Disentangling Content and Motion for Text-Based Neural Video Manipulation

Levent Karacan[1]
levent.karacan@iste.edu.tr
Tolga Kerimoğlu, İsmail İnan[2]
{tolga.kerimoglu,ismail.inan}@boun.edu.tr
Tolga Birdal[3]
tbirdal@imperial.ac.uk
Erkut Erdem[4]
erkut@cs.hacettepe.edu.tr
Aykut Erdem[5]
aerdem@ku.edu.tr

[1] Computer Engineering Department
Iskenderun Technical University
[2] Computer Engineering Department
Boğaziçi University
[3] Department of Computing
Imperial College London
[4] Computer Engineering Department
Hacettepe University
[5] Computer Engineering Department
Koc University

Abstract

Giving machines the ability to imagine possible new objects or scenes from linguistic descriptions and produce their realistic renderings is arguably one of the most challenging problems in computer vision. Recent advances in deep generative models have led to new approaches that give promising results towards this goal. In this paper, we introduce a new method called DiCoMoGAN for manipulating videos with natural language, aiming to perform local and semantic edits on a video clip to alter the appearances of an object of interest. Our GAN architecture allows for better utilization of multiple observations by **di**sentangling **co**ntent and **mo**tion to enable controllable semantic edits. To this end, we introduce two tightly coupled networks: (i) a *representation network* for constructing a concise understanding of motion dynamics and temporally invariant content, and (ii) a *translation network* that exploits the extracted latent content representation to actuate the manipulation according to the target description. Our qualitative and quantitative evaluations demonstrate that DiCoMoGAN significantly outperforms existing frame-based methods, producing temporally coherent and semantically more meaningful results.

1 Introduction

Making desired edits on an image or video using tools like Adobe Photoshop, Adobe Premiere Pro and Apple Final Cut Pro is quite challenging and requires extensive training and experience. Thanks to the proliferation of deep learning, some user-friendly solutions are proposed for editing images [33]. Yet, democratizing the video editing process to improve accessibility and empower the non-experts still requires rethinking modern architectures.

Towards this end, we set off to ask: *Can we learn to semantically manipulate videos through natural language descriptions in a temporally consistent way?* (*c.f.* Fig. 1) The existing literature approaches this problem on a frame-by-frame basis applying minimal necessary modifications specified by the input text, disjointly to the each and every input frame.

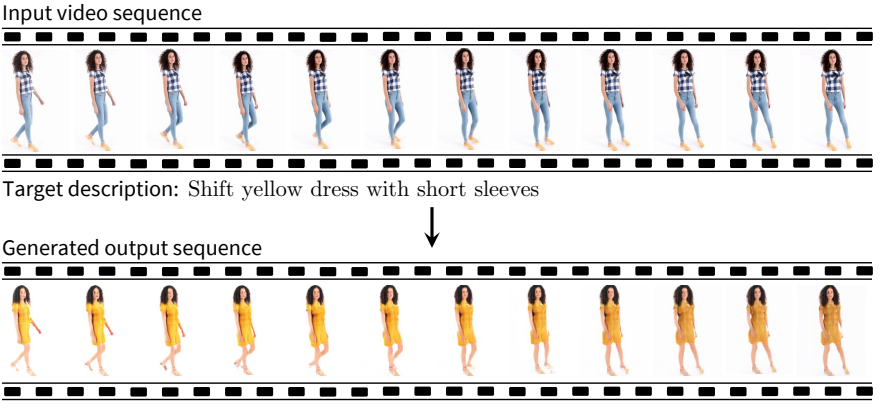

Figure 1: **Text-based video manipulation.** Given a video sequence and a target description, our DiCoMoGAN model generates a temporally coherent output sequence, carrying out the necessary structural changes while preserving the attributes not referred in the text (*e.g.* hair style, identity), and does this without any extra guidance like semantic layout information.

Almost all of the image editing methods use encoder-decoder architectures [14, 33, 34, 41, 44, 46, 61] and employ adversarial learning strategies [26, 30] to provide the agreement between the resulting images and the target text and to generate photo-realistic outcomes. However, performing language-driven edits on videos requires models to not only understand the frame content but also be aware of the global video context and its temporal unidirectionality. Moreover, a harmonious editing demands the target descriptions to be related not only to static frames but to the entire video to achieve *good gestalt*.

To achieve all these, we propose a new data-driven text based video manipulation model called **DiCoMoGAN**. The key to our approach is a unified network model consisting of a *representation network* (RepNet) and a *translation network* (TraNet), which jointly learn to disentangle video content and motion dynamics and to perform the text-specified edits on a given video sequence. Under the assumption that textual description is strongly related to appearance, we create a structured latent space composed of *text relevant*, *text irrelevant* and *dynamic* subspaces (*c.f.* Fig. 2). To ensure the former, we steer the latent subspace to be shared between global video descriptor and the text features, encoded by CLIP [44]. We then use the features from this structured latent space along with text features to condition *multi-feature modulation* (MFMOD) blocks. We train this integrated architecture via a multi-task loss function in an end to end manner to encode scene specific transformations effectively while capturing the relationships between the spatiotemporal data and the text input. Our experiments on the standard 3D Shapes benchmark [5] as well as on our new dataset *Fashion Videos* demonstrate that DiCoMoGAN can produce high quality, temporally consistent videos faithfully reflecting the *intentions* stated in the target descriptions. In summary, our contributions are as follows: (1) Our representation network, RepNet, implements a neural architecture that explicitly enforces the separation of static and dynamic features via a set-based β-VAE model [23] equipped with a Latent ODE [49]. (2) Our translation network, TraNet, follows an encoder-decoder architecture which is guided by the representation network through a novel multi-feature modulation block called MFMOD where the residual activation maps are modulated based on both the given textual description and the disentangled content code. (3) To test the capabilities of our model in a more realistic setting, we collect a new dataset containing *Fashion Videos* with the related textual descriptions.

2 Related Work

Disentangled representations. The aim of unsupervised disentangled representation learning is to discover underlying factors of variation in a training data, in which each dimension encodes a unique and semantically meaningful aspect of the data [4]. To this end, most of the existing approaches are based on VAEs [32, 48] with slight modifications in the VAE objective, such as β-VAE [23], FactorVAE [31]. Similarly, some prior work tweak the objective of GANs [19] to achieve disentanglement, *e.g*. InfoGAN [8], IB-GAN [27]. Locatello *et al*. [37] showed that unsupervised learning of disentangled generative factors can not be achieved without strong inductive biases on both the models and the data, which can be alleviated using weak supervision or a few labeled training samples [54]. Key to the success of our model, in our work we especially focus on disentangling motion and content, which has been previously studied in a fully unsupervised setting [10], or using action/attribute labels [21]. But we instead utilize natural language descriptions as weak supervision.

VAE-GAN hybrids. GANs are superior to VAEs in terms of visual quality, but VAEs provide better disentangled representations. There is a line of research that explores combining VAE and GAN frameworks, ALI [15], BiGAN [13], IntroVAE [24], to name a few. The promise of these so-called hybrid approaches is to combine the advantages of both models, while providing a much stable training and improved diversity in the generated samples.

Video synthesis. Video synthesis aims at generating temporally coherent video clips either from scratch [11, 17, 36, 40, 60, 63], or according to a single image or a short sequence of images [1, 12, 50, 55, 58]. Motivated by these works, there are also some efforts to allow the users to control the video generation process by introducing natural text [3, 35, 39, 43] or spoken language [6, 9, 53, 66, 67] as an additional input.

Language based image manipulation. In text-to-image synthesis, the goal is to generate an image with a natural language description [46, 62, 65, 68]. On the other hand, semantic image manipulation aka language guided image editing models [14, 33, 34, 41] aim at modifying a source image according to a given textual description summarizing the desired object characteristics. SISGAN [14] involves a text-conditioned encoder-decoder architecture. TAGAN [41] learns to disentangle different semantic attributes of the target object during training by considering a text-adaptive discriminator. ManiGAN [33] and LightweightGAN [34], on the other hand, utilize text-image combination modules, which are used to match semantic attributes with certain words in the given descriptions, along with explicit word-level discriminators to improve the quality of the results. The recently proposed TediGAN [61], Latent Transformer [64], and StyleCLIP [42] models are also capable of performing language-driven edits on a given image, but they all require a StyleGAN model pre-trained for a specific domain (*e.g*. faces), which is hard to train for less structured domains like full body images. Recently, Jiang et al. [28] propose a new language-guided editing model specifically designed for performing global edits such as changing brightness or color tone of an image.

Language based video manipulation. Our task of video-editing using natural language descriptions is a relatively new one. There are two studies worth mentioning which are concurrent to this work: (i) [18] tackles a similar problem by proposing a transformer-based architecture, but they did not make their implementation freely available; (ii) [2] presents a StyleGAN3-based video-editing framework, but, it only considers manipulations based on a single attribute. The latter belongs to the family of GAN-inversion based methods which

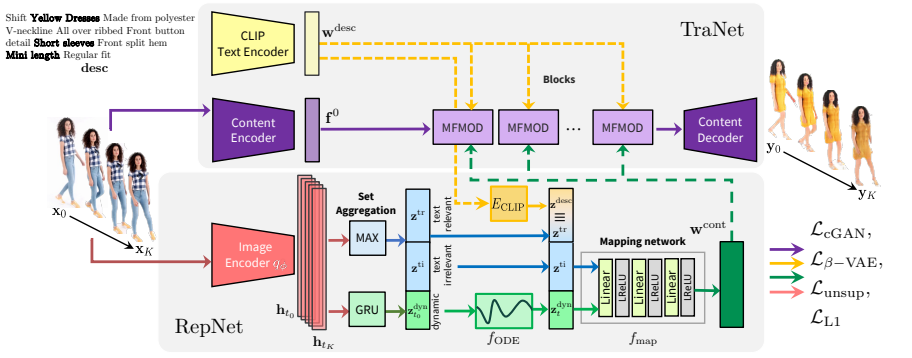

Figure 2: **Schematic illustration of our DiCoMoGAN model.** DiCoMoGAN consists of a representation network (RepNet) and a translation network (TraNet) trained in a harmonious manner. While the former aims to disentangle motion and content by a set-based formulation combining VAEs, latent ODEs and set operations, the latter takes advantage of the extracted latent features to better guide the manipulation by employing a conditional normalization method which we call multi-feature modulation (MFMOD).

uses the latent space of a pretrained StyleGAN model for editing purposes, where the existing methods focus on distinct domains such as aligned faces, as the style-based generators do not work well on unstructured datasets. Note that while inversion constitutes a sensible research direction for GANs, inverting diffusion models without significant distortion remains a challenge. Exploring these directions is future work. Nonetheless, our approach and concept of disentangled video editing can be used regardless of the backbone architectures.

3 DiCoMoGAN

Problem setting. We consider the problem of manipulating a given input video according to a provided textual description. Inputs to our text-based video manipulation approach, called DiCoMoGAN, are a short video clip of a single object and a target text description summarizing the object's new look. We represent the source video as an image sequence denoted by $X = (\mathbf{x}_i \in \mathbb{R}^{3\times H\times W})_{i=1}^{N}$, with i being the frame index and N indicating the total number of frames. Our goal is to perform seamless and semantically meaningful edits on each video frame $\mathbf{x}_i$ to reflect what is being described in the target text **desc**, and accordingly generate an output sequence $Y = (\mathbf{y}_i)_{i=1}^{N}$ of the same spatial dimensions as the input – with the desired look. Our model carries this out by:

1. a **representation network** to learn a disentangled latent space in which static and dynamic semantic scene characteristics are encoded independently,
2. a **translation network** capable of transferring the target look stated in the textual description to the source video in a truthful and temporally coherent manner, and
3. unifying (1) and (2) with a **combined neural architecture** where the two networks are trained simultaneously in an end-to-end manner.

In what follows, we describe DiCoMoGAN in detail, following the structure shown in Fig. 2.

3.1 Network Architecture

Representation network (RepNet). Unlike prior work [14, 41] focusing on language-driven image edits, our aim is to perform edits on short video clips. Thus, in our formulation, capturing the intrinsic characteristics of the scene and the object depicted in the source video plays a key role. As a remedy, we design RepNet for the purpose of extracting a disentangled representation of the input video from the complementary video frames $\{\mathbf{x}_i\}$. To this end, we employ a β-VAE architecture [23] enriched with a Latent ODE [49] to encode an input video X in a latent space. In particular, we split the latent space into two parts as static and dynamic: $\mathbf{z} = \left[\mathbf{z}^{\text{ST}} \;\; \mathbf{z}^{\text{dyn}}\right]$. Static latent codes $\mathbf{z}^{\text{ST}}$ do not change across consecutive frames and encode properties like object color and identity, etc. Dynamic codes $\mathbf{z}^{\text{dyn}}$ are steered by the Latent ODE and encode characteristics that smoothly vary across frames like pose, orientation. Such explicit architecture design coupled with respective loss functions (to be precised later) builds the appropriate inductive bias, encoding distinct features by certain dimensions of the latent space.

The main part of RepNet is an image encoder network q_ϕ including CNN layers, a GRU module and a Neural ODE [7] which is responsible for extracting dynamic latent codes. We assume that RepNet takes a set of frames $\{\mathbf{x}_j\}_{j=0}^{K}$ at times $\{t_j\}_{j=0}^{K}$, where $K(K<N)$ denotes the number of frames irregularly sampled from the input video clip. Its convolutional layers encode each observation individually to a feature map, resulting in the set $\{\mathbf{h}_j\}_{j=t_0}^{t_K}$.

Inspired by [47], we disentangle the motion dynamics from appearance (content). To obtain the *static*, *i.e.*non-time varying latent codes, we first max-pool a subspace of those hidden features to get a permutation invariant representation $\hat{\mathbf{h}}^{\text{ST}}$, which is then mapped to a static latent code $\mathbf{z}^{\text{ST}}$ through a linear layer. Note, $\mathbf{z}^{\text{ST}}$ is shared among all the input video frames and carries the global context. Dynamic codes are obtained by feeding the hidden features in the remaining subspace to a GRU module in reverse order with time gaps $\Delta t = t_i - t_{i-1}$ according to time stamps $(t_K > t_{K-1} > ... > t_0)$. GRU module produces a dynamic hidden feature $\hat{\mathbf{h}}_{t_0}^{\text{dyn}}$ at t_0 using the update rule as given by:

$$\hat{\mathbf{h}}_{t_{i-1}}^{\text{dyn}} = \text{GRU}(\hat{\mathbf{h}}_{t}^{\text{dyn}}, \Delta t, \mathbf{h}_{t_{i-1}}) \,, \tag{1}$$

A linear layer then maps $\hat{\mathbf{h}}_{t_0}^{\text{dyn}}$ to a dynamic latent code $\mathbf{z}_{t_0}^{\text{dyn}}$. Once the dynamic latent code $\mathbf{z}_{t_0}^{\text{dyn}}$ is calculated for t_0, a Neural ODE function f_{ODE} is learned to predict dynamic latent codes $\mathbf{z}_{t}^{\text{dyn}}$ of the input video at all time stamps $t = t_0, t_1, ..., t_K$ using an ODE solver:

$$[\mathbf{z}_{t_0}^{\text{dyn}}, \mathbf{z}_{t_1}^{\text{dyn}}, ... \mathbf{z}_{t_K}^{\text{dyn}}] = \text{ODESolve}(f_{\text{ODE}}, \mathbf{z}_{t_0}^{\text{dyn}}, (t_0, t_1, ..., t_K)) \tag{2}$$

The final latent code $\mathbf{z}_{t_i}$ for an input video at time step t_i is built up by concatenating the calculated static and dynamic latent vectors as $\mathbf{z}_{t_i} = \left[\mathbf{z}^{\text{ST}} \;\; \mathbf{z}_{t_i}^{\text{dyn}}\right]$. From here on, we omit time subscripts whenever possible for notational convenience, *i.e.* $\mathbf{z}$ for $\mathbf{z}_{t_i}$.

Our task requires learning to make local structural changes depending on the input text description like completely replacing an outfit with a new one. As such, text irrelevant regions must be preserved while performing required changes. Hence, we introduce a modified β-VAE to learn to pass only text irrelevant codes to TraNet as condition. To this end, we split $\mathbf{z}^{\text{ST}}$ into *text relevant* $\mathbf{z}^{\text{tr}}$ and *text irrelevant* $\mathbf{z}^{\text{ti}}$ parts. To ensure better disentanglement in the latent space, we jointly let $\mathbf{z}^{\text{tr}}$ live in the space of text features *i.e.* $\mathbf{z}^{desc}$, whose details will be precised in Sec. 3.2. Altogether, we write $\mathbf{z}' = \left[\mathbf{z}^{\text{tr}} \;\; \mathbf{z}^{\text{ti}} \;\; \mathbf{z}^{\text{dyn}}\right]$ representing the video

frame at t_i. This representation aggregates information from multiple frames and is informed about the temporal dynamics. This cue is key in guiding TraNet in manipulating the source frames according to the target text. In what follows, we pass the text irrelevant latent codes $\mathbf{z}^{\text{cont}} = \left[\mathbf{z}^{\text{ti}}\ \mathbf{z}^{\text{dyn}}\right]$ to TraNet as the content condition.

Translation network (TraNet). As argued before, guiding the manipulation process based only on the target text is suboptimal since the textual description usually carries little information about which image regions to keep unchanged. Hence, we design TraNet as an encoder-decoder network with multiple conditioned residual blocks resembling a combination of pix2pixHD [59] and Semi-StyleGAN [42]. The latent motion and static codes extracted from multiple frames help alleviating this by bringing additional conditioning.

TraNet uses a special conditional normalization method, which we call *multi-feature modulation* (MFMOD), that modulates the residual feature activation maps based on both text and content codes derived from RepNet. It learns optimum weights for each of these two conditions to perform feature modulation in an harmonious manner (*c.f.* Fig. 3).

For a batch of N samples, let the activation map before the i^{th} MFMOD block be $\mathbf{f}^i \in \mathbb{R}^{C_i \times H_i \times W_i}$ where C_i is the number of feature map channels and H_i and W_i are the spatial dimensions. The text condition $\mathbf{w}^{\text{desc}} \in \mathbb{R}^{512}$ is obtained by the text encoder of the pretrained CLIP model [45]. For the content latent $\mathbf{z}^{\text{cont}}$, however, we employ a mapping network f_{map} (a shallow subnetwork composed of 4 fully connected layers) to map $\mathbf{z}^{\text{cont}}$ to a higher-dimensional vector $\mathbf{w}^{\text{cont}} \in \mathbb{R}^{256}$. Our proposed normalization scheme can be interpreted as a special AdaIN operation [25] with an adaptive multi-feature blending before the activation feature modulation (*c.f.* Fig. 3). The normalized activation value at site $(n \in N, c \in C_i, y \in H_i, x \in W_i)$ is given by

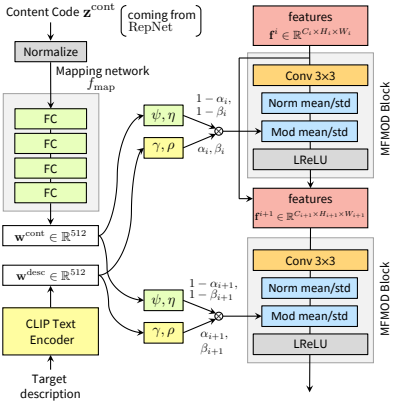

Figure 3: **MFMOD block.** The proposed conditional normalization scheme modulates residual activation maps based on text and content code by learning optimal modulation parameters and blending weights.

$$\left(\alpha_i \gamma^i_{c,y,x}(\mathbf{w}_{\text{desc}}) + (1-\alpha_i)\psi^i_{c,y,x}(\mathbf{w}_{\text{cont}})\right)\frac{h^i_{n,c,y,x} - \mu^i_c}{\sigma^i_c} + \beta_i \rho^i_{c,y,x}(\mathbf{w}_{\text{desc}}) + (1-\beta_i)\eta^i_{c,y,x}(\mathbf{w}_{\text{cont}}) \quad (3)$$

where $f^i_{n,c,y,x}$ is the preactivation, μ^i_c, σ^i_c are the mean and the standard deviation of the activations in the channel c given by:

$$\mu^i_c = \frac{1}{NH^iW^i}\sum_{n,y,x} f^i_{n,c,y,x}, \qquad \sigma^i_c = \sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}\left(\left(f^i_{n,c,y,x}\right)^2 - \left(\mu^i_c\right)^2\right)},$$

with $\gamma^i_{c,y,x}$, $\rho^i_{c,y,x}$, $\psi^i_{c,y,x}$, $\eta^i_{c,y,x}$ respectively denoting the learned modulation parameters for the description and content conditions $\mathbf{w}_{\text{desc}}$ and $\mathbf{w}_{\text{cont}}$. Note that the blending values α_i and β_i are not fixed, but learned during the training phase.

After multi-conditional residual blocks, the last (conditioned) feature map is fed to the decoder, which consists of several convolutional transpose layers to upsample it to the original resolution to obtain the manipulated frame $\mathbf{y}$. In the decoder, we also apply instance normalization in all convolutional transpose layers except the last layer. We use ReLU activation in all convolutional and convolutional transpose layers in all parts of the network. Please refer to the supplementary material for details.

3.2 Training

Our representation and translation networks, RepNet and TraNet, are trained jointly in an end to end manner, by minimizing a non-convex, multi-task loss:

$$\mathcal{L} = \mathcal{L}_{\text{RepNet}} + \lambda_T \mathcal{L}_{\text{TraNet}}, \qquad \mathcal{L}_{\text{RepNet}} = (\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{rec}'})/2 - \beta(\mathcal{L}_{\text{KL}}^{\text{ST}} + \mathcal{L}_{\text{KL}}^{\text{dyn}}) \tag{4}$$
$$\mathcal{L}_{\text{TraNet}} = \min_{\text{TraNet}} \left(\max_{\text{Discr}} \mathcal{L}_{\text{cGAN}} + \lambda_{\text{L1}} \mathcal{L}_{\text{L1}} + \lambda_{\text{U}} \mathcal{L}_{\text{unsup}} \right)$$

where $\lambda_{\text{L1}} = 1, \lambda_{\text{U}} = 0.5, \lambda_T = 1$ are set empirically. We now define each of the loss terms.

Disentanglement losses $\mathcal{L}_{\text{KL}}^{\text{dyn}}$ and $\mathcal{L}_{\text{KL}}^{\text{ST}}$. We enforce the latent code $\mathbf{z}$ to disentangle latent factors of variation. This is measured by computing the KL-divergence individually for static and dynamic distributions:

$$\mathcal{L}_{\text{KL}}^{\text{ST}} = \frac{1}{K} \sum_{t=0}^{K} \mathbb{E}_{q_\phi(\mathbf{z}^{\text{ST}}|\mathbf{x}_t)} D_{KL}(q_\phi(\mathbf{z}^{\text{ST}}|\mathbf{x}_t) \| p(\mathbf{z}^{\text{ST}})), \quad \mathcal{L}_{\text{KL}}^{\text{dyn}} = \mathbb{E}_{q_\phi\left(\mathbf{z}_{t_0}^{\text{dyn}}|\mathbf{x}_{t_0}\right)} D_{KL}\left(q_\phi\left(\mathbf{z}_{t_0}^{\text{dyn}}|\mathbf{x}_{t_0}\right) \| p(\mathbf{z}_{t_0}^{\text{dyn}})\right)$$

Reconstruction losses $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{rec}'}$. During training $\mathbf{z}_{t_i}$ is passed to the image decoder p_θ to reconstruct the input video frame at time t_i from its latent code. This reconstruction loss in our β-VAE objective reads:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right] \tag{5}$$

We also introduce an auxiliary text encoder E_{CLIP}, whose output is aligned with text relevant code $\mathbf{z}^{\text{tr}}$ creating a joint latent space. From the given textual description, text features $\mathbf{w}^{\text{desc}}$ are extracted by first using the text encoder E_{CLIP} of the off-the-shelf CLIP model [45] and then feeding these CLIP embeddings to a series of linear projects to obtain a lower dimensional text representation $\mathbf{z}^{\text{desc}}$ that is the same dimension with that of $\mathbf{z}^{\text{tr}}$. We then define an additional reconstruction loss for learning to specify text relevant part of latent space as:

$$\mathcal{L}_{\text{rec}'} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}') \right] , \tag{6}$$

This extra supervision enforces text relevant subspace of the latent code to be aligned with the CLIP space, improving the disentanglement ability of RepNet. Note that the image decoder p_θ and the auxiliary text encoder E_{CLIP} is not used in inference.

GAN loss $\mathcal{L}_{\text{cGAN}}$. The first cue for training TraNet comes from the conditional adversarial loss. We employ a discriminator network Discr, which resembles the multi-scale PatchGAN discriminator [26, 59], with the only difference being the proposed MFMOD normalization block added after the last conv layer to improve conditioning.

Perceptual loss $\mathcal{L}_{\text{L1}}$. To ensure the quality of the generated images, we employ a *perceptual loss* [29] that minimizes the $L1$ distance between the feature maps of each input frame $\mathbf{x}_i$ and the manipulation result $\mathbf{y}_i$ extracted by a VGG-19 network [52] trained on ImageNet [11].

$$\mathcal{L}_{\text{L1}} = \|\Phi_{VGG}(\mathbf{x}) - \Phi_{VGG}(\mathbf{y})\|_1 \tag{7}$$

Unsupervised loss $\mathcal{L}_{\text{unsup}}$. Finally, to enforce consistency between latent codes of input frames (x) and their manipulated versions (y), we introduce an unsupervised loss defined as the L_2-distance between outputs from the β-VAE encoder of RepNet as:

$$\mathcal{L}_{\text{unsup}} = \|q_\phi(\mathbf{z}|\mathbf{x}) - q_\phi(\mathbf{z}|\mathbf{y})\|_2 \tag{8}$$

We observe that while the contribution of this unsupervised loss to the final quality is only marginal, it helps to stabilize the training process.

Training details. We adopt a learning schedule on the image encoder of RepNet while back propagating the loss from TraNet. This is because at the beginning of training, content code from RepNet is incomplete, which hurts the training of TraNet. In particular, we gradually increase the learning rate from zero to a certain value along certain number of iterations. We provide further details in the supplementary material.

4 Experiments

Datasets. First, we use the **3D Shapes dataset** [5] which is proposed for learning and assessing factors of variation from data. This dataset has 480K images of 64×64 resolution. There are 6 ground truth independent latent factors. They are *floor color*, *wall color*, *object color*, *scale*, *shape* and *orientation*. For our purpose, we build simple text descriptions which covers object related latent factors *object color*, *scale* and *shape*, *e.g*. *"There is a big blue capsule."*. To prevent scale ambiguity, we remove two elements of the scale factor which is of length 8, originally. In that case, "small", "medium" and "big" in the descriptions correspond to the first two, middle two and the last two values, respectively. Moreover, we consider the orientation factor as a dynamic dimension taking 15 different values. We have 19.2K train and 4.8K test videos with 15 frames and simple text descriptions for each video.

Second, to explore how well our model generalizes to more challenging datasets, we collected a new video dataset, **Fashion Videos**, from an online shopping site, containing short video clips of individuals wearing different kinds of garments. Each clip includes full-body images of a single person moving around a scene, showing how the clothing looks from different angles. Moreover, the clip is endowed with a textual product description of the garment, detailing its visual features (color, material properties, and design details) as well as its category (*dress*, *jumpsuits*, *trousers*, *jumper*, *skirt*, *pant*). After pre-processing, we obtained 3178 video clips (109K frames), out of which 2579 are used for training and 598 for testing. More details are provided in the supplementary material and Fashion Videos will be made publicly available.

Evaluation metrics. We evaluate the results via Inception Score (IS)[51], Fréchet Inception Distance (FID) [22], and Fréchet Video Distance (FVD) [56]. Moreover, we modify and use the manipulative precision (MP) metric suggested in [33] to assess the manipulation performance of the models according to the target natural language descriptions. Our version (MP_{CLIP}) measures the similarity between the manipulated video frames and their corresponding target texts through the cosine similarity in CLIP embedding space [45]. Further details on the precise definitions of these metrics are found in our supplementary.

Baselines. As, to the best of our knowledge, the literature lacks a strong language-driven video manipulation model, we compare DiCoMoGAN against SISGAN [14], TAGAN [41] and ManiGAN [33][1]. For video editing, we conduct a frame-by-frame translation.

4.1 Evaluations

Implementation details. For 3D Shapes, we use a 6-dim latent code for the frames in which

[1]We exclude LightweightGAN [34] from our analysis as it requires part-of-speech (POS) tagging, and our analysis revealed that existing POS taggers do not give satisfactory results on domain-specific fashion descriptions.

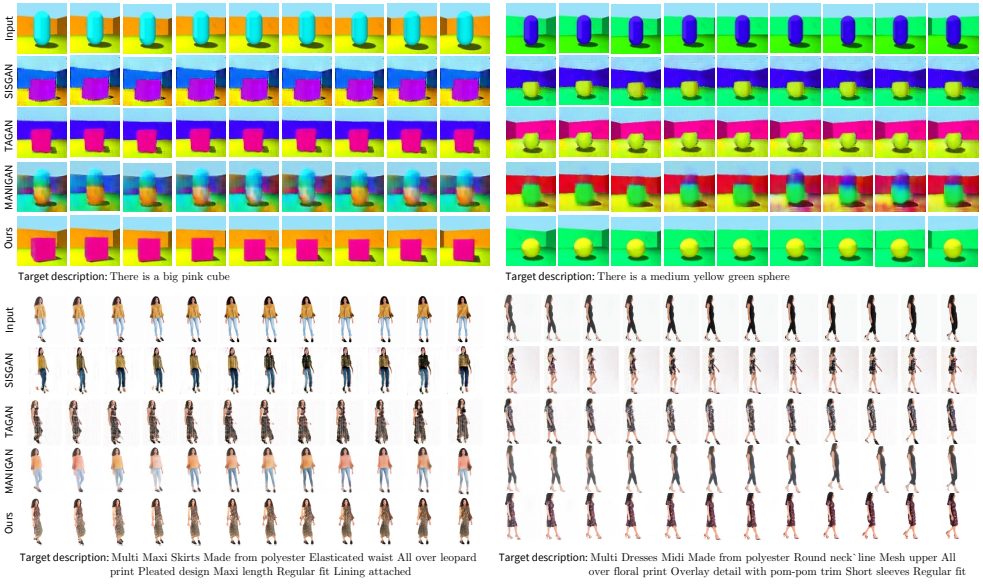

Figure 4: **Qualitative results on 3D Shapes [5] and Fashion Videos datasets**. As compared to SISGAN [14], TAGAN [41] and ManiGAN [33], DiCoMoGAN gives sharper images faithful to the target descriptions while preserving inherent features not mentioned in the text, *e.g.* wall and floor colors, identity, hair style, much better than the competing approaches.

Table 1: **Quantitative results.** Our approach outperforms the existing frame-based methods by a large margin in terms of all evaluation measures.

	Model	IS (↑)	FID (↓)	FVD (↓)	MP_{CLIP} (↑)	Model	IS (↑)	FID (↓)	FVD (↓)	MP_{CLIP} (↑)
3D Shapes	SISGAN [14]	2.29	138.78	1185.48	0.18	Fashion Videos SISGAN [14]	2.13	80.15	2274.69	0.19
	TAGAN [41]	2.34	88.95	974.59	0.19	TAGAN [41]	2.24	87.60	1294.72	0.24
	ManiGAN [33]	2.71	26.90	753.89	0.18	ManiGAN [33]	2.76	37.22	392.59	0.22
	DiCoMoGAN	**2.76**	**9.08**	**69.30**	**0.26**	DiCoMoGAN	**2.96**	**15.34**	**53.75**	**0.25**

the first three encode the text relevant static features, the next two the text-irrelevant static features, and the last one the dynamic feature. We set $\beta = 32$. For Fashion Videos, we do not have access to the ground truth factors of variation. Thus, we consider a 16-dim latent code in which the first eight encode the static text relevant features and the next eight the static text-irrelevant ones. The last four are reserved for the dynamic features. We set $\beta = 1$.

Manipulation results. In Tab. 1, we provide our quantitative analysis on the 3D Shapes and the Fashion datasets. As compared to the state-of-the-art, our method gives the best results in terms of all of the evaluation metrics. In particular, our method achieves much better FVD values on both datasets. The qualitative results in Fig. 4 indicate that our model can produce high quality results as compared to the existing models. SISGAN and TAGAN fail to preserve the text irrelevant parts like the wall color or the identity of the person. ManiGAN tends to keep the original structure intact and fails to produce the necessary structural changes. Our method, on the other hand, performs more relevant edits on the input video sequences according to the target descriptions, altering only the necessary parts of the frames while keeping what is not mentioned in text unchanged. Please refer to the supplementary material for additional higher resolution results. Our main goal is to analyze disentangling factors in

Table 2: **Ablation study.** Analysis of the components of our DiCoMoGAN model.

	Model	IS (↑)	FID (↓)	FVD (↓)	MP_{CLIP} (↑)	Model	IS (↑)	FID (↓)	FVD (↓)	MP_{CLIP} (↑)
3D Shapes	DiCoMoGAN	2.76	9.08	69.30	0.26	DiCoMoGAN	2.96	15.34	53.75	0.25
	w/o Latent ODE	2.91	12.50	99.79	0.26	*w/o* Latent ODE	3.10	5.35	225.58	0.24
	w/o RepNet	2.82	12.32	113.89	0.25	*w/o* RepNet	3.06	13.97	498.21	0.24

videos for text-based manipulation, and there is definitely room for improvement for visual quality. In the supplementary material, we also show that better results can be obtained when we train a local enhancer network on top of TraNet in a similar vein to pix2pixHD [59].

Measuring disentanglement. For disentanglement performance, we carry out experiments on 3D Shapes, where we have ground truth factors of variation, by considering the latent space discovered by RepNet's image encoder. Fig. 5 shows sample traversals in the latent dimensions learned by our method. These traversals clearly depict interpretable properties of the images exist in the 3D Shapes dataset. While the last latent dimension steered by the Latent ODE encodes the orientation of the camera (camera movement), all the others encode static features of the scene and the object. In fact, while the first three static dims. are about text relevant features like object color, shape and size of the object, the last two encode wall and floor colors – successfully identified without even referred in the provided textual descriptions. We provide further quantitative analysis in the supplementary.

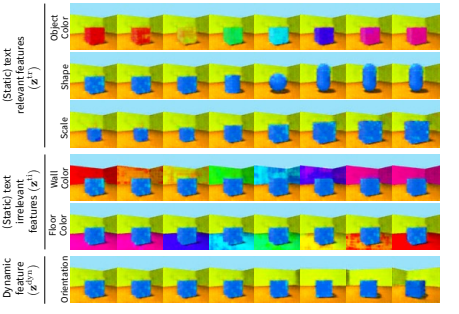

Figure 5: **Latent traversals.** DiCoMoGAN learns latent variables depicting highly interpretable concepts decomposed into text relevant, text irrelevant static, and dynamic features. Note that wall and floor colors are not mentioned in the descriptions during training.

Ablation study In Tab. 2, we show the results of our ablation study in which we examine the contribution of certain components of our method on the performance. Latent ODE within RepNet plays a key role in achieving high-quality manipulation results, which can be attributed to its ability of effectively disentangling motion and content. In the supplementary material, we also analyze the effect of different loss functions and the MFMOD block.

5 Conclusion and Future Work

We presented DiCoMoGAN to tackle the challenging task of manipulation of videos using textual descriptions. As a first step towards solving this problem, we developed a new neural model that incorporates multiple observations to disentangle motion dynamics and visual content to better perform semantically relevant and temporally coherent edits. Our approach gives significantly better results than existing frame-based methods. As such, there are also a number of ways this work could be extended. It is possible to explore more complicated feature aggregation schemes like self-attention [57]) to learn permutation invariant representations. Our current model assumes that input video clips include a single object of interest. An exciting future research direction is to incorporate an object-centric approach [16, 20, 38] so that it can support manipulation of multiple objects.

References

[1] Adil Kaan Akan, Erkut Erdem, Aykut Erdem, and Fatma Güney. SLAMP: Stochastic latent appearance and motion prediction. In *ICCV*, 2021.

[2] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. *arXiv preprint arXiv:2201.13433*, 2023.

[3] Yogesh Balaji, Martin Renqiang Min, Bing Bai, Rama Chellappa, and Hans Peter Graf. Conditional GAN with discriminative filter generation for text-to-video synthesis. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *PAMI*, 35(8):1798–1828, 2013.

[5] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[6] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *CVPR*, 2021.

[7] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

[9] Dipanjan Das, Sandika Biswas, Sanjana Sinha, and Brojeshwar Bhowmick. Speech-driven facial animation using cascaded GANs for learning of motion and texture. In *ECCV*, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[11] E. L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 2017.

[12] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018.

[13] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2016.

[14] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *ICCV*, 2017.

[15] Vincent Dumoulin, Mohamed Ishmael Diwan Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. In *ICLR*, 2017.

[16] Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In *ICML*, 2021.

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

[18] Tsu-Jui Fu, Xin Eric Wang, Scott T. Grafton, Miguel P. Eckstein, and William Yang Wang. M3L: Language-based video editing via multi-modal multi-level transformers. In *CVPR*, 2023.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[20] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.

[21] Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[23] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[24] Huaibo Huang, zhihang li, Ran He, Zhenan Sun, and Tieniu Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. In *NeurIPS*, pages 52–63, 2018.

[25] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[27] Insu Jeon, Wonkwang Lee, Myeongjang Pyeon, and Gunhee Kim. Ib-gan: Disentangled representation learning with information bottleneck generative adversarial networks. In *AAAI*, 2021.

[28] Wentao Jiang, Ning Xu, Jiayun Wang, Chen Gao, Jing Shi, Zhe Lin, and Si Liu. Language-guided global image editing via cross-modal cyclic mechanism. In *ICCV*, 2021.

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[31] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[33] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. ManiGAN: Text-guided image manipulation. In *CVPR*, 2020.

[34] Bowen Li, Xiaojuan Qi, Philip H. S. Torr, and Thomas Lukasiewicz. Lightweight generative adversarial networks for text-guided image manipulation. In *NeurIPS*, 2020.

[35] Yitong Li, Martin Renqiang Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *AAAI*, 2018.

[36] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion GAN for future-flow embedded video prediction. In *NeurIPS*, 2017.

[37] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019.

[38] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

[39] Tanya Marwah, Gaurav Mittal, and Vineeth N. Balasubramanian. Attentive semantic video generation using captions. In *CVPR*, 2017.

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

[41] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *NeurIPS*, pages 42–51, 2018.

[42] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. Semi-supervised StyleGAN for disentanglement learning. In *International Conference on Machine Learning*, pages 7360–7369, 2020.

[43] Yingwei Pan, Zhaofan Qiu, Ting Yao Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. In *ACMMM*, 2017.

[44] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[46] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069, 2016.

[47] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J. Guibas. Caspr: Learning canonical spatiotemporal point cloud representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[48] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[49] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *NeurIPS*, 32:5320–5330, 2019.

[50] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017.

[51] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[52] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[53] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, 2020.

[54] Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *ICML*, 2021.

[55] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018.

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

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[58] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016.

[59] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *CVPR*, pages 1–13, 2018.

[60] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkorei. Scaling autoregressive video models. In *ICLR*, 2020.

[61] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. TediGAN: Text-guided diverse face image generation and manipulation. In *CVPR*, 2021.

[62] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, pages 1316–1324, 2018.

[63] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, 2016.

[64] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *ICCV*, 2021.

[65] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5908–5916, 2017.

[66] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *CVPR*, 2019.

[67] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. MakeltTalk: Speaker-aware talking-head animation. *ACM Trans. Graph.*, 39(6), nov 2020.

[68] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis. In *CVPR*, 2019.