Video Generative Adversarial Networks: A Review

NUHA ALDAUSARI, ARCOT SOWMYA, NADINE MARCUS, and
GELAREH MOHAMMADI, School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia

With the increasing interest in the content creation field in multiple sectors such as media, education, and entertainment, there is an increased trend in the papers that use AI algorithms to generate content such as images, videos, audio, and text. **Generative Adversarial Networks (GANs)** is one of the promising models that synthesizes data samples that are similar to real data samples. While the variations of GANs models in general have been covered to some extent in several survey papers, to the best of our knowledge, this is the first paper that reviews the state-of-the-art video GANs models. This paper first categorizes GANs review papers into general GANs review papers, image GANs review papers, and special field GANs review papers such as anomaly detection, medical imaging, or cybersecurity. The paper then summarizes the main improvements in GANs that are not necessarily applied in the video domain in the first run but have been adopted in multiple video GANs variations. Then, a comprehensive review of video GANs models are provided under two main divisions based on existence of a condition. The conditional models are then further classified according to the provided condition into audio, text, video, and image. The paper concludes with the main challenges and limitations of the current video GANs models.

CCS Concepts: • **Computing methodologies → Computer vision tasks**; • **General and reference → Surveys and overviews**;

Additional Key Words and Phrases: Generative Adversarial Networks, video synthesis, multimodal data, conditional generation

ACM Reference format:
Nuha Aldausari, Arcot Sowmya, Nadine Marcus, and Gelareh Mohammadi. 2022. Video Generative Adversarial Networks: A Review. *ACM Comput. Surv.* 55, 2, Article 30 (January 2022), 25 pages.
https://doi.org/10.1145/3487891

1 INTRODUCTION

The field of Computer Vision mainly deals with two types of data, namely images and videos, and this data can be used in many real-life applications for data generation, editing and classification. Data generation has gained significant attention since Ian Goodfellow released a model called **Generative Adversarial Networks (GANs)** in 2014 [1]. According to Google Scholar, there is an upward trend since the mid 2010's in publications when specifying "generative adversarial networks" as a search keyword, as demonstrated in Figure 1.

Nuha Aldausari is supported by a scholarship from Princess Nourah bint Abdulrahman University, KSA.
Authors' address: N. Aldausari, A. Sowmya, N. Marcus, and G. Mohammadi, School of Computer Science and Engineering, University of New South Wales, K17 Barker St., UNSW Sydney, NSW 2052, Australia.

0360-0300/2022/01-ART30
https://doi.org/10.1145/3487891

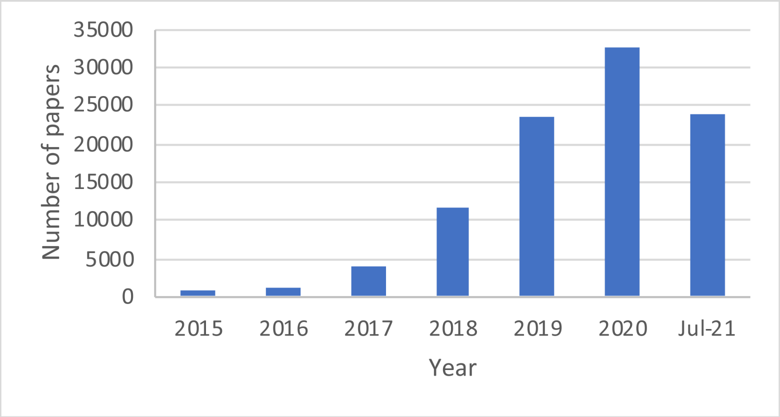

Fig. 1. Line chart represents the number of papers published in each year according to Google Scholar from 2015 to July 2021 (vertical axes).

The goal of generative models, in general, is to generate new data points that conform to the distribution of the training dataset. To accomplish this goal, GANs consists of two networks, one named the generator that gets a random noise vector as input and outputs images. The second network is the discriminator that differentiates between real training images and fake ones created by the generator. In other words, the discriminator D classifies real images x as D(x) = 1 and the fake ones as D(x) = 0. The networks are trained in an adversarial manner to reach the Nash equilibrium, which is an optimal state where D(x) = ½ for each image x, which means that the discriminator is not able to differentiate between real and fake samples [1].

One reason behind the success of GANs frameworks is that they have overcome some of the limitations of other generative models such as **Variational Autoencoders (VAEs)** [2]. For example, GANs frameworks produce sharper images compared to VAEs. The reconstruction loss function of VAEs is a pixel-wise similarity metric, while that of GANs is a semantic loss function [3, 4]. The issue with element-wise measures is that they do not align with the human visual system in the image domain; in other words, there are images that reflect a high element-wise error, however humans cannot distinguish between these images, and vice versa. The adversarial training in GANs facilitates building the reconstruction loss of the generator implicitly through the back-propagating gradients of the discriminator model that is responsible for differentiating between real images and fake ones.

GANs have been applied successfully on images and produce 1024*1024 images [5] that humans cannot differentiate from photographed images. Because a video is a sequence of images, it is also possible to employ GANs in the video domain. However, the main challenge in synthesizing videos is that a video contains multiple images rather than a single image. Besides, a video constitutes multimodal data with different aspects such as speed, motion, picture, and soundtrack. Moreover, the temporal dimension and the dependency between the frames adds to the challenges of generating videos. Although dealing with video in GANs can be more complicated than handling images, many research works have already applied GANs to video datasets, with the first attempt by Vondrick et al. in 2016 [6]. This was followed by others to leverage the use and value of applying GANs to a video dataset. Since then, several review papers have included a discussion of video GANs models [7–10]. As far as is known, this paper is the first attempt to review video GANs models more extensively.

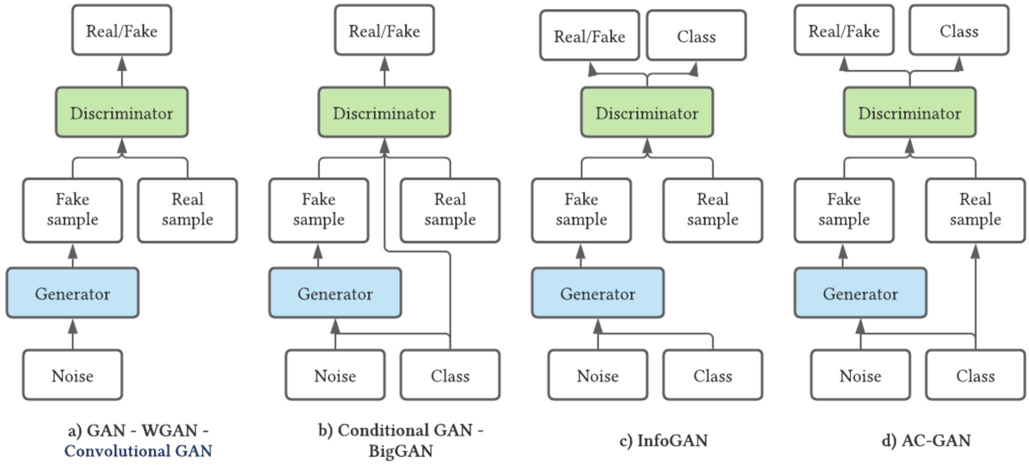

Fig. 2. Architecture of GANs variants. From left to right: (a) is that of GANs, WGAN, and convolutional GANs, which share the same architecture. (b) is conditional GAN, and BigGAN is based on it. (c) and (d) are of InfoGAN and AC-GAN, respectively.

This paper is thus structured as follows: first an overview of the main enhancements of GANs models and their variations are presented in Section 2. Other GANs review papers are then summarized in Section 3, and gaps in the literature identified. Next, video GANs models are categorized into unconditional and conditional models in Section 4 and each category reviewed. Finally, Section 5 concludes the paper.

2 RECENT ADVANCES In GANs

This section presents remarkable GANs variations that are employed in video GANs models that are detailed in Section 4. The original GANs model [1] is presented in Section 2.1. Three GANs frameworks with conditional settings are described in Section 2.2. In Section 2.3, an overview of another type of GANs named convolutional GANs [11] is provided. In Sections 2.4 and 2.5, other types of GANs called BigGAN and Wasserstein GAN, respectively, are described. A visual comparison of the overall architectures is illustrated in Figure 2.

2.1 Vanilla GANs

GANs framework was first introduced by Ian Goodfellow and his colleagues in 2014 [1]. Since then, GANs has attracted a lot of attention and played a significant role in the field of generative models. This is because GANs surpass other generative models with its unprecedented ability to generate new samples. GANs have multiple applications in different domains, including text, images, and sounds. Yann LeCun referred to GANs as “the coolest idea in machine learning in the last twenty years” [12].

The abbreviation GANs is based on three words: “Generative” means synthesizing new data based on training sets; “Adversarial” indicates that the two components of GANs, namely the generator and the discriminator, contest against each other, while the word “Networks” illustrates that the model consists of two networks. The networks could be fully connected neural networks, convolutional neural networks, recurrent neural networks, long short-term memory neural networks, autoencoders, or any combination thereof.

GANs consist of two networks competing in a minimax game, as illustrated in Figure 2(a). One network, called the generator, takes a random noise vector as input, and produces new instances by learning to follow real data distribution. On the other hand, the discriminator accepts training data and the data that is synthesized by the generator. The discriminator is a classification network that is supposed to classify real training data samples as “1”s and the generated data points as “0”s. The objective of the generator is to generate samples that cannot be distinguished from the real data samples by the discriminator. On the contrary, the discriminator targets the discrimination of real data samples from fake ones via classification. The two networks compete in a minimax game to improve each other’s performance. The ultimate goal for both networks is to reach the Nash equilibrium, which is a state where neither of the networks can improve by changing their parameters. In practice, however, it is difficult to find the Nash equilibrium [12].

The objective function for the generator (G) and the discriminator (D), as previously stated, is defined as a loss function (L) [1]:

$$min_G \ max_D \ L\left(D,\ G\right) \ = \ E_{x\sim p_r(x)} \ \left[log\, D\left(x\right)\right] \ + \ E_{z\sim p_z(z)} \ \left[log\left(1 \ - \ D\left(G\left(z\right)\right)\right)\right] \tag{1}$$

In Equation (1), E_{p} is the expectation with respect to a distribution p, $\mathrm{p_r}$ refers to the distribution of the real data, and $\mathrm{p_z}$ is the distribution of the input noise vector z. The discriminator is trained to recognize the real samples x, and produce high values close to one. Therefore $E_{x\sim p_r(x)}$ $[log\ D(x)]$ should be maximized. Meanwhile, the discriminator is also trained to recognize the fake samples $G(z)$ and produces low values close to zero, which means maximizing $E_{z\sim p_z(z)}$ $[log(1\ -\ D(G(z)))]$. In contrast, the generator needs to generate samples $G(z)$ that are similar to the real samples in order to fool the discriminator $E_{z\sim p_z(z)}$ $[log(1\ -\ D(G(z)))$.

2.2 Conditional GAN

In vanilla GANs [1], the model is unable to control the type of the generated samples. The generated data points could be from any category of the training data distribution. When sampling occurs, the generated samples might not represent all possible variations of the training data. In contrast, **conditional GAN (CGAN)** [13] adds a condition to both the generator and the discriminator, as shown in Figure 2(b). The condition might be a class, text or any other type of data, and the generated data is expected to match the condition. The loss function for CGAN is a modified version of the GANs loss function in Equation (1):

$$min_G \ max_D \ L(D,\ G) \ = \ E_{x\sim p_r(x)} \ [log\ D(x|c)] + E_{z\sim p_z(z)} \ [log(1 \ - \ D(G(zc))] \ [13] \tag{2}$$

where c is the condition added to the model.

Information Maximizing GAN, InfoGAN [14] for short, uses a slightly different approach to control the generation process. As illustrated in Figure 2(c), the input in InfoGAN [14] to the generator is a noise vector along with another variable. Unlike the condition in CGAN, the variable vector in InfoGAN is unknown. The purpose of using the variable is to control specific properties in the generated samples by maximizing the mutual information between this variable and the generated samples. Through training InfoGAN, the generator learns how to disentangle certain properties in the generated samples in an unsupervised manner through another model called the auxiliary model. The auxiliary model shares the same parameters as the discriminator. The aim of the auxiliary model is to predict the properties that are disentangled while the discriminator’s purpose is to distinguish real samples from fake ones. After training, the generated samples can be controlled by specifying some features that are learned such as colour, shape, rotation in image samples or class [14]. The loss function for InfoGAN is defined as follows:

$$min_G \ max_D \ L\left(D,\ G\right) \ - \ \lambda I \ \left(c;\ G \ \left(z,\ c\right)\right) \ [14] \tag{3}$$

The loss function is the same as CGAN, but with an additional term $\lambda I\ (c;\ G\ (z,\ c))$ to represent the mutual information loss between the variable c and the generated samples [14].

Auxiliary Classifier GAN (AC-GAN) [15] is another framework under conditional GANs-based architectures. AC-GAN frameworks share characteristic with InfoGAN and another with CGAN. In terms of the similarity with CGAN, a condition is fed into the generator, which can be a text, class or any other type of data. However, the condition is not an input for the discriminator. The common factor between AC-GANs and InfoGAN is that there is an auxiliary classifier that outputs the class of the input sample. The differences in the overall architecture between CGAN, InfoGAN, and AC-GAN are demonstrated in Figure 2(b), 2(c), and 2(d), respectively. The loss function of AC-GAN is divided into two terms: one for evaluating the predicted class and another one to discriminate fake from real samples.

The reviewed papers under conditional video generation (see Section 4.2) use one of the conditional GANs-based architectures. The reason behind choosing the conditional architecture is that conditions can enhance the network stability and provide higher quality samples [13].

2.3 Convolutional GANs

Vanilla GANs [1], considered as the simplest type of GANs, uses the **Multi-Layer Perceptron (MLP)** in both the generator and the discriminator. However, one of the main disadvantages of vanilla GANs is that the training process is not stable [12]. One of the possible solutions to this problem is to use a **convolutional neural network (CNN)** instead. In convolutional GANs [11], the generator uses a deconvolution structure, while the discriminator applies convolutional layers to classify generated images from the real images. While the type of networks in the generator and discriminator are different in vanilla GANs and convolutional GANs, the overall architecture in vanilla GANs and convolutional GANs are identical (see Figure 2(a)).

A significant number of recent GANs frameworks adopt CNN in their generators, discriminators, or in both. This is because the CNN in Convolutional GANs outperforms MLP in vanilla GANs in terms of the performance, quality of the resulted samples (usually images), and stability of the training process.

2.4 High Resolution GANs

The GANs frameworks discussed so far, when applied in the image domain, produce low resolution images as output, i.e 64 × 64 or 128 × 128 images. In contrast, BigGAN [16], ProGAN [5], StyleGAN v1 [17] and StyleGAN v2 [18] scale up the output resolution to 256 × 256 and 512 × 512 using different methods. BigGAN [16] produces high-resolution images by increasing the number of parameters and scaling up the batch size. The architecture of BigGAN is based on the **self-attention GAN (SAGAN**) [19]. The main advantage of SAGAN is that it focuses on different parts of the image by introducing an attention map that is applied to the feature maps in the deep convolutional model architecture. ProGAN [5] implements a different technique by progressively adding layers to the network to scale up the generated images. The training procedure starts with 4 × 4 images and shallow convolutional layers until convergence, then builds up more layers to generate 8 × 8 images and so on until reaching the desired resolution. StyleGAN v1 [17] adopts a progressive architecture for the generator. The visual features can be controlled by lower layers, i.e 4 × 4 while finer styles are managed by 1024 resolution layers. StyleGAN v1 controls the generated features by mapping a random noise vector to an intermediate space using fully connected layers of the mapping network. Then, the intermediate space is used to control the features in the generator by **Adaptive Instance Normalization (AdaIN)** layers. AdaIN is an instance normalization technique that adds the intermediate style space to the content of the images. StyleGAN v2 [18] overcomes the artefacts that are generated by styleGAN v1 by removing AdaIN that has proven to be the main issue.

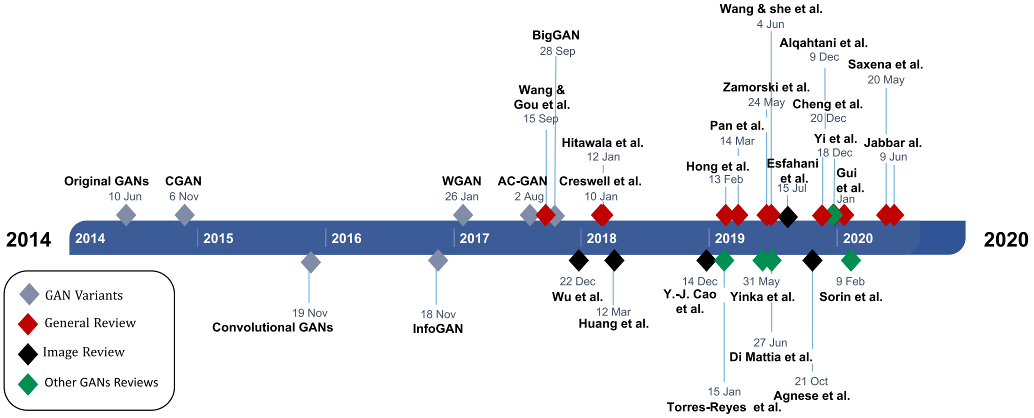

Fig. 3. Timeline of the review papers in Section 3, along with recent GANs advances in Section 2 (light blue).

2.5 Wasserstein GAN

Another variation of GANs that has been used in video generation is called **Wasserstein GAN (WGAN).** In vanilla GANs [1], the discriminator attempts to classify real data points from fake ones. However, WGAN's discriminator [20] is a "critic", whose responsibility is to assign a score that represents the distance between the distribution of the observed real data and the distribution of observed fake samples. WGAN uses Wasserstein distance instead of the **Jensen-Shannon (JS)** divergence and **Kullback-Leibler (KL)** divergence that are used in other generative models, and there is substantial improvement in the quality of the generated images and in the training stability.

3 RELATED WORK

As the study of GANs is accelerating rapidly, there are constantly new GANs frameworks that were not covered in the existing review papers. The timeline of survey papers found by searching Google Scholar for the keywords "overview of generative adversarial networks", "survey of generative adversarial networks", and "review of generative adversarial networks", as well as the papers cited in the retrieved papers, is illustrated in Figure 3. The timeline also includes the publication dates of the review papers, which will be discussed in this section, besides the publication dates of notable GANs frameworks discussed in Section 2. While some of the review papers provide overviews of the state-of-the-art GANs, others focused on GANs for a specific domain (e.g. image generation). Reviews of GANs for general visual image datasets exceed other specialized GANs reviews, including those in cybersecurity, anomaly detection and medical imaging (see below). As shown in Figure 3, the red diamonds that represent general reviews of GANs dominate other categories.

3.1 General GANs

The majority of the survey papers discuss GANs in general [7–10, 21–25]. One of the first attempts to review general GANs was in late 2017 [24]. Another general GANs survey was published in early 2018 that draws an analogy between GANs concepts and signal processing concepts to facilitate the understanding of GANs from a signal processing perspective [22]. Around the same time, Hitawala et al. [25] provided a review of the main improvements in the GANs frameworks. In 2019 and 2020, there was an increase in the number of papers that review GANs in general [7–10, 21, 23, 26–28]. It is worth mentioning that a more comprehensive and lengthy general review is available [19], and Cheng et al. [27] conducted comparative qualitative experiments on the mainstream GANs applied to the MNIST dataset [29], in which AC-GANs obtained the top classification accuracy. Saxena et al. [28] concentrate on models that address GANs disadvantages such as training instability and model collapse by modifying the architecture, the loss function or the optimization method.

3.2 Image GANs

Since the original GANs frameworks were initially built upon images, there is no doubt that the number of GANs applications in the image domain surpasses other areas such as text, voice, and video. Multiple reviews [30–34] focus on image synthesis, even though the major proportion of the general GANs reviews mentioned above, are also in the image domain. Huang et al. [31] categorize the image synthesis GANs frameworks into three types based on the overall architecture. These include direct architectures based on vanilla GANs, hierarchal models and iterative models consisting of multiple generators and discriminators. While each generator in a hierarchical architecture is tasked to deal with a different aspect of the disentangled representations of training images, the goal of a generator in the iterative models is to refine the quality of the generated images. Wu et al. [32] place image GANs models in two main categories, namely conditional image synthesis models and unconditional image synthesis models. In the unconditional models, different network modules handle texture, image super resolution, image inpainting, face synthesis, and human synthesis. A comparative study [33] of image GANs frameworks was conducted using two datasets: MNIST [29] and Fashion-MNIST [35]. The paper reviews different applications of GANs such as style transfer, image inpainting, super-resolution, and text to image. Similar applications are also reviewed elsewhere [30], with additional applications such as face ageing and 3D image synthesis. Moreover, Agnese et al. [34] direct attention to GANs models that are conditioned on text and produce images. Some of these models fall under semantic enhancement GANs, where the main goal is to ensure that text is semantically coherent with the generated image. Another category focusses on producing high resolution images conditioned on text. An additional category is for ensuring the diversity of synthesized images based on input text. The last type is text to video models that consider the temporal dimension of the training samples.

3.3 Other GANs

GANs have been applied successfully in other areas including the medical field, with medical GANs frameworks reviewed [36, 37]. Yi et al. [36] provide illustrations of juxtaposed GANs variation architectures. In addition, medical imaging GANs models are categorized based on the aim, such as quality improvement of the synthesized images, data augmentation, segmentation, classification, registration, and object detection. Another review [37] specializes in radiology imaging based on 33 papers. GANs models facilitate synthesis of new radiology images, improving the quality of existing ones, converting radiology images from one type to another and localizing a specific object.

Anomaly detection, which deals with finding data samples that deviate from the normal, has also taken advantage of GANs models. A survey paper [38] reviews existing GANs models that contribute to identifying anomalies, and re-implements the reviewed models to further verify the effectiveness of such models.

The field of synthesizing and enhancing audio using GANs architectures has also been reviewed [39]. In the audio generation models, audio can be generated from a noise vector or text, while in audio enhancement GANs models, the model input is a noisy audio signal while the output is a refined signal. Audio GANs models are usually trained on audio spectrogram representations or raw audio waveforms.

Yinka-Banjo et al. [40] review studies that utilize GANs frameworks in cybersecurity systems. Such models are used for protection or attack purposes. In the case of protection, GANs models synthesize new poisoning samples, and the cybersecurity system learns how to protect itself against such samples that might cause an attack. GANs systems can also be used to spoof other GANs cybersecurity systems by creating adversarial samples that are similar to the real authorized training data points.

Besides general GANs reviews, there are also domain-specific reviews that focus on image, audio and medical imaging as listed above. In addition, several review papers are focused on specific tasks such as anomaly detection and cybersecurity. However, no systematic review of video GANs has been performed, which have very specific characteristics due to their temporal dimension and the necessity to maintain temporal cohesiveness. The next section contributes to filling this gap and pays close attention to GANs models that generate videos and capture their temporal behavior.

4 VIDEO GANs

Compared to image GANs, video GANs require different treatment because of the data complexity. A video consists of multiple images with the additional time dimension. Therefore, a video GAN generator follows mostly one of four strategies. While most image GANs generators utilize 2D convolutional networks to synthesize images, video generators combine an RNN architecture with 2D convolutional networks to address the time-series nature of video data [41, 42]. Another approach is to use 3D convolutional networks [6] instead of 2D convolutional networks in image GANs to account for the time dimension. Coarse-to-fine strategy is another method that was first introduced in image GANs in an architecture called progressive growing GAN [5], then it was widely adopted in video GANs generators [43, 44]. The purpose of the progressive architecture in the video GANs realm is to generate the data first, and the generated data is fed into another generator to produce an enhanced result. The fourth method is the two-stream architecture video GANs generators [6, 45]. This method is not exclusive to video GANs generators as it is commonly used in image generation using GANs [46, 47]. The motivation for a two-stream architecture is to specialize each stream for a different aspect of the video [6, 45]. The discriminator in video GANs could be the same as the generator's strategy or use other strategies. It can use two-stream [42, 48–53] or a 3D convolutional network [54].

Video GANs models vary depending on the condition settings. While at one end, there are video GANs frameworks that are not supplied with conditional signals, as discussed in Section 4.1, other models are conditioned on audios, texts, semantic maps, images and videos, as discussed in Sections 4.2.1, 4.2.2, 4.2.3, and 4.2.4, respectively. The following sections review the video GANs models based on the presence or absence of input conditions and the variations within each category.

4.1 Unconditional Video Generation

This section is a review of unsupervised GANs frameworks in the video domain, and Table 1 summarizes unconditional video GANs models and their main goals. Moreover, a summary of these frameworks, the datasets and evaluation metrics used can be found in Table S4. Producing videos without prior information is more challenging as the model needs to capture the data distribution without help from the input signal which can help to narrow the target space. Although training unconditional video GANs can be difficult, some of the unconditional models have become the foundation for conditional frameworks. For instance, MoCoGAN [42] architecture, which is an unconditional model, is used in **Text-Filter conditioning Generative Adversarial Network (TFGAN)** [55] and storyGAN [41], both of which are conditional models.

Video-GAN (VGAN) [6] was the first attempt to generate videos using GANs. The generator consists of two convolutional networks: the first is a 3D spatio-temporal convolutional network that captures moving objects in the foreground, while the second is a 2D spatial convolutional model for the static background. The generated frames from the two-stream generator are combined, and then fed to the discriminator to distinguish real videos from the fake ones.

Table 1. The Reviewed Unconditional Video GANs Frameworks in Section 4.1

Publication	Conditional information	Task
VGAN [6]	no condition	Generating videos by generating background and moving objects separately.
FTGAN [56]	no condition	Generating videos from noise vectors by generating optical flow then the texture.
MoCoGAN [42]	no condition	Generating videos from noise vectors and controlling motion and content separately.
TGAN [54]	no condition	Generating videos from noise vectors by generating motion features then the texture.
TGANv2 [57]	no condition	Generating videos from noise vectors and focus on decreasing the computational costs.
DVD-GAN [48]	no condition	Generating videos from noise vectors on complex dataset (Kinetics-600).
G^3AN [58]	no condition	Generating videos from noise vectors and controlling motion and content separately.

The second column provides the type of condition while the third one gives information on the main task.

In VGAN [6], the foreground stream captures the foreground objects and their motions. However, the foreground layer in the generated result usually contains some flaws in temporal or spatial aspects. **Flow and Texture Generative Adversarial Networks (FTGAN)** [56] adds optical flow for representing the object motion more effectively. FTGAN follows a progressive architecture that starts with a GANs framework to capture the optical flow, followed by another GANs model to generate the texture that is conditioned on the result of the previous optical flow GANs, and produces the desired frames. Both texture and flow generators in FTGAN adopt VGAN structure by separating the foreground from the background, and setting the background to zero for the flow generator.

VGAN is based on disentangling the foreground from the background. Similarly, **Motion Content GAN (MoCoGAN)** [42], which is another type of unconditional video generator, separates the content from the movements to provide more control over these components. While VGAN [6] and FTGAN [56] map a video to a point in the latent vector, the MoCoGAN framework traverses N latent points, one per frame, where each vector can be decomposed into the motion vector and content vector. Therefore, it consists of an N-to-N RNN that accepts N random variables and produces N latent motion vectors. The motion vectors are combined with a fixed content vector for all N motion variables and fed to the generators to synthesize N images, each of which is a frame in the generated video. The generated images and videos are evaluated using two discriminators: one for the images, and the other for the generated video.

Similar to MoCoGAN [42], **Temporal Generative Adversarial Nets (TGAN)** [54] use N latent vectors for N frames. However, each frame is generated from a latent vector in TGAN while in MoCoGAN [42], a frame is generated from a combination of a motion vector and a fixed content vector shared across the frames. Another difference is that MoCoGAN utilizes RNN structure for the generators while in TGAN, there are N 2D image deconvolutional generators to produce N frames. The resultant frames along with the videos in the training set are fed into the 3D convolutional discriminator. TGAN employs WGAN and fulfills K-Lipschitz constraint by proposing a parameter clipping method called singular value clipping using WGAN to provide stable training. Temporal GAN v2 (TGANv2) [57] focusses on a training technique based on understanding the relationship between the resolution of the generated images and the computational cost. The main

Table 2. The Reviewed Speech to Video GANs Frameworks in Section 4.2.1

Publication	Conditional information	Task
Vougioukas et al. [49]	audio, initial image	Generating talking person videos from speech and image of the target.
DAVS [60]	(audio or video), initial image	Generating talking person videos from speech by disentangling visual and audio features.
Mittal et al. [50]	audio-based content representations, initial image	Generating talking person videos from disentangled audio representations.
Chen et al. [59]	audio, initial image	Generating only lip region from speech.
Jalalifar et al. [61]	audio, lip landmarks	Generating talking person videos from lip landmarks and speech.

The second column provides the type of condition while the third one gives information on the main task.

cause of the increase in computational cost is the end part of the generator. This is because the spatial resolution / feature map is increasing when moving forward in the generator net. Since TGANv2 is an unsupervised model, it is essential to supply the model with a larger batch size in order to generalize properly. A subsampling layer is introduced to reduce the batch size by stochastically sampling videos within a mini-batch, and then sampling a frame within each chosen video. Applying the subsampling technique multiple times in the generator network facilitates reduction in the size of the mini-batch.

Both MoCoGAN [42] and TGAN [54] synthesise N frames based on N motion vectors. However, Wang et al. [58] argue that having N motion vectors adds complexity to model training. Thus, the model discards the role of these vectors and produces frames with similar motions. G^3GAN overcomes this limitation by having one vector for the motion and another one for the content. G^3GAN consists of three streams; one each for the motion, the content, and for maintaining the coherence between spatio-temporal features. G^3GAN applies self-attention to enhance the quality in the generation results.

Dual video discriminator GAN (DVD-GAN) [48] expands BigGAN [16] capabilities in the video domain to produce 48 high quality images up to 256*256 based on complex datasets such as Kinetics human action dataset. DVD-GAN is trained on the entire dataset, Kinetics, and this is not the case in prior works [44, 55] that use only a subset and pre-processed samples. Similar to MoCoGAN [42], there are two discriminators to deal with the temporal and spatial aspects of a video.

4.2 Conditional Video Generation

There are several works that employ a conditional signal in GANs to direct the process and control modes of the generated data; The condition may be audio signal, text, semantic map, image, or video. The following subsections review conditional GANs based on the condition type.

4.2.1 Speech to Video Synthesis. This subsection discusses the GANs frameworks that are used to synchronize speech audio with facial movements, and Table 2 provides information on the main task of these frameworks. Also, Table S5 summarizes speech-to-video synthesis models.

Lips movement generation frameworks were the initial attempt at synchronizing a moving head with audio. Chen et al. [59] proposed a model that encodes the starting image and audio file. Then, the encoded features are combined and used as input to a decoder to generate videos. The synthesis videos are evaluated using a three-stream discriminator.

Table 3. The Reviewed Text to Video GANs Frameworks in Section 4.2.2

Publication	Conditional information	Task
TGANs-C [62]	one sentence	Generating video based on text using three discriminators: image, video, frame-caption matching.
Li et al. [44]	one sentence, first frame	Generating videos given encoded gist signal that includes initial frame and text description.
TFGAN [55]	one sentence	Generating videos based on text using multi-scale text-conditioning scheme in the discriminator.
StoryGAN [41]	multiple sentences	Generating animated stories based on a paragraph.

The second column provides the type of condition while the third one gives information on the main task.

While Chen et al. [59] consider only lips, other works [49, 50, 60, 61] study the synchronization between audio and the entire face. Jalalifar et al. [61] proposed a progressive framework that combined LSTM with CGAN. The purpose of LSTM is to extract the landmarks of the mouth region. Given the landmarks as a conditional setting, CGAN synthesizes a synchronized talking face for the audio signal. Vougioukas et al. [49] converts an audio waveform file to a synchronized spoken person, without an intermediate step for extracting the landmarks as in Jalalifar et al. [61]. In this model, given the initial frame, a temporal GANs model with two discriminators, one for the frame level and the other for the sequence level, are trained to produce synchronized videos for the audios.

The approach of disentangled representations has been considered [50, 60]. The intuition behind the **Disentangled Audio-Visual System (DAVS**) [60] is to decouple the talking head information into person-related information and speech-related information in order to overcome the blurry and incoherent videos generated from a plain speech file. DAVS is able to generate videos of talking faces based on audio files or other video files, while other studies [49] [50] are conditioned only on the audio file. DAVS is able to retarget face movements from one video to another.

Mittal et al. [50] decoupled the audio into three aspects: content, emotion, and noise, contrary to other works [49, 59, 60] that utilize the audio file without any per-possessing step. Decomposing audio representations using **Variational AutoEncoder (VAE)** at the first stage facilitates discarding of the background noise that appears in real world recorded datasets. Moreover, eliminating emotion from content reduces the effect of emotion on the generated videos. In the second stage, the content component of the disentangled audio with an image from the video is fed into the generator to produce a frame. There are two discriminators in this framework: a frame level discriminator and a video level discriminator.

4.2.2 Text to Video Synthesis. This subsection considers GANs-based frameworks that aim to produce videos according to a conditional text. Table 3 summarizes the main purpose of the reviewed models. These frameworks have two main purposes. The first is to maintain semantic consistency between the condition and the generated video. The second purpose is to generate realistic quality videos that preserve the coherence and consistency within the frames. More information on the main text to video GANs frameworks, their datasets, loss function and evaluation are summarized in Table S6.

Temporal GANs conditioning on captions (TGANs-C) framework [62] first encodes the text using an LSTM based encoder. The output of the sentence encoder is concatenated with a noise vector and then given to the generator, which is a 3D deconvolution network. The model has three discriminators for the video level, frame level, and the motion level, to ensure that adjacent frames have coherent motion.

Table 4. The Reviewed Semantic Map or Missing Regions to Video GANs Frameworks in Section 4.2.3

Publication	Conditional information	Task
Vid2vid [51]	semantic video	Generating videos based on semantic images such as area division masks
Few-shot-vid2vid [52]	semantic video, initial image	Generating videos based on target image and semantic images such as area division masks
Pan et al. [63]	semantic video, initial image	Generating videos based on a single semantic map.
Chang et al. [64]	video with missing regions	Generating complete videos given the missing regions by 3D gated convolutions.
STTN [65]	video with missing regions	Generating complete videos given the missing regions by self-attention.

The second column provides the type of condition while the third one gives information on the main task.

While videos in TGANs-C [62] are generated using a single generator, the GANs framework proposed elsewhere [44] generates videos progressively using multiple generators in several stages. Firstly, the conditional variational autoencoder that is conditioned on encoded text produces the initial image. This initial image provides an overall representation, which may be the background image, the colours of the image and its structure. The initial image with the encoded text is an input to a CGAN to generate higher quality images.

The original method used in CGAN [13] to incorporate a condition is to concatenate the condition and the noise vector; this method is also applied in text to video GANs frameworks [44, 62]. However, TFGAN [55] introduces a multi-scale text-conditioning method, where the text features are extracted from the encoded text to generate convolution filters. Then, the convolution filters are input to discriminator network to facilitate strengthening of the associations between the texts and the videos.

In video generation models based on text reviewed so far [44, 55, 62], a model synthesizes a video according to one conditional sentence per video. In contrast, storyGAN [41] is a story visualization model that is conditioned on multiple sentences. StoryGAN [41] contains a context encoder and a story encoder. The story encoder encodes the entire story as a low dimensional vector that serves as an initial input to the context encoder. At each time step in the RNN context encoder, one sentence with concatenated noise is introduced along with encoded story vector to produce a Gist vector, which is combined information about a specific sentence and the story. The generator then accepts a Gist vector and produces an image. There is a discriminator for the image and another discriminator for the story. Different to other video generation frameworks, storyGAN pays less attention to the continuity of motion and instead focusses on the global consistency of the story.

4.2.3 Semantic Map/Missing Region to Video Synthesis. This section represents video GANs frameworks that are conditioned on semantic maps or masked regions, and Table 4 presents the main task of the reviewed models. This section could potentially fall under Section 4.2.5, video to video synthesis, since the frameworks are conditioned on videos. However, the conditioning videos are pre-processed into semantic maps or masked regions first, and Table S7 provides a summary of the cited semantic map or missing region to video frameworks.

Video to video (vid2vid) [51] is a conditional GANs framework that converts semantic videos into frames. Semantic videos consist of semantic maps, where each map is a collection of segmented objects that are labeled with different colours. The semantic videos could be in the form of segmentation masks or boundaries. Few-shot-vid2vid [52] is an extension to vid2vid. Both vid2vid and few-shot-vid2vid share an overall architecture for the generator network that is conditioned on the previous frame, a previous semantic image, and the source semantic videos. The generator consists of three modules, namely W to extract the optical flow, M to predict the occlusion map, and H to generate the intermediate frames. The main difference between vid2vid and few-shot-vid2vid is that the module H in vid2vid has fixed weights, whereas the weights are dynamic in few-shot-vid2vid. An adaptive network with dynamic weights [52] facilitates generation of videos of unseen objects in the training dataset by providing multiple images of the object at test time. There are two discriminators in both architectures: a video discriminator and an image discriminator. The modules are trained in a progressive manner, which means that the number of frames and the quality of generated images increase gradually.

While the conditional signal in vid2vid and few-shot-vide2vid [51, 52] is a sequence of semantic maps, Pan et al. [63] only use one semantic label map. They claim that providing a single semantic map helps loosen the restrictions during the synthesizing process. To generate a video conditioned on a single semantic map, there are two phases. The first is an image-to-image phase that is conditioned on the semantic map to generate the initial frame with fine details. The second phase produces a video given the starting frame using conditional VAE.

This section also covers the video task of video inpainting, where the model tries to fill the missing masked regions with suitable and coherent content. This task has several applications such as video retargetting, object removal and video restoration. Chang et al. [64] proposed an inpainting framework with a 3D gated convolution network as generator to handle unmasked features. The discriminator is Temporal PatchGAN which is trained to balance between local and global features because the mask could be in any location in the frame. In addition, Temporal PatchGAN considers temporal features by using 3D convolutional layers. One limitation in their proposed model is that the generator can capture information from nearby frames but cannot capture content for distant frames. **Spatial-Temporal Transformer Network (STTN)** [65] retains Temporal PatchGAN [64] and changes the generator to transformers with attention mechanism to extract content from distant frame dependencies.

4.2.4 Image to Video Synthesis. The main purpose of image-to-video GANs frameworks is to predict future frames based on a given frame, and Table 5 provides the main aim of these frameworks and Table S8 lists the reviewed image to video frameworks and some of their properties. Early versions of image to video architectures [66–69] do not disentangle the representations of the training videos, resulting in blurriness in the synthesized videos. Mathieu et al. [66] made the first attempt to employ adversarial training in the video prediction domain. The generator is a multiscale network that is focussed on synthesizing coherent frames conditioned on front frames. The adversarial network solves the issue related to blurry frames that results from standard mean squared error loss function. Lee et al. [67] incorporate VAEs with GANs for a video prediction system. Combining VAEs with GANs was first performed in the image domain [4, 70]. The reason for using both GANs and VAEs networks is that a GANs model helps produce more realistic images, while VAE facilitates diversity in the generated images. **Multi-Discriminator GAN (MD-GAN)** [68] employs two consecutive GANs. While the first generates the content of the frames, the other GANs model refines the output of the first stage. Similar to MD-GAN [68], the model by Cai et al. [69] adopts a two-stage framework based on pose estimation. The first network generates human pose sequences of the conditioned type of pose, whereas the second stage network maps from the pose space to pixel space, given a reference image.

Several studies [45, 53, 71–74] on frame prediction have shown that decomposing the information in videos enhances the quality of the synthesized videos. **Motion Content Network (MCnet)** [71] encodes motion and content using separate encoders in an unsupervised manner. The outputs

Table 5. The Reviewed Image to Video GANs Frameworks in Section 4.2.4

Publication	Conditional information	Task
Mathieu et al. [66]	input frames	Predicting future frames using multiscale network.
Lee et al. [67]	input frame	Predicting future frames using combination of GAN and VAE.
MCnet [71]	input frames	Predicting future frames by dealing with motion and content separately.
Walker et al. [72]	input frames	Predicting future frames by extracting human skeleton.
TwoStreamVAN [45]	input frame, category, motion map	Predicting future frames by dealing with motion and content separately.
Liang et al. [73]	input frames	Predicting future frames by dealing with motion and content separately.
DRNET [74]	input frames	Predicting future frames by dealing with motion and content separately.
Hu et al. [53]	input frame, stroke	Predicting future frames given the first frame and the guided motion stroke.
MD-GAN [68]	input frame	Predicting future frames using two GANs one for generation and one for refinement.
Cai et al. [69]	input image, label	Generating, predicting, and completing frames.
FW-GAN [76]	two images, motion map, previous frames	Generating colored videos given the gray-scale version and example of a style colored image.
Zhang et al. [75]	two images, previous frame	Generating video of person trying a piece of cloth given image of person, desired cloth, and desired poses.

The second column provides the type of condition while the third one gives information on the main task.

of the previous stages are combined and decoded to produce the next frame. Walker et al. [72] also tackle content and motion in a progressive manner by utilizing two architectures, namely VAEs and GANs, as do Lee et al. [67]. The VAEs framework predicts future human poses and motions, while the GANs framework is conditioned on the motions to predict future frames as a pixel-level representation. Similar to Villegas et al. [71] and Walker et al. [72], both Sun et al. [45] and Liang et al. [73] attempt to separate the dynamics from the content by employing dual GANs. One of the GANs is dedicated to generating the content of a frame, while the other GANs model predicts the dynamics. The difference between the two architectures is that Liang et al. [73] use one encoder to encode the front frames, while TwoStreamVAN [45] encodes an initial frame with the content encoder and the motion map is encoded using the motion encoder. Using same approach as TwoStreamVAN, DRNET [74] has two encoder networks: one for the dynamic content and the other for time-independent content. The encoded representations are concatenated and fed into the decoder to predict the next frames. When decomposing motion from content to forecast a future frame, the motion component is detected from the input frames [45, 71–74]. However, Hu et al. [53] utilize a motion stroke, which is a continuous line that represents object motion, instead of extracting motion from the input frames. The model first encodes the initial image and the motion strokes. This is followed by an additional encoder to encode the encoded initial image, encoded motion strokes and features of the previous frame. Next, the generator is used to generate the sequence frames with two discriminators, one each for image level and sequence level.

Table 6. The Reviewed Video to Video GANs Frameworks in Section 4.2.5

Publication	Conditional information	Task
Zhou et al. [78]	video, image	Generating videos of a dancing person different from input person while maintaining motion as the input video using body parts heat map.
Chan et al. [77]	video	Generating videos of a dancing person different from input person while maintaining motion as the input video using joint map.
Vid2Game [83]	video, control signal	Generating videos and control the pose of the person in the frame by control signals.
Monkey-net [81]	video, image	Generating videos of a person different from input person while maintaining motion as the input video using key points.
Recycle-GAN [87]	video	Generating videos with styles as the given video but different content from the input video.
Liu et al. [82]	video, image	Generating videos of a person different from input person while maintaining motion as the input video using 3D character model.
Deep Video Portraits [80]	video	Generating talking person where the face is different from input face while maintaining motion as the input video using monocular face.
ReenactGAN [84]	video	Generating talking person where the face is different from input face while maintaining motion as the input video using face's boundary.
TransMoMo [79]	video	Generating videos of a person different from input person while maintaining motion as the input video using skeleton.

The second column provides the type of condition while the third one gives information on the main task.

The dominant application of image-to-video GAN frameworks is video prediction applications, yet there are other applications such as video colorization [75] and video virtual try-on [76]. Zhang et al. [75] introduce GANs in exemplar-based colorization problems. The framework can propagate the color from the style frame to a video while maintaining temporal consistency. The framework learns the semantic similarity between the style frame and the target frame to guide the colorization process. Dong et al.'s [76] model was the first attempt at applying GANs to a video virtual fitting application. The proposed architecture consists of four encoders to encode the target person image, clothing image, past generated frame sequence and joint-point information sequence. There are two decoders, one to predict the optical flow, the other to wrap the clothes with the encoded flow and encoded feature from the second decoder to produce the target person with the desired clothes.

4.2.5 Video to Video Synthesis. One of the major applications of video to video synthesis is object animation, where motion is retargeted from one object to another. Some GANs frameworks in this domain are limited to a specific dataset, while others can be applied in different domains; Table 6 summarizes the domains these frameworks are applied to and Table S9 lists other properties of each framework. Many works [77–79] share the main objective of converting a person's dance movements to those of another immature dancer. Zhou et al. [78] build their architecture in a progressive manner, where the first phase synthesizes frames for the immature dancer based on the pose of the mature dancer and body parts of the immature dancer. The second network provides more realism to the final output by fusing the target performer with a background and adding necessary shadows to combine the foreground with the background. Chan et al. [77] start with a pose detector network, whose result is inserted as an input to a GANs framework. Then, the GANs face framework is used to enhance realism. Yang et al.'s architecture [79] is similar to Kim et al.'s [80], where the model starts with disentangling the video representations of the source and target videos into distinct parameters that can be combined to produce retargeted video. Often, the datasets [77, 79] are limited, with a small number of participants performing a wide range of movements. Thus, Chan et al's [77] computational module translates the poses more easily than Zhou et al.'s [78], as the latter's dataset is collected from YouTube and it is not feasible to collect different poses of the same person.

Studies by Siarohin et al. [81] and Liu et al. [82] aim to generate a video that has similar action as an input video and containing an object similar to ones in an input image; in these models, the action is not restricted to a dance movement. When comparing the methods for pose extraction, it is important to mention that models by Zhou et al. Chan et al. [77, 78] are based on pre-trained networks for extracting subject movements. In contrast, **MOviNg KEYpoints (Monkey-net)** [81] is based on a self-supervised framework while Liu et al. [82] used 3D reconstruction software to rebuild the desired poses. In addition, other generation processes [52, 78, 81, 82] are conditioned on a specific human target, whereas Chan et al. [77] use a random human subject from the dataset in generated videos. Monkey-net is composed of three stages: the first stage extracts key points of two random images of the input video; the second stage is a motion prediction network that computes the optical flow of the result of the previous stage; and the last stage performs image synthesis using a Variational AutoEncoder. Liu et al. [82] start with extracting a 3D object from the static images and motion of the given video, and end with a CGAN framework to produce realistic frames.

vid2Game [83] follows a different approach to control the human subject in generated frames. The synthesized images are conditioned on a low dimensional signal such as joystick movements. There are two networks: the first is Pose2pose, which generates a new pose given the current one and the control signal using an autoencoder. The control signal is input to the residual blocks in the middle of the autoencoder to facilitate smooth motion. The second network is the Pose2frame network, which generates a mask and an image. The image is combined with the mask to produce a frame. In other work [51, 52, 77, 78], the generator network is conditioned on the pose extracted from real images of the training dataset, while in vid2Game [83], the network is conditioned on synthesized poses. Clearly, working with poses extracted from synthesized frames requires more effort compared to working with poses from the training images, as artefacts in the generated poses need to be dealt with.

While some studies [64, 65, 68, 95] retarget the motion of an individual body to another individual, others focus only on the head [80, 84]. This application can be between images [85, 86], or between image and video [80, 84]. Deep video portraits [80] outputs transferred motion video that can be modified by changing adjustable parameters such as head pose and facial expression. Deep video portraits translate both source and target videos to low dimensional parameter vectors that are adjusted to input to a rendering video. ReenactGAN [84] extracts the face boundary as an initial step, then the source boundary is aligned with the target boundary to produce target video animation. Wu et al. [84] claim that ReenactGAN is better at representing minor changes in the face between frames than other frameworks [80].

Recycle-GAN [87] is another video retargetting application that takes a different approach. While other frameworks transfer motion from one object to another [64, 65, 68, 95, 80, 84], Recycle-GAN converts a sequence of frames from one domain to another while maintaining the style of the second domain. A general object retargetting model is trained on only the source domain, and the target domain video is provided at testing time. In contrast, Recycle-GAN contains two GANs frameworks trained on two datasets from the source and target domains. Unlike previous video retargetting models [77, 78, 81, 83], Recycle-GAN is conditioned on videos from one domain, and not on semantic maps or detected poses. Recycle-GAN is an extension of Cycle-GAN [88], which is an unsupervised method used in the image domain to translate unpaired images form one domain to another by applying the cycle consistency loss. Additionally, Recycle-GAN incorporates spatial and temporal constraints within a GANs framework.

5 DISCUSSION

The main objective in image generation GANs is to ensure that the generated images are realistic while video GANs extend this objective to maintain smooth motion between the generated frames. There are different methods to enforce the second objective including applying flow loss function [58, 59, 74] or adding temporal discriminator [42, 52]. Other methods to deal with the generator architecture include 3D convolution, latent space disentanglement and RNN-based architectures. This section provides an overview of the evaluation metrics, datasets and loss functions that are widely used in video GANs.

5.1 Loss function

One reason that GANs generated results surpass VAE outcomes is the use of adversarial loss function in GANs to produce more appealing images than the reconstruction loss in VAE [3, 4]. Early video GANs models tended to use adversarial loss alone [6, 42]. However, one problem with the adversarial loss is that the produced images may have blended unseen artefacts since the model tries to make the generated images close to the real images. Thus, current studies include additional reconstruction loss such as L1 and L2 losses that help in eliminating the hallucinatory artefacts by penalizing at the pixel level if there are unseen artefacts that are not present in the real images [89]. It is worth mentioning that L1 loss produces less blurry images than L2 based loss in experiments [90–92], and so L1 is widely used in video GANs models. VAE includes **Kullback-Leibler (KL)** in its objective function in addition to the reconstruction loss. While the reconstruction loss is used to minimise the distance between generated images and real ones, KL is used as a regulariser to avoid overfitting and generate images as copies of the real ones. The VAE architecture is introduced in video GANs for multiple reasons. Mittal et al. [50] utilise VAE as a pre-processing step to disentangle content from emotion in audio representation. VAE is also used to encode a signal in a low dimension tensor [41, 63]. While the reconstruction loss is used to calculate the distance between two data distributions, triplet loss or ranking loss can measure the distance between three distributions. MD-GAN [68] and TransMoMo [79] utilise triplet loss to compare the motion of three motion sequences. The objective is to make the generated video closer to a real body sequence while maintaining a distance from the unrefined video in terms of motion. While the mentioned reconstruction losses are pixel-wise losses, perceptual loss compares in terms of the semantics between images. It is used to compare high level features such as content and style differences. Perceptual loss was first introduced in the style transfer domain [93]. Then, it was adopted in general contexts to produce sharper images. Usually, high level features are extracted using pretrained VGGNet that is trained on ImageNet first. Then, perceptual loss is calculated between generated images and real ones. Feature Matching Loss is similar to perceptual loss, with the main difference being that perceptual loss uses pre-trained VGGNet to extract the features while the feature matching loss used the discriminator model. Both feature matching loss and perceptual loss help in faster training and convergence. However, because feature matching loss and perceptual loss tend to have faster training, it results in fewer deep features with less fine details and realistic textures [94]. These loss functions are the ones which are commonly used, but an exhaustive list of all the different loss functions used in the reviewed paper are presented in Table S3.

5.2 Dataset

Video generation is an application of deep learning models that requires a myriad of data samples to produce high quality generations. When surveying the video GANs literature, we noticed that the datasets in these papers were originally used for other purposes. For example, UCF-101 and Kinetics dataset are used in action recognition. Some of the papers created their own datasets in order to have a specific feature that is not available in the existing dataset [6, 64, 78, 82]. However, most of these papers used at least one of the existing datasets in order to evaluate their model. In this section we cover the datasets more commonly used, but Table S1 lists all datasets used in the reviewed papers coded with prefix "D" and a number (e.g. D1 refers to Clever-sv dataset and so on). The table also presents the number of videos in the dataset, duration (if available), resolution and purpose. UCF101 [95] is the most cited dataset, with its samples collected from YouTube and has 101 action categories such as human interactions, sports, and playing musical instruments. This dataset is a benchmark for unconditional applications and prediction applications. This dataset is widely used because it is not as simple as Moving MNIST [96] that has only 10 classes, and not as complex as Kinetics human action [97] that has 600 classes. Kinetics is the second most used benchmark dataset that was created by Clark et al. [48] to illustrate their generative model's performance on a complex dataset. Kinetics is a human action dataset that is collected from YouTube as well with some similar categories as UCF101. Some works used Kinetics as a benchmark but using only a subset of the dataset [44, 98]. In speech synchronisation applications, there are two well-known datasets: one is GRID [99] while the other one is **Lip Reading in the Wild (LRW)** [100]. Both datasets have audio-visual samples. GRID consists of 1,000 sentences that are uttered by 34 speakers, while LRW has 500 words that are spoken by 1,000 speakers. LRW is more complex than GRID since LRW is extracted from British television programs, while GRID is recorded in an experimental office environment. The grammar in LRW is more diverse than GRID that follows simple grammar structures. Most common datasets that are used in image to video mapping include Human3.6M [101], Moving MNIST [96], and Weizmann human action [102]. Weizmann human action has 90 videos of 9 actors doing 10 actions, while Human3.6M consists of 800 sequences of 3D human skeletons of 11 actors. Moving MNIST is based on MNIST dataset which includes images of handwritten digits on a solid background. Moving MNIST extends MNIST by moving the digits around the frames. Other categories lack well-known datasets that are widely used.

5.3 Evaluation Metrics

The GANs evaluation metrics can be categorised into quantitative and qualitative metrics. Human evaluation is a qualitative metric that is used widely in GANs. Observers are given generated videos to evaluate and compare with synthesised videos from competitive benchmark models. The common criteria that observers evaluate against in the video GANs realm include naturalness of movement, realism of the frames, diversity of the samples, and synchronisation. However, according to Alqahtani et al. [103], one of the limitations associated with human evaluation is the generated samples visually might not necessarily reflect the model capabilities.

For quantitative measures, **Inception Score (IS)** [104], **Fréchet Inception Distance (FID)** [105], **Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM)** [106], and **Mean Squared Error (MSE)** [107] are prevalent for video GANs. IS [104] is proposed as an alternative measure to human evaluations in GAN frameworks. IS takes into consideration the quality and diversity of the generated images by computing the classification score and marginal distribution, respectively. IS score only considers the generated frames and does not account for the original frames. FID [105] overcame this limitation by comparing the synthesised videos and original videos. To calculate FID, first, spatio-temporal feature maps need to be extracted. Then, the mean and covariance matrixes are computed for all generated videos and training videos. Then, a combination of these statistics from the training videos and the synthesised videos is used to calculate the FID score. While IS and FID scores require a pre-trained model to extract the features, PSNR, MSE, and SSIM are computed at the pixel level. MSE represents the sum of the square of the distance between the original frames and generated ones. MSE is closely related to PSNR because the latter is the ratio between the maximum signal intensity and MSE. SSIM evaluates based on three aspects which are luminance, contrast, and structure of the original images and generated ones. SSIM has a stronger relationship with human judgment than PSNR since SSIM mimics the human visual system by considering the mentioned three factors [108].

6 CONCLUSION

Generative models such as GANs provide promising results in multiple domains including images, videos, audios and texts. Video synthesis is still in the early stages compared to other domains such as images. The current state of the art for video GANs suffers from low quality frames or low number of frames or both. One reason could be the higher requirement for computational power as videos are high dimensional data, and so necessitate networks with a large number of parameters. To handle such data, there is a need for a complex architecture that takes into consideration spatial as well as temporal data. For example, DVD-GAN uses one TPU and TGANv2 utilizes 8 GPUs. In addition, videos are usually multimodal and may include audio stream as well, which makes the processing even more complex. Collecting domain-specific videos is also more time-consuming and expensive comparing to other domains such as images, as automatic video retrieval algorithms are not yet very accurate and video data collection involves a lot of manual work to select, clean and pre-process the data. Nevertheless, the trend is upward and every year more studies are being done in this area. The applications of video GANs are broad and include speech animation, video prediction, video retargetting, generating stories from caption and video completion.

Although the progress on GANs in areas other than videos is well documented through several review papers, video GANs models have received less attention so far, and if at all included, they were only a section in other review papers despite their broad range. Considering the increasing number of studies on video GANs during the past few years, it is the right time to survey the field, categorise different models according to their applications and compare their differences. This paper is among the initial attempts to review GANs models that produce videos and highlight their main differences.

While 3D CNN GANs as proposed by Vondrick et al. [6] appear to be an intuitive choice to synthesise videos and represent frames along with time dimension, 3D convolutions may cause overfitting [109]. An alternative to 3D CNN is to utilize RNN with 2D convolution, as in MoCoGAN [42]. Using 2D convolution and 1D convolution to disentangle content from the motion dimension is another way to represent videos [54].

Videos can be generated using GANs either without conditional settings or by introducing a conditional signal such as an audio, image, video, text, label or semantic map. This survey paper initially groups the video GANs frameworks according to their conditional setting: unconditional video generation vs conditional video generation, and discusses the most important models proposed so far in each category and outlines the differences. Moreover, it goes deeper into the conditional frameworks and categorizes the methods according to their condition and presents and compare the different models in each of those categories. In addition, a list of all datasets used in the reviewed frameworks, their characteristics, evaluation metrics and the loss function applied in each work are presented in supplementary material. The hope is that this paper will serve as a good review of the domain so far and provide the reader with a better understanding of different frameworks and their potential applications.

It is important to note that this paper did not evaluate and compare video GANs models in terms of performance, as we do not believe this is practical for several reasons; First, input signals are different even for video synthesis frameworks that fall under the same application. For example, in the speech synchronization application, some GANs models use an image and audio as inputs [50, 110] while Jalalifar et al. [61] utilize lip landmarks as an additional input. Another hurdle is that video GANs models have been implemented on different datasets, and some models use a subset of another existing dataset. The entire Kinetics dataset is used in DVD-GAN [48] while in other models [44, 55], only a subset is used to train their models. Until now, there are no benchmark datasets for video synthesis that help researchers to compare the current state-of-the-art video GANs models and this can be one important future direction for the field.

REFERENCES

[1] I. Goodfellow et al. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] P. K. Diederik and M. Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR).*

[3] M. El-Kaddoury, A. Mahmoudi, and M. M. Himmi. 2019. Deep generative models for image generation: A practical comparison between variational autoencoders and generative sdversarial networks. In *International Conference on Mobile, Secure, and Programmable Networking*, 2019: Springer, pp. 1–8.

[4] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300.*

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen. 2017. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196.*

[6] C. Vondrick, H. Pirsiavash, and A. Torralba. 2016. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[7] Y. Hong, U. Hwang, J. Yoo, and S. Yoon. 2019. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 10.

[8] A. Jabbar, X. Li, and B. Omar. 2020. A survey on generative adversarial networks: Variants, applications, and training. *arXiv preprint arXiv:2006.05132.*

[9] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar. Applications of generative adversarial networks (GANs): An updated review. *Archives of Computational Methods in Engineering,* pp. 1–28.

[10] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye. 2020. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937.*

[11] A. Radford, L. Metz, and S. Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434.*

[12] I. Goodfellow. 2016. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160.*

[13] M. Mirza and S. Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784.*

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[15] A. Odena, C. Olah, and J. Shlens. 2017. Conditional image synthesis with auxiliary classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017: JMLR. org, pp. 2642–2651.

[16] A. Brock, J. Donahue, and K. Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096.*

[17] T. Karras, S. Laine, and T. Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[18] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. 2020. Analyzing and improving the image quality of styleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.

[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. 2019. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, 2019, pp. 7354–7363.

[20] M. Arjovsky, S. Chintala, and L. Bottou. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875.*

[21] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng. 2019. Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access,* vol. 7, pp. 36322–36333.

[22] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. 2018. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65.

[23] M. Zamorski, A. Zdobylak, M. Zięba, and J. Świątek. 2019. Generative Adversarial Networks: recent developments. In *International Conference on Artificial Intelligence and Soft Computing*, 2019: Springer, pp. 248–258.

[24] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang. 2017. Generative adversarial networks: Introduction and outlook. *IEEE/CAA Journal of Automatica Sinica* 4, 4 (2017), 588–598.

[25] S. Hitawala. 2018. Comparative study on generative adversarial networks. *arXiv preprint arXiv:1801.04271.*

[26] Z. Wang, Q. She, and T. E. Ward. 2019. Generative adversarial networks: A survey and taxonomy. *arXiv preprint arXiv:1906.01529.*

[27] K. Cheng, R. Tahir, L. K. Eric, and M. Li. An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. *Multimedia Tools and Applications,* pp. 1–28.

[28] D. Saxena and J. Cao. 2020. Generative adversarial networks (GANs): Challenges, solutions, and future directions. *arXiv preprint arXiv:2005.00065.*

[29] Y. LeCun, C. Cortes, and C. Burges. 2010. MNIST handwritten digit database.

[30] S. N. Esfahani and S. Latifi. A Survey of the State-of-the-Art GAN-based approaches to image synthesis.

[31] H. Huang, P. S. Yu, and C. Wang. 2018. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469.*

[32] X. Wu, K. Xu, and P. Hall. 2017. A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science and Technology* 22, 6 (2017), 660–674.

[33] Y.-J. Cao et al. 2018. Recent advances of generative adversarial networks in computer vision. *IEEE Access,* vol. 7, pp. 14985–15006.

[34] J. Agnese, J. Herrera, H. Tao, and X. Zhu. 2019. A Survey and Taxonomy of Adversarial Neural Networks for Text-to-Image Synthesis. *arXiv preprint arXiv:1910.09399.*

[35] H. Xiao, K. Rasul, and R. Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747.*

[36] X. Yi, E. Walia, and P. Babyn. 2019. Generative adversarial network in medical imaging: A review. *Medical ImageA-analysis,* p. 101552.

[37] V. Sorin, Y. Barash, E. Konen, and E. Klang. 2020. Creating artificial images for radiology applications using generative adversarial networks (GANs)–A systematic review. *Academic Radiology.*

[38] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi. 2019. A survey on GANs for anomaly detection. *arXiv preprint arXiv:1906.11632.*

[39] N. Torres-Reyes and S. Latifi. Audio enhancement and synthesis using generative adversarial networks: A survey. *International Journal of Computer Applications,* vol. 975, p. 8887.

[40] C. Yinka-Banjo and O.-A. Ugot. 2019. A review of generative adversarial networks and its application in cybersecurity. *Artificial Intelligence Review,* pp. 1–16.

[41] Y. Li et al. 2019. StoryGAN: A sequential conditional GAN for story visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6329–6338.

[42] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. 2018. MocoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1526–1535.

[43] B. Duan, W. Wang, H. Tang, H. Latapie, and Y. Yan. 2019. Cascade attention guided residue learning GAN for Cross-Modal translation. *arXiv preprint arXiv:1907.01826.*

[44] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin. 2018. Video generation from text. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

[45] X. Sun, H. Xu, and K. Saenko. 2018. A two-stream variational adversarial network for video generation. *arXiv preprint arXiv:1812.01037.*

[46] Q. Liu, H. Zhou, Q. Xu, X. Liu, and Y. Wang. 2020. PSGAN: A generative adversarial network for remote sensing image pan-sharpening. *IEEE Transactions on Geoscience and Remote Sensing.*

[47] W. Yu, M. Zhang, Z. He, and Y. Shen. 2021. Convolutional two-stream generative adversarial network-based hyperspectral feature extraction. *IEEE Transactions on Geoscience and Remote Sensing.*

[48] A. Clark, J. Donahue, and K. Simonyan. 2019. Efficient video generation on complex datasets. *arXiv preprint arXiv:1907.06571.*

[49] K. Vougioukas, S. Petridis, and M. Pantic. 2018. End-to-end speech-driven facial animation with temporal GANs. *arXiv preprint arXiv:1805.09313.*

[50] G. Mittal and B. Wang. 2020. Animating face using disentangled audio representations. In *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 3290–3298.

[51] T.-C. Wang et al. 2018. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601.*

[52] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro. 2019. Few-shot video-to-video synthesis. *arXiv preprint arXiv:1910.12713*.

[53] Q. Hu, A. Waelchli, T. Portenier, M. Zwicker, and P. Favaro. 2018. Video synthesis from a single image and motion stroke. *arXiv preprint arXiv:1812.01874*.

[54] M. Saito, E. Matsumoto, and S. Saito. 2017. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2830–2839.

[55] Y. Balaji, M. R. Min, B. Bai, R. Chellappa, and H. P. Graf. 2019. Conditional GAN with discriminative filter generation for text-to-video synthesis. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019: AAAI Press, pp. 1995–2001.

[56] K. Ohnishi, S. Yamamoto, Y. Ushiku, and T. Harada. 2018. Hierarchical video generation from orthogonal information: Optical flow and texture. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[57] M. Saito and S. Saito. 2018. TGANv2: Efficient training of large models for video generation with multiple subsampling layers. *arXiv preprint arXiv:1811.09245*.

[58] Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva. 2020. G3AN: Disentangling appearance and motion for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5264–5273.

[59] L. Chen, Z. Li, R. K. Maddox, Z. Duan, and C. Xu. 2018. Lip movements generation at a glance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 520–535.

[60] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang. 2019. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 9299–9306.

[61] S. A. Jalalifar, H. Hasani, and H. Aghajan. 2018. Speech-driven facial reenactment using conditional generative adversarial networks. *arXiv preprint arXiv:1803.07461*.

[62] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei. 2017. To create what you tell: Generating videos from captions. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017, pp. 1789–1798.

[63] J. Pan et al. 2019. Video generation from single semantic label map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3733–3742.

[64] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu. 2019. Free-form video inpainting with 3D gated convolution and temporal patchGAN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9066–9075.

[65] Y. Zeng, J. Fu, and H. Chao. 2020. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, 2020: Springer, pp. 528–543.

[66] M. Mathieu, C. Couprie, and Y. LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*.

[67] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. 2018. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*.

[68] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo. 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2364–2373.

[69] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang. 2018. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 366–382.

[70] J.-Y. Zhu et al. 2017. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017, pp. 465–476.

[71] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. 2017. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*.

[72] J. Walker, K. Marino, A. Gupta, and M. Hebert. 2017. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3332–3341.

[73] X. Liang, L. Lee, W. Dai, and E. P. Xing. 2017. Dual motion GAN for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1744–1752.

[74] E. L. Denton. 2017. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, 2017, pp. 4414–4423.

[75] B. Zhang et al. 2019. Deep exemplar-based video colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8052–8061.

[76] H. Dong, X. Liang, X. Shen, B. Wu, B.-C. Chen, and J. Yin. 2019. Fw-GAN: Flow-navigated warping GAN for video virtual try-on. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1161–1170.

[77] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. 2019. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5933–5942.

[78] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg. 2019. Dance dance generation: Motion transfer for internet videos. *arXiv preprint arXiv:1904.00129.*

[79] Z. Yang et al. 2020. TransMoMo: Invariance-driven unsupervised video motion retargeting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5306–5315.

[80] H. Kim et al. 2018. Deep video portraits. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

[81] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. 2019. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2377–2386.

[82] L. Liu et al. 2019. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 1–14.

[83] O. Gafni, L. Wolf, and Y. Taigman. 2019. Vid2game: Controllable characters extracted from real-world videos. *arXiv preprint arXiv:1904.08379.*

[84] W. Wu, Y. Zhang, C. Li, C. Qian, and C. Change Loy. 2018. ReenactGAN: Learning to reenact faces via boundary transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[85] J. Li, Z. Li, J. Cao, X. Song, and R. He. 2021. FaceInpainter: High Fidelity Face Adaptation to Heterogeneous Domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5089–5098.

[86] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen. 2019. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457.*

[87] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh. 2018. Recycle-GAN: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–135.

[88] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[89] J. P. Cohen, M. Luck, and S. Honari. 2018. Distribution matching losses can hallucinate features in medical image translation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2018: Springer, pp. 529–536.

[90] A. Pandey and D. Wang. 2018. On adversarial training and loss functions for speech enhancement. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018: IEEE, pp. 5414–5418.

[91] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[92] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[93] J. Johnson, A. Alahi, and L. Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016: Springer, pp. 694–711.

[94] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

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

[96] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278–2324.

[97] C. Schuldt, I. Laptev, and B. Caputo. 2004. Recognizing human actions: A local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2004, vol. 3: IEEE, pp. 32–36.

[98] Y. Balaji, M. R. Min, B. Bai, R. Chellappa, and H. P. Graf. 2018. TFGAN: Improving conditioning for Text-to-Video synthesis.

[99] M. Cooke, J. Barker, S. Cunningham, and X. Shao. 2006. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America* 120, 5 (2006), 2421–2424.

[100] J. S. Chung and A. Zisserman. 2016. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016: Springer, pp. 87–103.

[101] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. 2013. Human3. 6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, no. 7, pp. 1325–1339.

[102] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. 2005. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005, vol. 2: IEEE, pp. 1395–1402.

[103] H. Alqahtani, M. Kavakli-Thorne, G. Kumar, and F. SBSSTC. 2019. An analysis of evaluation metrics of GANs. In *International Conference on Information Technology and Applications (ICITA).*

[104] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[105] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.

[106] Z. Wang, E. P. Simoncelli, and A. C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, vol. 2: IEEE, pp. 1398–1402.

[107] Z. Wang and A. C. Bovik. 2009. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine,* vol. 26, no. 1, pp. 98–117.

[108] A. Horé and D. Ziou. 2013. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? *IET Image Processing,* vol. 7, no. 1, pp. 12–24.

[109] R. Pascanu, T. Mikolov, and Y. Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[110] L. Chen, S. Srivastava, Z. Duan, and C. Xu. 2017. Deep cross-modal audio-visual generation. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017: ACM, pp. 349–357.

[111] W. Zhang, M. Zhu, and K. G. Derpanis. 2013. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2248–2255.

[112] N. Harte and E. Gillen. 2015. TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia,* vol. 17, no. 5, pp. 603–615.

[113] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang. 2017. Deepstory: Video story qa by deep embedded memory networks. *arXiv preprint arXiv:1707.00836.*

[114] F. Ebert, C. Finn, A. X. Lee, and S. Levine. 2017. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268.*

[115] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. 2018. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179.*

[116] N. Aifanti, C. Papachristou, and A. Delopoulos. 2010. The MUG facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, 2010: IEEE, pp. 1–4.

[117] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[118] A. Gorban et al. 2015. THUMOS challenge: Action recognition with a large number of classes. ed.

[119] Y. LeCun, F. J. Huang, and L. Bottou. 2004. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004, vol. 2: IEEE, pp. II–104.

[120] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. 2017. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746–1754.

[121] C. Richie, S. Warburton, and M. Carter. 2009. *Audiovisual Database of Spoken American English.* Linguistic Data Consortium.

[122] G. Varol et al. 2017. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 109–117.

[123] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma. 2014. CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing,* vol. 5, no. 4, pp. 377–390.

[124] T. Afouras, J. S. Chung, and A. Zisserman. 2018. LRS3-TED: A large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496.*

[125] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.

[126] B. Schiele, P. Dollár, C. Wojek, and P. Perona. 2009. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition (CVPR).*

[127] H. Dibeklioğlu, A. A. Salah, and T. Gevers. 2012. Are you really smiling at me? Spontaneous versus posed enjoyment smiles. In *European Conference on Computer Vision*, 2012: Springer, pp. 525–538.

[128] S. R. Richter, Z. Hayder, and V. Koltun. 2017. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2213–2222.

[129] M. Cordts et al. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[130] X. Huang et al. 2018. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.

[131] G. Mittal, T. Marwah, and V. N. Balasubramanian. 2017. Sync-draw: Automatic video generation using deep recurrent attentive architectures. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017, pp. 1096–1104.

[132] D. Chen and W. B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 190–200.
[133] N. Xu et al. 2018. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*.
[134] S. Caelles et al. 2018. The 2018 Davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*.
[135] videvo. “videvo.” https://www.videvo.net/(accessed 2021).
[136] M. Marszalek, I. Laptev, and C. Schmid. 2009. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009: IEEE, pp. 2929–2936.

Received August 2020; revised August 2021; accepted September 2021