scene dynamics, Nakahira et al. [33] introduced Depth Conditional Video GAN (DCV-GAN) which incorporates three-dimensional-geometrical information, emphasizing the importance of optical information along with 3D geometry and color details. This approach surpasses MoCoGAN in generating realistic and diverse samples. To address instability and non-convergence issues, especially in generating the samples of better resolution, Acharya et al. [34] introduced the idea of generating the videos in a progressive fashion. This technique uses a coarse-to-fine approach, starting with smaller networks and gradually adding new layers in generator network as well as in discriminator network for generating samples of improved-resolution. Munoz et al. [35] conceptualized Temporal Shift GAN which instead of using the three dimensional generator network introduces a novel two dimensional generator for generating videos. This approach maintains temporal coherence between the frames as well as the relationship between the different regions. For generating high-resolution videos with computational efficiency, Tian et al. [36] introduced MoCoGAN- High Definition video synthesis (MoCoGAN-HD). The challenge of generating videos is formulated as searching for the motion trajectory using a motion generator in the latent codes that have been generated by a pre-defined image generator. Thus, motion generator works on the latent codes and obtains representations which are finally used to generate temporally coherent frames. In a recent development, Hong et al. [37] introduced Arrow GAN, a novel approach that utilizes an arrow-of-time discriminator (Arrow-D) to impart a sense of time to the generated content. Arrow-D can autonomously discern the direction of time without the need for explicit supervision and serves as a guiding force for the generators to produce more realistic and temporally consistent results.

Text-to-video GANs: Like generating images based on conditions, conditional GANs have also been used in the video domain to produce videos depending on conditions. Compared to other conditional video generation, the research studies on text conditioning video generation is deficient. It is a new, timely problem and a more challenging task. Pan et al. [38]. were the first to aim at producing videos based on the text description and proposed Temporal GANs conditioned on Captions (TGANs-C). The generator network receives a latent noise vector and embedded textual description concatenation, which aids in creating a frame sequence. The discriminator in TGAN not only performs its primary task of distinguishing between real and generated data samples, but it also serves an additional role. Integrating a GAN and Variational Autoencoder (VAE) network, Li et al. [24] used a hybrid approach for generating videos. By combining the strengths of GANs and VAEs, the authors designed a model that could effectively extract features from videos, taking into account both the static visual characteristics (color and structure) and the dynamic aspects of the content. This approach allowed for more comprehensive and informative representations of the videos, leading

to improved video generation performance. Both [24] and [38] generated videos of constant length and were trained on datasets having lower resolution.

For strengthening the relationship between input text and the generated video sample, Balaji et al. [39] developed a Text-Filter conditioning Generative Adversarial Network (TFGAN), which incorporates a multi-scale scheme and text conditioning to generate video frames based on given textual descriptions. One of the primary concerns they tackled was generating videos with fixed lengths, which can be limiting in terms of capturing diverse and dynamic scenes. Deng et al. [40] introduced the "Introspective Recurrent Convolutional GAN (IRC-GAN)" approach, which incorporates a recurrent generator to produce high-quality frames. To ensure temporal coherence, this generator combines LSTM cells with 2D transconvolutional networks, enabling it to generate new frames based on the previous ones. The proposed model also introduces a mutual-information introspection discriminator, which leverages information from generated samples to specifically evaluate the semantic relation between video samples and their corresponding descriptions. Li et al. [41] introduced StoryGAN, a model rooted in the sequential conditional GAN framework. The primary goal of StoryGAN is to visualize stories by generating a series of images. For each sentence in the story, the model produces one corresponding image. This approach allows the progression of the narrative to be visually represented through a sequence of generated images. Yu et al. [42] also introduced a recurrent deconvolutional generative adversarial network (RD-GAN) for the conditional generation of videos. In this model, skip-thoughts are employed to represent text as latent vectors, which serve as input for the generator to produce videos frame by frame. RD-GAN effectively addresses the problem of visual discontinuity, a common challenge faced by many video generation models that results in unrealistic output. However, one limitation of RD-GAN is its stability issue when trained with too many frames. Additionally, the model's reliance solely on feature extraction restricts its ability to generate diverse videos. Moreover, it faces challenges in generating sharp and longer videos. In contrast, Kim et al. [43] proposed Text to Video Generation TiVGAN, a model that produces full-length videos. Rather than seeking a mapping feature between the text and all video frames as a whole, TiVGAN is trained with respect to a single frame and progressively evolves to generate a video clip of the desired length. Experimental results show that TiVGAN not only accurately generates videos based on the given descriptions but also produces results of higher sharpness and better quality, addressing some of the shortcomings of the RD-GAN model.

Attention-GANs: Nowadays, attention mechanism has become a crucial part of many applications' effective sequence modeling and transduction models. It is widely used in both the natural language processing and computer vision domains. Alami et al. [44] used attention mechanism in image-to-image translation and enhanced the image quality significantly. Their proposed algorithm takes advantage of the discriminator's capability to learn accurate attention maps without the need for any additional supervision. Chen et al. [45] suggested attention-GAN for the task of object transfiguration, which is a part of image-to-image translation. The attention network is responsible for predicting the regions of interest in the input image. It identifies specific areas that are relevant and important for the transformation process. On the other hand, the transformation network is responsible for actually transforming the object from one class to another. It takes the input image and focuses on the regions of interest as indicated by the attention network. Zhang et al. [46] introduced self-attention based mechanism into convolutional GAN and proposed Self-Attention Generative Adversarial Network (SAGAN). Attention GANs have been successfully introduced in other tasks including aerial scene classification [47], video game generation [48], Data Augmentation on Medical Images [49], high-quality long-time series samples [50], and text-to-image generation [19, 51]. Recently, Chen et al. [52] proposed Bottom-Up GAN (BoGAN) for generating videos from the text that utilizes an attention mechanism and introduces a region-level loss, which enables it to focus on specific regions within the video and produce fine-grained details as specified in the input text. This approach results in the successful synthesis of videos that closely align with the provided textual descriptions, enhancing the realism and quality of the generated videos. Jiang et al. [53] succeeded in synthesizing images with 256×256 resolution, utilizing pure transformer-based architectures and a GAN entirely free of convolutions. Lee et al. [54] presented ViTGAN, which uses Vision Transformers [55] in GANs, and proposed critical strategies for assuring training stability and enhancing convergence. STrans-GAN, which also employs Transformers in GAN, was proposed by Xu et al. [56], providing competitive results in both unconditional and conditional image production. Other transformer-based GANs employed for high-resolution image synthesis include HiT [57] and Swin transformers [58]. The application of transformers, however, remains yet to be investigated in tasks like generation of images from textual description or generation of videos from textual description. Recently, Naveen et al. [59] have used a transformer to enhance AttnGAN for generating images from text data. To our knowledge, the proposed Video-Text Matcher Generative Adversarial Network (VTM-GAN) for text-to-video generation, for the first time, leverages a transformer with the text to video GAN, enabling it to generate fine-grained high-quality videos following the text.

3 Proposed method

The proposed VTM-GAN architecture comprises of two main components, as illustrated in Fig. 1: the Video-Text Matcher (VTM) and the Text-to-Video (T2V)-GAN model.

3.1 Video-text matcher (VTM)

The Video-Text Matcher model is composed of two main components: a Transformer [60] and a ResNet-101 [61]. It is trained using the parameters from Contrastive

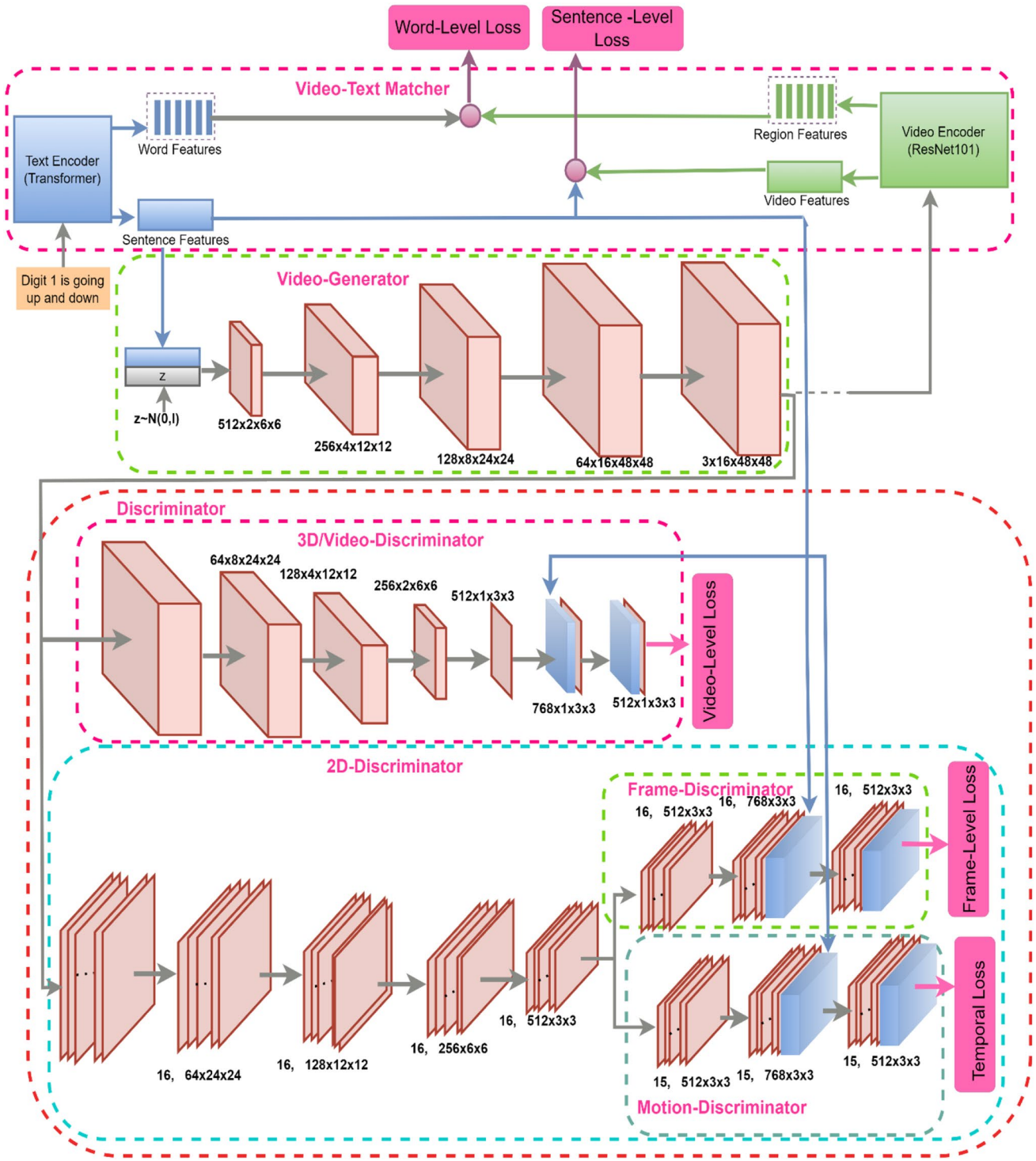

Fig. 1 The Proposed Model

Language-Image Pre- training (CLIP) [62]. CLIP is a training technique that combines the training of an image encoder with a text encoder by contrastive learning to achieve a compact embedding space between image and text pairs. Contrastive learning falls under the category of self-supervised learning, where augmentations are applied to the same input to create comparable representations. The contrastive objective in CLIP is inspired by previous studies on learning contrastive representations from images, which have shown its effectiveness compared to predictive objectives. CLIP predicts which pairings of the $n \times n$ potential (image, text) pairs actually happened across a batch with n pairs of (image, text). In order to learn a multi-modal embedding space, the image encoder is trained concurrently with the text encoder. This allows CLIP to maximize the cosine similarity between the text embeddings and the image of the n actual pairs in the batch while minimizing it for the $n^2 - n$ erroneous pairings. CLIP uses two different image encoding architectures, ResNet-50 and recently released Vision Transformer (ViT). The antialiased rect-2 blur pooling and the ResNet-D enhancements from [63] are used to make a number of adjustments to the original ResNet version. Additionally, it substitutes an attention pooling method for the global average pooling layer which is implemented as a single layer multi-head Query-Key-Value (*QKV)* attention with global average-pooled representation of the image as the query condition. Moreover, it also alters ViT by incorporating an extra layer normalization to the combined embedding of patch and position prior to the transformer. A Transformer that has been changed in accordance with the [64] specification serves as the text encoder. The actual size of CLIP has 63 M parameters, and is a 12 layered and 512-wide model containing 8 attention heads. The vocabulary size of the lower- case byte pair encoding (BPE) text representation employed by the transformer is 49,152 words. To enhance computing performance, the maximum sequence length is set to 76. The text sequence is delimited by start of sentence (SOS) and end of sentence (EOS) tokens, and the feature representation of the text is obtained from the activation of the transformer's top layer at the EOS token. This text feature representation undergoes layer normalization and is then linearly projected into the multi-modal embedding space.

The VTM architecture closely resembles that of CLIP in most aspects, but it diverges in a crucial way. While CLIP primarily focuses on capturing image and sentence-level features, VTM goes a step further by capturing region features from videos and word-level vectors. This distinction allows VTM to calculate a fine granularity consistency loss, unlike CLIP, since CLIP does not provide the word features and the image region features. To achieve this capability of calculating the region features and the word vectors, VTM introduces new neural network layers and trains a video-text matching task on our dataset. The video encoder and the text encoder in VTM are depicted in Figs. 2 and 3 respectively. The video encoder obtains output features from layer 3 to serve as initial region features. These are then fine-tuned using a 1×1 convolutional layer to facilitate the word-region level loss calculation. In the text encoder, an MLP layer and Layer Normalization is used for calculating the word vector based on the token vector produced by the Transformer. Utilizing the aforementioned techniques, VTM incorporates two distinct losses at different levels: word and region level,

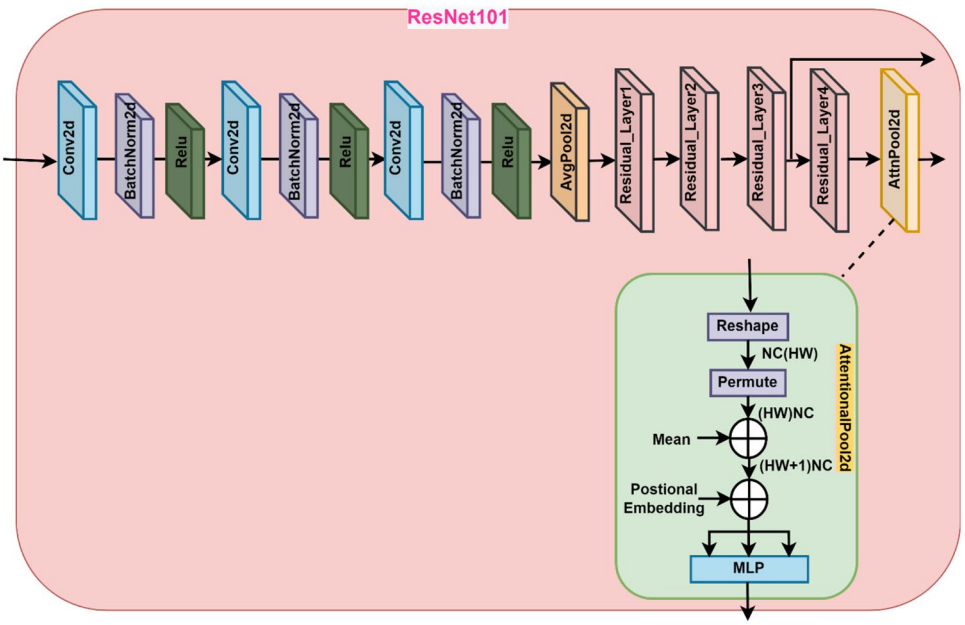

Fig. 2 Video Encoder used in VTM

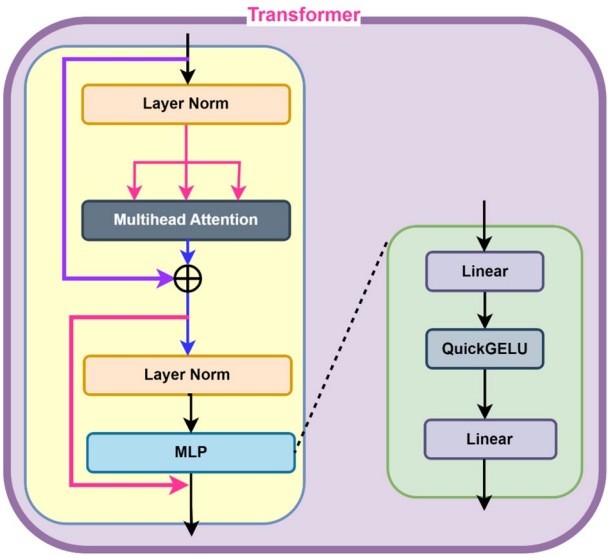

Fig. 3 Text Encoder used in VTM

as well as sentence and video level. The complete VTM block is presented in Fig. 4.

In the context of the provided description, the word vectors extracted from the text encoder are represented by the matrix $e \in R^{D \times T}$. The feature vector of i^{th} word is located in the i^{th} column denoted by e_i. This word feature matrix e has a dimension of D and total number of words is T. $\bar{e} \in R^{D}$ stands for the global sentence vector. On the other hand, the local video feature $\mathcal{F} \in R^{\bar{D} \times N}$ is taken from a ResNet-101 video encoder, where D represents the dimensions of local video feature matrix, and N representing total number of sub-regions comprising that video. The i^{th} column of $\mathcal{F}$ is a feature vector representing the i^{th} subregion. To bring the video data into a common semantic space with the text features, a perceptron layer P is included, and it performs the translation as shown in Eq. (1).

$$S = P\mathcal{F} \;\; \bar{S} = P\bar{\mathcal{F}} \tag{1}$$

where $S \in R^{\bar{D} \times N}$. S_i is the i^{th} column that represents the feature vector of i^{th} sub-region in the video in common space. $\bar{S} \in R^{\bar{D}}$ is the global feature vector of the entire video. $P \in R^{\bar{D} \times D}$ where D represents the dimension of common feature space of text embeddings and video features. To assess the similarity between each pair of words and video sub-regions, a similarity matrix is initially calculated. This matrix represents the similarity score for each potential pair of words in a sentence and video sub-regions. The calculation of this similarity matrix is performed using Eq. (2), which is mentioned as follows:

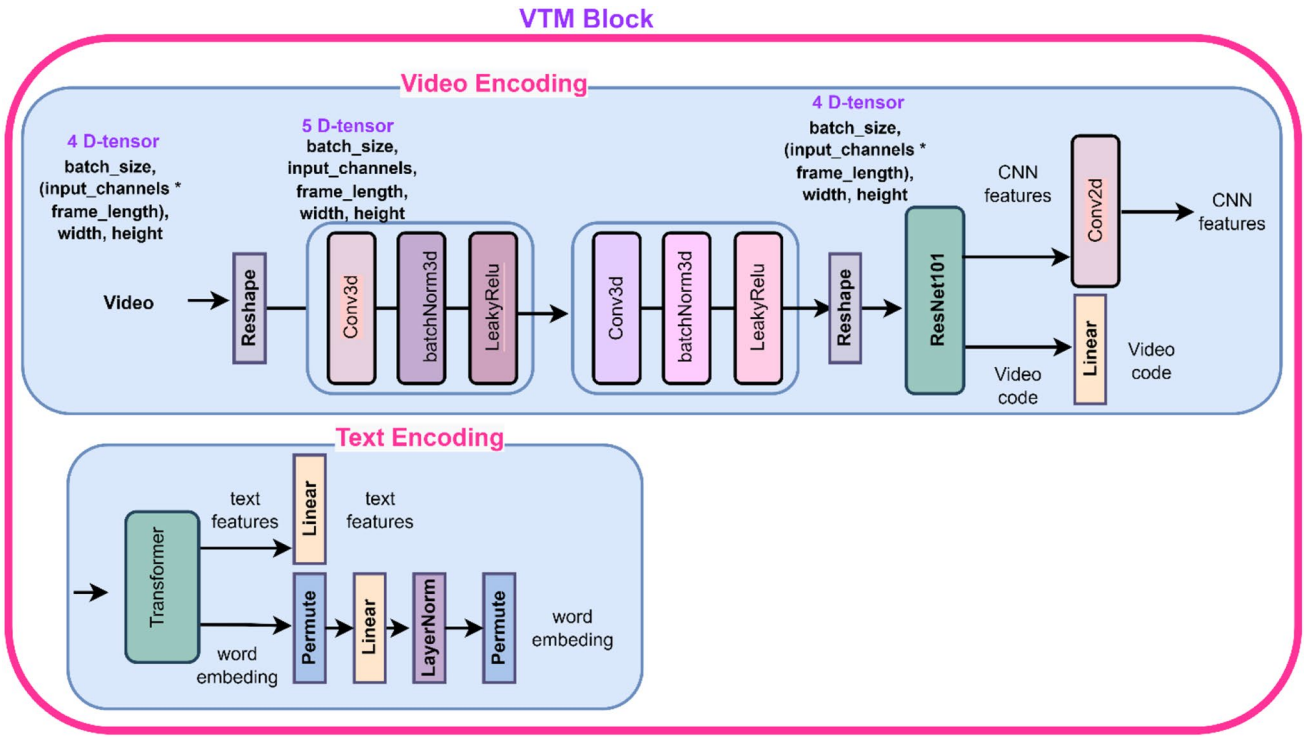

Fig. 4 VTM Block

$$\nu = e^T \mathrm{S} \tag{2}$$

where $\nu \in R^{T \times N}$ with ν_i , representing degree of dot-product similarity between the i^{th} word of the sentence and the j^{th} sub-region in the video. To make the similarity matrix more effective, it is normalized using Eq. (3):

$$\overline{\nu}_{i,j} = \frac{\exp\left(\nu_{i,j}\right)}{\sum_{k=0}^{T-1} \exp\left(\nu_{k,j}\right)} \tag{3}$$

Then, for each word (query), a region-context vector r_i is generated using an attention model. This vector r_i is a representation of the sub-regions of the video relative to the i^{th} word in sentence. The region-context vector is computed by calculating the weighted sum of all regional visual vectors, as shown in Eq. (4):

$$\mathrm{r}_i = \sum_{j=0}^{N} \mu_j, \mathrm{S}_j \quad \text{where} \quad \mu_j = \frac{\exp\left(\beta_1 \overline{\nu}_{i,j}\right)}{\sum_{k=1}^{N} \exp\left(\beta_1 \overline{\nu}_{i,j}\right)} \tag{4}$$

where factor β_1 determines how much attention is required for features of its relevant sub-regions when calculating region context vector for a word. It acts as a weighting factor that controls the importance of the visual information from the video's sub-regions in the region-context vector computation for each word in the sentence. After obtaining the region-context vector r_i for the i^{th} word in sentence, the correlation between the word and its corresponding video is established using the cosine similarity between r_i and e_i, as shown in Eq. (5):

$$X\left(\mathrm{r}_i, \mathrm{e}_i\right) = \frac{{\mathrm{r}_i}^T \mathrm{e}_i}{\|\mathrm{r}_i\| \|\mathrm{e}_i\|} \tag{5}$$

Taking inspiration from minimum classification error formulation in speech recognition, the relational score between whole video and the complete text driven by attention mechanism is given by Eq. (6):

$$X(V, E) = \log\left(\sum_{i=1}^{T-1} \exp\left(\beta_2 X\left(\mathrm{r}_i, \mathrm{e}_i\right)\right)\right)^{\frac{1}{2}} \tag{6}$$

where β_2 regulates the level of emphasis placed on the importance of the word-region pair that is most relevant. As β_2 approaches infinity ($\beta_2 \rightarrow \infty$), the function $X(V, E)$ converges towards finding the maximum value i.e. it approaches to $max_{i=1}^{T-1} X\left(\mathrm{r}_i, \mathrm{e}_i\right)$. The posterior probability of sentence E_i being a match with video V_i for a batch of pairs of video and sentence $\{(V, E)\}_{i=1}^{M}$ is determined by Eq. (7) as follows:

$$M\left(\mathrm{E}_i|\mathrm{V}_i\right) = \frac{\exp\left(\beta_3 X\left(\mathrm{V}_i, \mathrm{E}_i\right)\right)}{\sum_{j=1}^{M} \exp\left(\beta_3 X\left(\mathrm{V}_i, \mathrm{E}_j\right)\right)} \tag{7}$$

where β_3 represents a smoothing factor that is determined through experimental measurements. In this batch of sentences, only sentence E_i corresponds to video V_i; the other $\mathrm{M} - 1$ sentences are regarded as descriptions that don't match. To formulate the loss function, the negative logarithm of the posterior probability that the videos and their associated text descriptions are matched, is taken. The loss function is expressed as Eq. (8):

$$L_1^w = -\sum_{i=1}^{M} \log M\left(\mathrm{E}_i|\mathrm{V}_i\right) \tag{8}$$

where 'w' denotes "word". On a symmetrical basis, another loss function is derived that mirrors structure of Eq. (8). This new loss function is formulated to address a complementary aspect of the problem, thereby ensuring a balanced and comprehensive approach to evaluating the match between videos and their corresponding text descriptions. This new loss function is defined in Eq. (9):

$$L_2^w = -\sum_{i=1}^{M} \log M\left(\mathrm{V}_i|\mathrm{E}_i\right) \tag{9}$$

where

$$M\left(\mathrm{V}_i|\mathrm{E}_i\right) = \frac{\exp\left(\beta_3 X\left(\mathrm{V}_i, \mathrm{E}_i\right)\right)}{\sum_{j=1}^{M} \exp\left(\beta_3 X\left(\mathrm{V}_i, \mathrm{E}_j\right)\right)} \tag{10}$$

is the posterior probability that sentence E_i matches with its corresponding video V_i. Modifying Eq. (6) by Eq. (11):

$$X(V, \mathrm{E}) = \frac{\overline{S}^T \overline{e}}{\|\overline{S}\| \|\overline{e}\|} \tag{11}$$

and by substituting it in Eqs. (7), (8) and (9), two loss functions $L_1^{\int}$ and $L_2^{\int}$ are formulated with global sentence-vector $\overline{e}$ and global video-vector $\overline{S}$. Here $\int$ denotes sentence. The VTM loss is finally described by Eqs. (12) and (13):

$$L_{VTM} = \lambda_1 L_{VTM}^w + \lambda_2 L_{VTM}^{\int} \tag{12}$$

$$L_{VTM} = \lambda_1 \left(L_1^w + L_2^w\right) + \lambda_2 \left(L_1^{\int} + L_2^{\int}\right) \tag{13}$$

3.2 Video-generator

Given as the input, a sentence $\int$ concatenated with a noise vector z which is sampled from a normal distribution ($z \in R^{100}$), a generator network G is developed for generating sequence of frames:$\left\{R^{d_f,d_z}\right\} \rightarrow R^{d_c \times d_l \times d_h \times d_w}$ where d_c, d_l, d_h and d_w represent respectively the number of channels, length of the sequence, height of frames and width of frames. In order to capture both the spatial and the temporal information of the videos, 3D convolution filters are used with deconvolutions for concurrently synthesizing spatial information using 2D convolution filters and providing temporal coherence across adjacent frames. Initially, a fully-connected layer is used for learning a unified embedding m by concatenating text embedding $\overline{e}$ and noise variable. This unified embedding undergoes feature transformation, represented by Eq. (14):

$$m = W_{\overline{e}}\left[z.\overline{e}\right] \in R^{d_m+d_z} \tag{14}$$

where $W_{\overline{e}} \in R^{d_{\overline{e}}+d_m}$ and is the transformation matrix. Then generator G takes this latent variable as the input and generates the associated video as shown in Eq. (15).

$$Q = G(m) \in R^{d_c \times d_l \times d_h \times d_w} \tag{15}$$

Here, $Q = \left\{F_1, F_2 \ldots ..F_{d_l}\right\}$ is the generated video and F_i is the i^{th} frame of the video being generated where $F_i \in R^{d_c \times d_h \times d_w}$.

3.3 Discriminator

To ensure the generation of realistic videos, while maintaining temporal coherence across adjacent frames, a two-discriminator setup is employed: a 3D discriminator and a 2D discriminator.

3.3.1 3D discriminator $\boldsymbol{D_1(v, \overline{e})}$

The discriminator D_1 operates by taking two inputs: the video tensor v and the text embedding$\overline{e}$. Initially, it processes the video input through 3D convolutional layers, resulting in a video-level tensor$\mathcal{T}_v \in R^{d_c \times d_l \times d_h \times d_w}$. This tensor represents high-level features extracted from the video at different spatio-temporal locations. This follows with the augmentation of the video-level tensor with the text embedding $\overline{e}$, effectively incorporating the semantic information from the given description into the video representation. The augmented tensor is then passed through a dense layer with a softmax activation function for discriminating if the input video is sampled from the training data (real video) or generated data (synthetic video) and also if it is semantically consistent with the given description. This process is illustrated in Eq. (16):

$$D_1(v, \overline{e}) \rightarrow [0, 1] \tag{16}$$

Unlike the conventional discriminator that distinguishes between real and generated videos, here, an additional requirement is to maintain the semantic relationship between the generated video and its corresponding caption. Thus, a conditional discriminator is necessary which not only judges the authenticity of videos but also evaluates whether the video aligns with the given text description. Initially, during training, the discriminator disregards the conditioning information and promptly rejects samples generated by the model G as they may not appear realistic. However, as the generator G becomes proficient in generating plausible data, it must also learn to align them with the provided conditioning data. Similarly, the discriminator D must develop the ability to determine whether the samples from G satisfy the conditioning constraint. So for training such a discriminator, three types of inputs are provided which include: real video and semantically matched text $\left(\widetilde{Q}\right)$, synthetic video and semantically matched text (Q), and real video and semantically mismatched text $\left(\overline{Q}\right)$. The discriminator must score Q and $\overline{Q}$ as fake and $\widetilde{Q}$ as real. By introducing the third input that is real video and semantically mismatched text $\overline{Q}$, the discriminator learns to improve video-text matching in addition to generation of realistic videos, and thereby provides an additional signal to the generator. Consequently, the loss function designed to optimize D_1 is expressed as Eq. (17).

$$L_{D_1} = -\frac{1}{3}\left[\log\left(D_1\left(\widetilde{Q}, \overline{e}\right)\right) + \log\left(1 - D_1\left(\overline{Q}, \overline{e}\right)\right) + \log\left(1 - D_1(Q, \overline{e})\right)\right] \tag{17}$$

3.3.2 2D discriminator $\boldsymbol{D_1(v, \overline{e})}$

To further improve frame realism and semantic alignment with the given text and to maintain the temporal coherence, a 2D discriminator is used. So, this discriminator is actually a combination of two sub networks: Frame-discriminator (D_{frame}) and motion-discriminator (D_{motion}). Frame discriminator is responsible for determining the authenticity of individual frames in the video and assessing whether they exhibit semantic consistency with the given textual description. On the other hand, the motion-discriminator focuses on examining the temporal coherence within the video. It assesses the smoothness and natural flow of motion between adjacent frames.

Frame-Discriminator: $D_{\text{frame}}(F_i, \overline{e})$: Firstly, a common 2D convolutional model is used to extract the frame level tensor $\mathcal{T}_F \in R^{d_c \times d_h \times d_w}$ from every frame of the video. Then, augmentation of this frame-level tensor is done with the text embedding $\overline{e}$ and fed to the D_{frame} which discriminates

whether different frames of the video are both real and have semantic consistency with the caption. This evaluation is carried out as under by Eq. (18).

$$D_{\text{frame}}(F_i, \overline{e}) \rightarrow [0, 1] \tag{18}$$

The frame-level loss for optimizing D_{frame} is designed as shown in Eq. (19).

$$L_{D_{\text{frame}}} = -\frac{1}{3d_l}\left[\sum_{i=1}^{d_l}\log\left(D_{\text{frame}}\left(\widetilde{F}_i, \overline{e}\right)\right) + \sum_{i=1}^{d_l}\log\left(1 - D_{\text{frame}}\left(\overline{F}_i, \overline{e}\right)\right) + \sum_{i=1}^{d_l}\log\left(1 - D_{\text{frame}}\left(F_i, \overline{e}\right)\right)\right] \tag{19}$$

where $\widetilde{F}_i$, $\overline{F}_i$, and F_i are the i^{th} frames in $\widetilde{Q}$, $\overline{Q}$ and Q respectively.

Motion-Discriminator: $D_{\text{frame}}(\overrightarrow{\mathcal{T}_{F_i}}, \overline{e})$: In order to make the adjacent frames temporally coherent, similarity between two successive frames is determined using the Euclidean distances between their frame-level tensors. In other words, the magnitude of motion tensor is calculated as depicted in Eq. (20):

$$Dist\left(F_i, F_{i-1}\right) = \|\mathcal{T}_{F_i}, \mathcal{T}_{F_{i-1}}\|_2^2 = \|\vec{\mathcal{T}}_{F_i}\|_2^2 = \Delta\mathcal{T}_{F_i} \tag{20}$$

where $\Delta\mathcal{T}_{F_i}$ is the difference of frame-tensors of consecutive frames F_i and F_{i-1} indicating magnitude of motion between them and $\|\|_2$ is the L2-norm.The temporal coherence adversarial loss that optimizes motion discriminator is given by Eq. (21), as under:

$$L_{D_{\text{motion}}} = p - \frac{1}{3(d_l - 1)}\left[\sum_{i=2}^{d_l}\log\left(D_{\text{motion}}\left(\Delta\mathcal{T}_{\widetilde{F}_i}, \overline{e}\right)\right) + \sum_{i=2}^{d_l}\log\left(1 - D_{\text{motion}}\left(\Delta\mathcal{T}_{\overline{F}_i}, \overline{e}\right)\right) + \sum_{i=2}^{d_l}\log\left(1 - D_{\text{motion}}\left(\Delta\mathcal{T}_{F_i}, \overline{e}\right)\right)\right]. \tag{21}$$

Here, $\Delta\mathcal{T}_{\widetilde{F}_i}$, $\Delta\mathcal{T}_{\overline{F}_i}$ and $\Delta\mathcal{T}_{F_i}$ respectively denote the motion features between i^{th} frame and $(i-1)^{th}$ frame in $\widetilde{Q}$, $\overline{Q}$ and Q respectively.

3.4 Optimization

The overall optimization of discriminator can be done by minimizing the integrated losses at both the video-level, and frame-level, while also considering the loss for temporal coherence. This is represented by Eq. (22) below:

$$L_{Discriminator} = L_{D_1} + L_{D_2} = L_{D_1} + L_{D_{\text{frame}}} + L_{D_{\text{motion}}} \tag{22}$$

By minimizing Eq. (22), discriminator D learns to categorize videos as well as their frames as true or counterfeit while also aligning them with semantically appropriate descriptions. Additionally, it also trains the discriminator to recognize the temporal changes between frames. For the generator network, the adversarial losses for optimizing G at video level and at frame level are defined by Eqs. (23) and (24) as under:

$$L_{G_{\text{video}}} = -\frac{1}{3}\log\left(1 - D_1\left(Q, \overline{e}\right)\right) \tag{23}$$

$$L_{G_{\text{frame}}} = -\frac{1}{3d_l}\left[\frac{1}{d_l}\sum_{i=1}^{d_l}\log\left(1 - D_{\text{frame}}\left(F_i, \overline{e}\right)\right) + \frac{1}{d_l - 1}\sum_{i=2}^{d_l}\log\left(1 - D_{\text{motion}}\left(\Delta\mathcal{T}_{F_i}, \overline{e}\right)\right)\right] \tag{24}$$

The losses $L_{G_{\text{video}}}$ and $L_{G_{\text{frame}}}$ train generator to produce realistic and semantically aligned videos at frame-level as well as at video-level. Moreover, $L_{D_{\text{frame}}}$ loss also enforces the temporal coherence across the frames, thereby enhancing realism of the generated videos. In addition to these two losses, the VTM-loss L_{VTM}, as elaborated in Sect. 3.1 is introduced to further enforce the restrictions for maintaining fine granularity consistency of a video with its description. This VTM model provides two level losses that is loss at sentence level and loss at word level. In order to produce convincing videos with realism, overall final objective function of generator G is given by Eq. (25) as under:

$$L_{Generator} = L_{G_{\text{video}}} + L_{G_{\text{frame}}} + L_{VTM} \tag{25}$$

The entire methodology of training the VTM-GAN is presented in Algorithm 1.

Algorithm 1: Training VTM-GAN

Step 1 Training VTM-Module
  For t=1 to T, do
    Get the word level features e and the sentence level features $\bar{e}$ from the text encoder
    Get the region level features S and the video level features $\bar{S}$ from the video encoder
    Compute word-region level loss and sentence-video level loss using Eq.(13)
    Train the VTM-module using Eq.(13)
Step 2 Training T2V-GAN-Module
  For k=1 to K, do
    Get the noise vector $z \sim \mathrm{N}(0,1)$
    Concatenate the noise vector z with the sentence embedding $\bar{e}$, yielding the input m
    Generate the video $Q = G(m)$, where G is the generator
    Compute the video-level loss using Eq.(17)
    Compute the frame-level loss using Eq.(19)
    Compute the temporal loss using Eq.(21)
    Update the discriminator by minimizing Eq.(22)
    Update the generator by minimizing Eq.(25)
  End for

4 Experiments and results

The proposed VTM-GAN, designed for text-to-video generation, is thoroughly evaluated by comparing it with numerous state-of-the-art techniques. This evaluation involves tackling the challenging task of generating videos using the SBMG dataset [65].

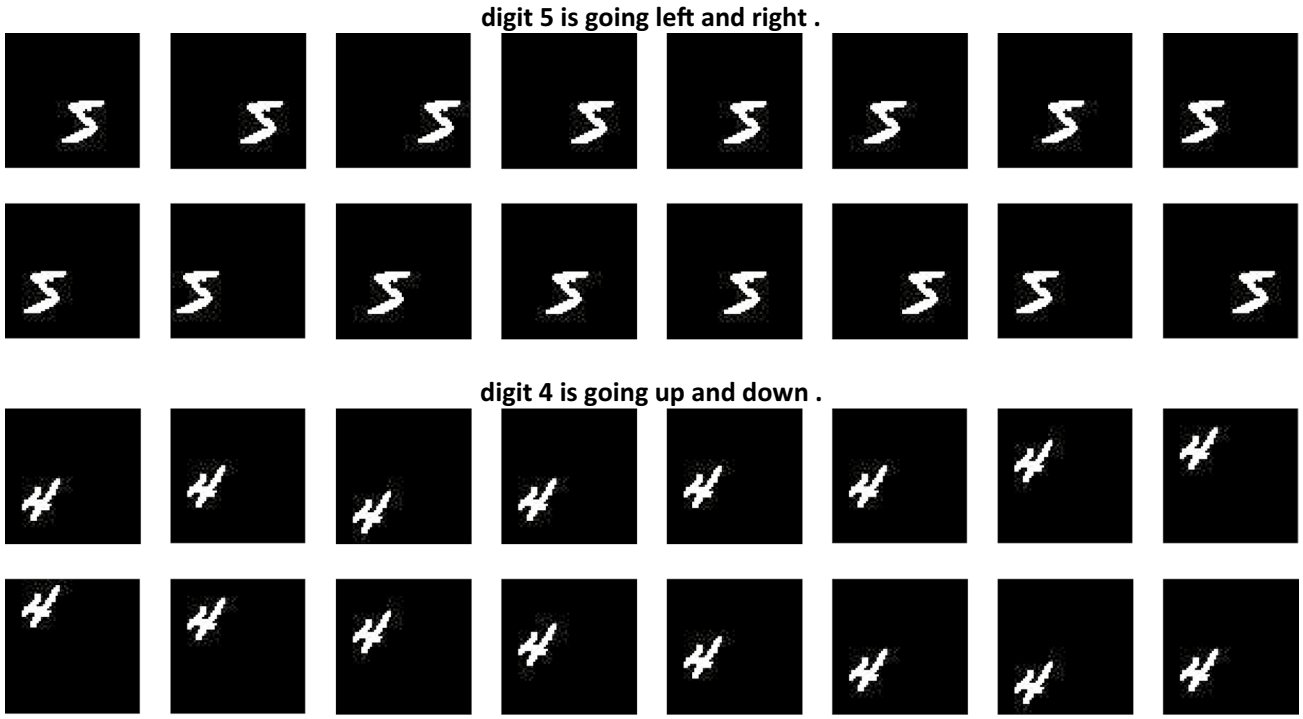

Fig. 5 Frame sequence along with corresponding captions from SBMG [65] dataset

4.1 Dataset used

Single Digit Bouncing MNIST GIFs (SBMG) [65]: This dataset was originally developed for the purpose of generating videos from textual descriptions and introduced in [65]. It is a synthetic dataset containing 12 GIFs, each depicting a handwritten digit moving within a 64×64 frame. These GIFs are 16 frames long and feature a single 28×28 digit that moves in either an upward/downward or leftward/rightward direction. Additionally, each GIF is accompanied by a single sentence that describes the digit's movement direction, as illustrated in Fig. 5.

4.2 Parameter setting

The dataset used for training and testing was resized to 48×48, and the training dataset comprised 4,78,961 images. For encoding the text, input and hidden layers in VTM encoder all had the dimension fixed to256. The dimension of the sentence embedding in the generator is also 256 which is concatenated with the noise vector of dimension 100. The no. of the frames in a video is same as in the original dataset, $d_l = 16$ and number of channels, $d_c = 1$. However, height along with width of frames is set as, $d_h = d_w = 48$. In the discriminator, the size of video tensor $\mathcal{T}_v$ was set to $512 \times 1 \times 3 \times 3$ for 3D discriminator and the size of frame-level tensor $\mathcal{T}_F$ was set to $512 \times 3 \times 3$ for 2D discriminator. The optimizer used was Adam optimizer having momentum terms set to 0.9 and 0.999. Other parameters used were: $\beta_1 = 4.0, \beta_2 = 5.0$, $\beta_3 = 10.0$, $\lambda_1 = 4.0$ and $\lambda_2 = 1.0$. The learning rate of the encoder was fixed as 0.002 whereas learning rates of both discriminator and generator were set to 0.0002. The VTM module was trained for 599 epochs, with batch size set to 96 whereas the final model was trained for 299 epochs with batch size set to 192. It took nearly 20 days for training VTM module and about 35 days for training the final model. The model was trained and tested on a system with the following configuration: RTX 3090 24 GB, Core i9 CPU with 128 GB RAM.

Table 1 Comparative Analysis of performance of proposed method with other text to video generation models

Model	FID-image	FID-video	IS
IRC-GAN	45.17	4.02	3.76
BoGAN	43.61	**3.36**	4.37
VTM-GAN	**42.46**	3.43	**4.52**

Better results obtained have been highlighted

4.3 Evaluation metrics used

To quantitatively evaluate the results and compare them with state-of-the-art methodologies, three metrics were used. These include Frechet Inception Distance at image level (FID-image), Frechet Inception Distance at video level (FID-video) and Inception Score (IS). FID-image is utilized to assess the quality of each individual frame. It quantifies the Frechet Distance between the features extracted from real frames and the synthesized frames. A trained Inception v3 network is used to extracts these features. Realistic frames yield significantly lower FID values, indicating better quality. FID-video, a variant of FID-image, goes beyond single frames to measure both visual quality and temporal consistency at the video level. Lower values of FID-video suggest improved overall results, reflecting higher visual fidelity and better temporal coherence in the generated videos. Inception Score (IS) serves as an automated metric to evaluate the quality of image and video generative models. A higher IS value indicates superior performance.

4.4 Quantitative analysis

The VTM-GAN proposed in this study demonstrates better proficiency while generating moving digits from the textual description. It exhibits more encouraging outcomes than state-of-the-art Text-to-Video GAN methods. The proposed technique has been contrasted with most recently introduced approaches for text-to-video generation which include IRC-GAN [40] and BoGAN [52]. To assess the quality of generated videos from captions, quantitative results of VTM-GAN evaluated on the Single Digit Bouncing MNIST GIFs (SBMG) dataset were compared with various state-of-the-art methods, as shown in Table 1. Notably, VTM-GAN outperforms both the compared techniques in terms of FID-image and IS metrics. VTM-GAN reduces FID-image from 45.17 to 42.46 as well as increases IS from 3.76 to 4.52 signifying the higher quality of generated frames. In terms of FID-video, though better results are shown by BoGAN, VTM-GAN still achieved comparable performance in this metric.

4.5 Qualitative analysis

In Fig. 6, the qualitative result analysis for the SBMG dataset showcases the performance of the proposed VTM-GAN in comparison to various existing GAN models. The results provide clear evidence that VTM-GAN exhibits superior performance over current state-of-the-art techniques. The outcomes demonstrate that VTM-GAN has the capability to generate entire videos that correspond to the given textual descriptions. These generated videos not only possess visual and semantic consistency but also maintain temporal

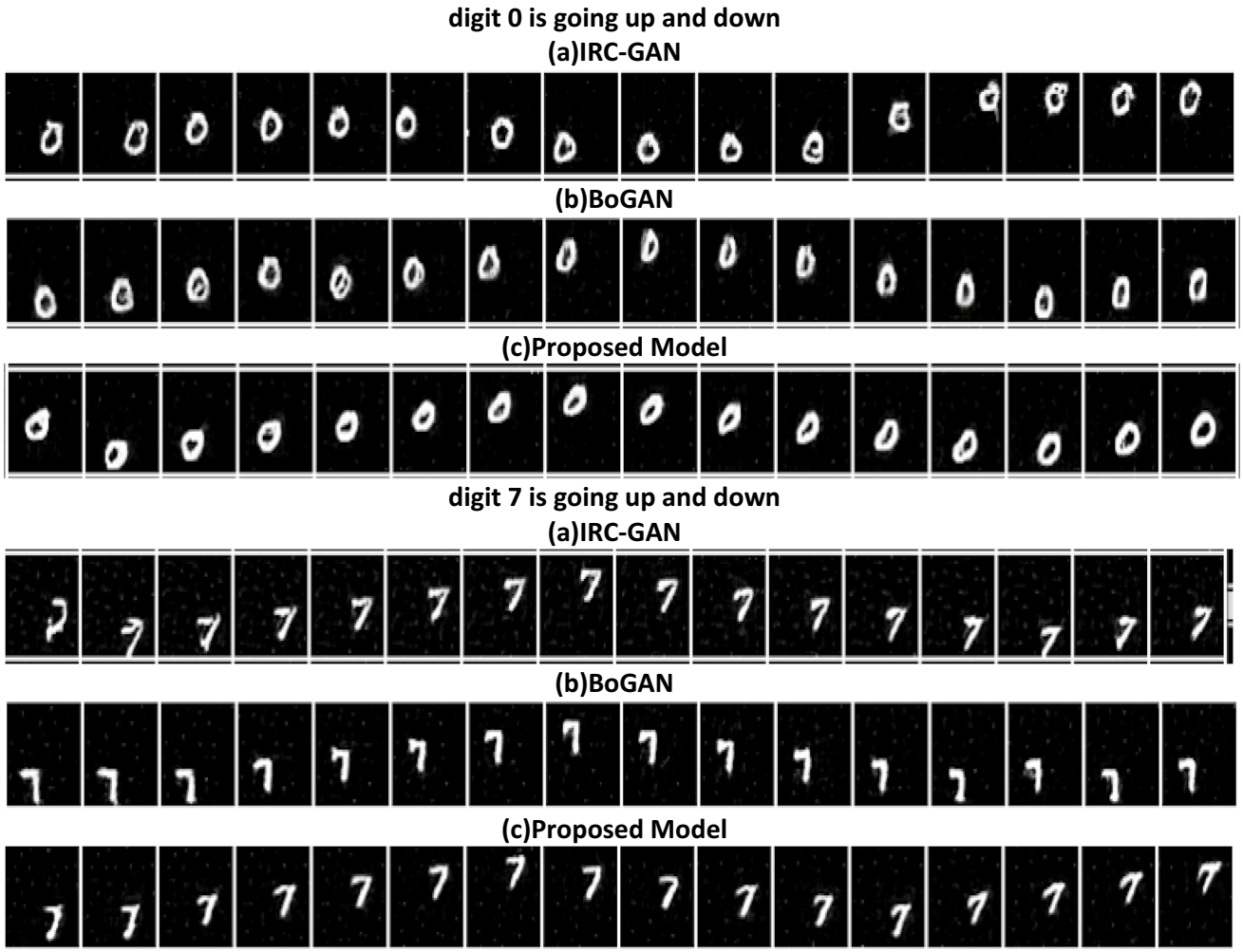

Fig. 6 Samples generated from the textual descriptions by **a** IRC-GAN, **b** BoGAN, **c** Proposed Model

Table 2 Ablation study on using different losses

	FID-image	FID-video	IS
Using only discriminator losses			
L_{D_1}	79.21	4.46	3.72
L_{D_2}	93.57	4.83	3.35
$L_{D_1} + L_{D_2}$	56.62	4.18	3.77
Introducing clip loss			
ClipLoss + L_{D_1}	62.10	4.27	4.38
ClipLoss + L_{D_2}	71.43	4.60	4.07
ClipLoss + L_{D_1}+L_{D_2}	43.86	4.01	4.43
Introducing VTM loss			
VTMLoss + L_{D_1}	59.29	4.11	4.35
VTMLoss + L_{D_2}	53.41	4.34	4.19
V = TMLoss + L_{D_1}+L_{D_2}	42.46	3.43	4.52

coherence throughout their duration. The VTM-based mechanism employed in the proposed model plays a pivotal role in this success. By considering similarity at both the sentence and video level, as well as at the word and region level, the model incorporates consistency losses. This ensures that the generated videos align well with the provided text, resulting in higher-quality and more coherent outputs.

4.6 Ablation analysis

An ablation study was conducted to investigate the impact of different losses applied in the model. The quantitative results obtained from using these different losses are presented in Table 2. From the results in Table 2, it is evident that employing video level loss yields superior outcomes in

terms of FID-image and IS when compared to using frame level loss alone. Furthermore, the notable improvements are achieved by combining both losses, as all three metrics used for analysis exhibit better performance. Additionally, the inclusion of clip loss alongside video level loss and frame level loss leads to reductions in FID-image and FID-video from 56.62 to 43.86 and 4.18 to 4.01, respectively. Moreover, the Inception Score (IS) improves from 3.77 to 4.43. However, when VTM-loss is employed in conjunction with frame level and video level losses, there are reductions in FID-image and FID-video from 56.62 to 42.46 and 4.18 to 3.43, respectively,whereas IS increases from 3.77 to 4.52, indicating that superior results are obtained using VTM loss alongside both frame level and video level losses, as utilized in the proposed method.

5 Conclusion and future work

This paper introduces a novel method called VTM-GAN for generating videos corresponding to textual descriptions. Its key component is the VTM module, which combines the training of video encoder and text encoder using contrastive learning to create a compact embedding space between paired video-text elements. VTM is an improved version of CLIP, addressing the limitation of CLIP in providing word features and video region features, which affects its ability to calculate fine granularity consistency loss. The architecture of VTM is similar to that of CLIP, but VTM stands out as it effectively captures the region features of the video and word-level vectors in combination with features of video and sentence-level vectors. This is achieved through the addition of new neural network layers dedicated to calculating region features and word vectors. On the text encoder side, an MLP layer and Layer Normalization are utilized to compute the word vector based on the token vector generated by the Transformer. The proposed VTM-GAN model shows significant improvement over earlier state-of-the-art GAN models. On the SBMG dataset, it achieves notable reductions in FID-image values from 45.17 to 42.46, FID-video values from 4.02 to 3.43, and an increase in IS from 3.76 to 4.52. Extensive experimental findings provide ample evidence of the efficiency of the suggested strategy for producing semantically aligned videos from textual descriptions, highlighting the superiority of VTM-GAN over previous GAN models. The future efforts will primarily concentrate on scaling up the model to handle higher-resolution videos, enabling the synthesis of videos with greater visual clarity and quality. Another key focus will be to enhance the model's capabilities in generating videos based on longer texts, enabling the model to understand and interpret more complex and diverse captions. Additionally, extending the existing framework to generate longer-duration videos, enabling the model to generate videos of extended lengths, providing more comprehensive and immersive visual storytelling experiences.

Acknowledgements This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Diqi M, Hiswati ME, Nur AS (2022) Stockgan: robust stock price prediction using gan algorithm. Int J Inf Technol 14(5):2309–2315
2. Iliyasu AS, Deng H (2022) N-GAN: a novel anomaly-based network intrusion detection with generative adversarial networks. Int J Inf Technol 14(7):3365–3375
3. Diqi M (2023) Twittergan: robust spam detection in twitter using novel generative adversarial networks. Int J Inf Technol 15:3103–3111
4. Abdelhalim ISA, Mohamed MF, Mahdy YB (2021) Data augmentation for skin lesion using self-attention based progressive generative adversarial network. Expert Syst Appl 165:113922
5. Ledig C, Theis L, Husź ar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4681–4690
6. Li W, Zhou K, Qi L, Lu L, Jiang N, Lu J, Jia J (2021) Best- buddy gans for highly detailed image super-resolution. arXiv preprint arXiv:2103.15295
7. Pattanaik A, Balabantaray RC (2023) Mish-dctgan based combined image super-resolution and deblurring approach for blurry license plates. Int J Inf Technol 15:2767–2775
8. Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196
9. Zhang Z, Pan X, Jiang S, Zhao P (2020) High-quality face image generation based on generative adversarial networks. J Visual Commun Image Represent 71:102719
10. Yu X, Porikli F (2016) Ultra-resolving face images by discriminative generative networks. European conference on computer vision. Springer, pp 318–333
11. Huang H, He R, Sun Z, Tan T (2019) Wavelet domain generative adversarial network for multi-scale face hallucination. Int J Comput Vision 127(6):763–784
12. Balayesu N, Kalluri HK (2020) An extensive survey on traditional and deep learning-based face sketch synthesis models. Int J Inf Technol 12(3):995–1004
13. Denton E, Gross S, Fergus R (2016) Semi-supervised learning with context-conditional generative adversarial networks. arXiv preprint arXiv:1611.06430

14. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2019) Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp 4471–4480
15. Goodfellow I (2016) Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160
16. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
17. Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis. International conference on machine learning. PMLR, pp 1060–1069
18. Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas DN (2017) Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp 5907–5915
19. Xu T, Zhang P, Huang Q, Zhang H, Gan Z, Huang X, He X (2018) Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1316–1324
20. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp 2223–2232
21. Yuxin D, Longfei W (2022) Multidomain image-to-image translation model based on hidden space sharing. Neural Comput Appl 34(1):283–298
22. Liu Z, Deng J, Li L, Cai S, Xu Q, Wang S, Huang Q (2020) Irgan: Image manipulation with linguistic instruction by increment reasoning. In: Proceedings of the 28th ACM International Conference on Multimedia. pp 322–330
23. Aldausari N, Sowmya A, Marcus N, Mohammadi G (2022) Video generative adversarial networks: a review. ACM Computing Surveys (CSUR) 55(2):1–25
24. Li Y, Min M, Shen D, Carlson D, Carin L (2018) Video generation from text. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 32. pp 7065–7072
25. Reed SE, Akata Z, Mohan S, Tenka S, Schiele B, Lee H (2016) Learning what and where to draw. In: NIPS. pp 1–7
26. Tao M, Tang H, Wu S, Sebe N, Jing XY, Wu F, Bao B (2020) Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. arXiv preprint arXiv:2008.05865
27. Vondrick C, Pirsiavash H, Torralba A (2016) Generating videos with scene dynamics. In Proceedings of the 30th conference on neural information processing systems. pp 613–621
28. Saito M, Matsumoto E, Saito S (2017) Temporal generative adversarial nets with singular value clipping. In: Proceedings of the IEEE international conference on computer vision. pp 2830–2839
29. Tulyakov S, Liu MY, Yang X, Kautz J (2018) Mocogan: Decomposing motion and content for video generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1526–1535
30. Saito M, Saito S (2018) Tganv2: Efficient training of large models for video generation with multiple subsampling layers. arXiv preprint arXiv:1811.09245 2(6)
31. Clark A, Donahue J, Simonyan K (2019) Efficient video generation on complex datasets. 2(3):4. arXiv preprint arXiv:1907.06571
32. Ohnishi K, Yamamoto S, Ushiku Y, Harada T (2018) Hierarchical video generation from orthogonal information: Optical flow and texture. In: Proceedings of the AAAI Conference on Artificial Intelligence vol. 32.
33. Nakahira Y, Kawamoto K (2019) Dcvgan: Depth conditional video generation. 2019 IEEE International Conference on Image Processing (ICIP). IEEE, pp 749–753
34. Acharya D, Huang Z, Paudel DP, Van Gool L (2018) Towards high resolution video generation with progressive growing of sliced Wasserstein gans. arXiv preprint arXiv:1810.02419
35. Munoz A, Zolfaghari M, Argus M, Brox T (2021) Temporal shift gan for large scale video generation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp 3179–3188
36. Tian Y, Ren J, Chai M, Olszewski K, Peng X, Metaxas DN, Tulyakov S (2021) A good image generator is what you need for high-resolution video synthesis. arXiv preprint arXiv:2104.15069
37. Hong K, Uh Y, Byun H (2021) Arrowgan: learning to generate videos by learning arrow of time. Neurocomputing 438:223–234
38. Pan Y, Qiu Z, Yao T, Li H, Mei T (2017) To create what you tell: Generating videos from captions. In: Proceedings of the 25th ACM international conference on Multimedia. pp 1789–1798
39. Balaji Y, Min MR, Bai B, Chellappa R, Graf HP (2019) Conditional gan with discriminative filter generation for text-to-video synthesis. IJCAI. 1:2
40. Deng K, Fei T, Huang X, Peng Y (2019) Irc-gan: Introspective recurrent convolutional gan for text-to-video generation. In: IJCAI. pp 2216–2222
41. Li Y, Gan Z, Shen Y, Liu J, Cheng Y, Wu Y, Carin L, Carlson D, Gao J (2019) Storygan: A sequential conditional gan for story visualization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 6329–6338
42. Yu H, Huang Y, Pi L, Wang L (2020) Recurrent deconvolutional generative adversarial networks with application to text guided video generation. arXiv preprint arXiv:2008.05856
43. Kim D, Joo D, Kim J (2020) Tivgan: text to image to video generation with step-by-step evolutionary generator. IEEE Access 8:153113–153122
44. Alami Mejjati Y, Richardt C, Tompkin J, Cosker D, Kim KI (2018) Unsupervised attention-guided image-to-image translation. In Advances in neural information processing systems. pp 3697–3707
45. Chen X, Xu C, Yang X, Tao D (2018) Attention-gan for object transfiguration in wild images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp 164–180
46. Zhang H, Goodfellow I, Metaxas D, Odena A (2019) Self-attention generative adversarial networks. International conference on machine learning. PMLR, pp 7354–7363
47. Yu Y, Li X, Liu F (2019) Attention gans: unsupervised deep feature learning for aerial scene classification. IEEE Trans Geosci Remote Sens 58(1):519–531
48. Torrado RR, Khalifa A, Green MC, Justesen N, Risi S, Togelius J (2020) Bootstrapping conditional gans for video game level generation. 2020 IEEE Conference on Games (CoG). IEEE, pp 41–48
49. Qi C, Chen J, Xu G, Xu Z, Lukasiewicz T, Liu Y (2020) Sag-gan: Semi-supervised attention-guided gans for data augmentation on medical images. arXiv preprint arXiv:2011.07534
50. Jeha P, Bohlke-Schneider M, Mercado P, Kapoor S, Nirwan RS, Flunkert V, Gasthaus J, Januschowski T (2021) Psa-gan: Progressive self-attention gans for synthetic time series. In: International Conference on Learning Representations.
51. Schulze H, Yaman D, Waibel A (2021) Cagan Text-to-image generation with combined attention generative adversarial networks. DAGM German Conference on Pattern Recognition. Springer, pp 392–404
52. Chen Q, Wu Q, Chen J, Wu Q, van den Hengel A, Tan M (2020) Scripted video generation with a bottom-up generative adversarial network. IEEE Trans Image Process 29:7454–7467

53. Jiang Y, Chang S, Wang Z (2021) Transgan: two pure transformers can make one strong gan, and that can scale up. Adv Neural Inf Process Syst 34:14745–14758
54. Lee K, Chang H, Jiang L, Zhang H, Tu Z, Liu C (2021) Vitgan: training gans with vision transformers. arXiv preprint arXiv:2107.04589
55. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
56. Xu R, Xu X, Chen K, Zhou B, Loy CC (2021) Stransgan: an empirical study on transformer in gans. arXiv preprint arXiv:2110.13107
57. Zhao L, Zhang Z, Chen T, Metaxas D, Zhang H (2021) Improved transformer for high-resolution gans. Adv Neural Inf Process Syst 34:18367–18380
58. Zhang B, Gu S, Zhang B, Bao J, Chen D, Wen F, Wang Y, Guo B (2021) Styleswin: transformer-based gan for high-resolution image generation. arXiv preprint arXiv:2112.10762
59. Naveen S, Kiran MSR, Indupriya M, Manikanta T, Sudeep P (2021) Transformer models for enhancing attngan based text to image generation. Image Vis Comput 115:104284
60. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In Advances in neural information processing systems. pp 5998–6008
61. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
62. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J et al (2021) Learning transferable visual models from natural language supervision. International Conference on Machine Learning. PMLR, pp 8748–8763
63. He T, Zhang Z, Zhang H, Zhang Z, Xie J, Li M (2019) Bag of tricks for image classification with convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 558–567
64. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I et al (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
65. Mittal G, Marwah T, Balasubramanian VN (2017) Sync-draw: automatic video generation using deep recurrent attentive architectures. In: Proceedings of the 25th ACM international conference on Multimedia. pp 1096–1104