

G³AN: Disentangling Appearance and Motion for Video Generation

Yaohui Wang^{1,2} Piotr Bilinski³ Francois Bremond^{1,2} Antitza Dantcheva^{1,2}
¹Inria ² Université Côte d’Azur ³University of Warsaw

{yaohui.wang, francois.bremond, antitza.dantcheva}@inria.fr bilinski@mimuw.edu.pl

Abstract

Creating realistic human videos entails the challenge of being able to simultaneously generate both appearance, as well as motion. To tackle this challenge, we introduce G³AN, a novel spatio-temporal generative model, which seeks to capture the distribution of high dimensional video data and to model appearance and motion in disentangled manner. The latter is achieved by decomposing appearance and motion in a three-stream Generator, where the main stream aims to model spatio-temporal consistency, whereas the two auxiliary streams augment the main stream with multi-scale appearance and motion features, respectively. An extensive quantitative and qualitative analysis shows that our model systematically and significantly outperforms state-of-the-art methods on the facial expression datasets MUG and UvA-NEMO, as well as the Weizmann and UCF101 datasets on human action. Additional analysis on the learned latent representations confirms the successful decomposition of appearance and motion. Source code and pre-trained models are publicly available ¹.

1. Introduction

Generative Adversarial Networks (GANs) [10] have witnessed increasing attention due to their ability to model complex data distributions, which allows them to generate realistic images [5, 17, 18, 21, 24, 25, 41, 44], as well as to translate images [2, 15, 29, 32]. While realistic video generation is the natural sequel, it is substantially more challenging w.r.t. complexity and computation, associated to the simultaneous modeling of appearance, as well as motion.

Specifically, in inferring and modeling the distribution of human videos, generative models face three main challenges: (a) generating uncertain motion, (b) retaining of human appearance throughout the generated video, as well as (c) modeling spatio-temporal consistency. Such challenges have been alleviated by conditioning the generation on potent priors such as input images [42], human keypoints [7] and optical flows [23]. This relates to learning to sample from conditional distributions, assuming access

to the marginal distributions instead of learning joint distributions [27].

Deviating from such approaches, in this work we focus on the highly intricate problem of video generation without prior knowledge w.r.t. either appearance or motion. Specifically, based on noise variables, we generate an appearance, e.g. human face and body, which we concurrently animate, by a facial expression or human action.

G³AN, our new generative model, is streamlined to learn a disentangled representation of the video generative factors appearance and motion, allowing for manipulation of both. A disentangled representation has been defined as one, where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors [4]. In this context, our G³AN is endowed with a three-stream Generator-architecture, where the main stream encodes spatio-temporal video representation, augmented by two auxiliary streams, representing the independent generative factors appearance and motion. A self-attention mechanism targeted towards high level feature maps ensures satisfactory video quality.

G³AN is hence able to generate realistic videos (tackling challenges (a) and (c)) by following a training distribution and without additional input, as well as is able to manipulate the appearance and motion disjointly, while placing emphasis on preserving appearance (challenge (b)).

In summary, our **main technical contributions** include the following.

- A novel generative model, G³AN, which seeks to learn disentangled representations of the generative factors appearance and motion from human video data. The representations allow for individual manipulation of both factors.
- A novel three-stream generator, which takes into account the learning of individual appearance features (spatial stream), motion features (temporal stream) and smoothing generated videos (main stream) at the same time.
- A novel factorized spatio-temporal self-attention (F-SA), which is considered as the first self-attention module applied to video generation, in order to model global spatio-temporal representations and improve the quality of generated videos.

¹<https://wyhsirius.github.io/G3AN/>

- Extensive qualitative and quantitative evaluation, which demonstrates that G³AN systematically and significantly outperforms state-of-the-art baselines on a set of datasets.

2. Related Work

Despite the impressive progress of image generation, the extension to *video* generation is surprisingly challenging. While videos constitute sequences of temporally coherent images, video generation encompasses a majority of challenges that have to do with generation of plausible and realistic appearance, coherent and realistic motion, as well as spatio-temporal consistency. A further challenge, namely the generation of uncertain local or global motion, associated to future uncertainty, allows for multiple correct, equally probable next frames [37]. Finding suitable representation learning methods, which are able to address these challenges is critical. Existing methods include approaches based on Variational Autoencoders (VAEs) [20], auto-regressive models, as well as most prominently Generative Adversarial Networks (GANs) [10].

While video generation tasks aim at generating realistic temporal dynamics, such tasks vary with the *level of conditioning*. We have video generation based on additional priors related to motion or appearance, as well as contrarily, video generation following merely the training distribution. We note that the latter is more challenging from a modeling perspective, due to lack of additional input concerning *e.g.* structure of the generated video. Therefore the majority of approaches to date include a conditioning of some kind.

Video generation with additional input. Due to challenges in modeling of high dimensional video data, additional information such as semantic maps [26, 39], human keypoints [16, 42, 38, 7], 3D face mesh [45] and optical flow [23] can be instrumental as guidance for appearance and motion generation. This additional information is either pre-computed throughout the generated video [16, 45, 7] or predicted based on an initial input image [42]. The additional information guides conditional image translation, which though results in lack of modeling of spatio-temporal correlations.

Video generation from noise. Directly generating videos from noise requires the capturing and modeling of a dataset distribution. Existing works tend to reduce related complexity by decomposing either the output [36] or latent representation [30, 35]. VGAN [36] was equipped with a two-stream spatio-temporal Generator, generating foreground and background separately. TGAN [30] decomposed the latent representation of each frame into a *slow part* and a *fast part*. Due to jointly modeling appearance and motion, generated results from VGAN and TGAN might comprise spatially unrealistic artefacts, see Figure 5. The closest work to ours is MoCoGAN [35], which decomposed the latent representation of *each frame* into motion and content, aiming at controlling both factors. However,

there are two crucial differences between MoCoGAN and G³AN. Firstly, instead of only sampling two noise vectors for each video, MoCoGAN sampled a sequence of noise vectors as motion and a fixed noise as content. However, involving random noise for each frame to represent motion increases the learning difficulty, since the model has to map these noise vectors to a consecutive human movement in the generated videos. As a result, MoCoGAN gradually ignores the input noise and tends to produce a similar motion, as we illustrate in Figure 8. Secondly, MoCoGAN incorporated a simple image Generator aiming at generating each frame sequentially, after which content and motion features were jointly generated. This leads to *incomplete disentanglement* of motion and content. Deviating from that, we design a novel Generator architecture, able to entirely decompose appearance and motion in both, latent and feature spaces. We show that such design generates realistic videos of good quality and ensures factor disentanglement.

Disentangled representation learning. Learning disentangled representations of data has been beneficial in a large variety of tasks and domains [4]. Disentangling a number of factors in *still images* has been widely explored in recent works [8, 24, 33, 22]. In the context of *video generation*, an early approach for motion and appearance decomposition was incorporated in MoCoGAN. However, experiments, which we present later (see Figure 6), suggest that the results are not satisfactory.

3. Approach

In this work, we propose G³AN, a novel GAN architecture, aiming at generating videos in a disentangled manner from two noise vectors, $z_a \in Z_A$ and $z_m \in Z_M$, which represent appearance and motion, respectively. G³AN consists of a three-stream Generator G , as well as a two-stream Discriminator D , as illustrated in Figure 1. While G aims at generating videos with the ability to modulate appearance and motion disjointly, D accounts for distinguishing generated samples from real data, in both, videos and frames, respectively.

3.1. Generator

Hierarchical Generator with G³-modules. We design G in a hierarchical structure of G³ modules. Specifically, we have N levels of hierarchy, denoted as $G_{n=0 \dots N-1}^3$. The first G³ module, G_0^3 accepts as input the two noise vectors z_a and z_m . The remaining modules $G_{n=1 \dots N-1}^3$, inherit the three feature maps $F_{S_{n-1}}$, $F_{V_{n-1}}$ and $F_{T_{n-1}}$ as their inputs from each previous G_{n-1}^3 module, see Figures 1 and 2.

Each G_n^3 module consists of three parallel streams: a spatial stream G_{S_n} , a temporal stream G_{T_n} , as well as a video stream G_{V_n} (Figures 1 and 2). They are designed to generate three different types of features. The spatial stream G_{S_n} , denoted by a blue line in Figures 1 and 2, takes as input z_a for $n = 0$ and $F_{S_{n-1}}$ for $n > 1$, and generates 2D appearance features F_{S_n} by upsampling input features with a

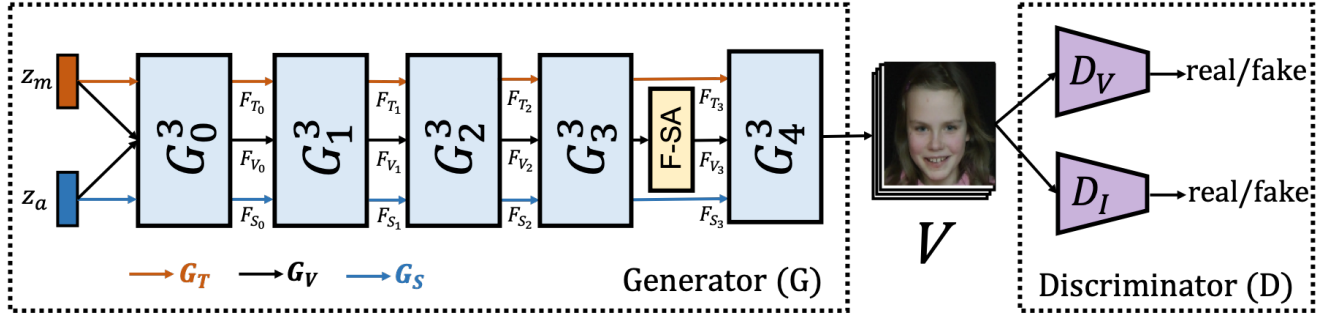


Figure 1: **Overview of our G^3 AN architecture.** G^3 AN consists of a three-stream Generator and a two-stream Discriminator. The Generator contains five stacked G^3 modules, a factorized self-attention (F-SA) module, and takes as input two random noise vectors, z_a and z_m , aiming at representing appearance and motion, respectively. Details of architecture can be found in Supplementary Material (SM).

transposed 2D convolutional layer. These features evolve in spatial dimension and are shared at all time instances. The temporal stream G_{T_n} , denoted by an orange line, accepts as input z_m for $n = 0$ and $F_{T_{n-1}}$ for $n > 1$, and seeks to generate 1D motion features F_{T_n} by upsampling input features with a transposed 1D convolutional layer. These features evolve in temporal dimension and contain global information of each time step. Then, the video stream G_{V_n} , denoted by a black line, takes as input the concatenation of z_a and z_m for $n = 0$ and $F_{V_{n-1}}$ for $n > 1$. It models spatio-temporal consistency and produces 3D joint embeddings $F_{V'_n}$ by upsampling input features with a factorized transposed spatio-temporal convolution, see below. Then, F_{S_n} and F_{T_n} are catapulted to the spatio-temporal fusion block, where they are fused with $F_{V'_n}$, resulting in F_{V_n} . Finally, F_{S_n} , F_{T_n} and F_{V_n} serve as inputs of the next hierarchy-layer G^3_{n+1} .

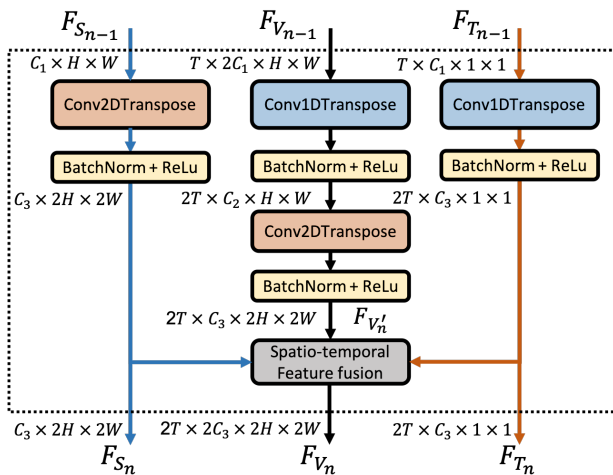


Figure 2: G^3 module architecture.

Factorized transposed spatio-temporal convolution has been proposed by Wang *et al.* in [40]. It explicitly factorizes transposed 3D convolution into two separate and successive operations, M transposed 1D temporal convolution followed by a 2D separate spatial convolution, which is

referred to as transposed (1+2)D convolution. Such decomposition brings an additional nonlinear activation between these two operations and facilitates optimization. Crucially, factorizing transposed 3D convolution yields significant gains in video quality, see Section 4.

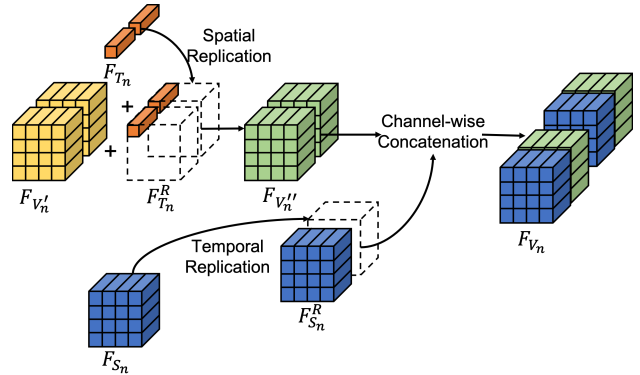


Figure 3: **Spatio-temporal fusion.**

Spatio-temporal fusion is the key-element to learn well disentangled features, the inputs of which are output feature maps F_{S_n} , F_{T_n} and $F_{V'_n}$ from the convolutional layers in each G^3_n module. The fusion contains three steps (see Figure 3). Firstly, spatial and temporal replications are applied on F_{T_n} and F_{S_n} respectively, in order to obtain two new feature maps $F_{T_n}^R$ and $F_{S_n}^R$. Both new feature maps have the same spatio-temporal size as $F_{V'_n}$. Next, $F_{T_n}^R$ and $F_{V'_n}$ are combined through a position-wise addition, creating a new spatio-temporal embedding $F_{V''_n}$. Finally, $F_{S_n}^R$ is channel-wise concatenated with $F_{V''_n}$, obtaining the final fused feature map F_{V_n} . The feature maps F_{S_n} , F_{T_n} and F_{V_n} represent inputs for the following G^3_{n+1} module.

Factorized spatio-temporal Self-Attention (F-SA). Despite Self-Attention (SA) has been successfully applied in image generation within SAGAN [43], however it has not been explored in the context of spatio-temporal video generation yet. Here, we incorporate a spatio-temporal SA module, enabling G to utilize cues from all spatio-temporal feature positions and model relationships between widely sep-

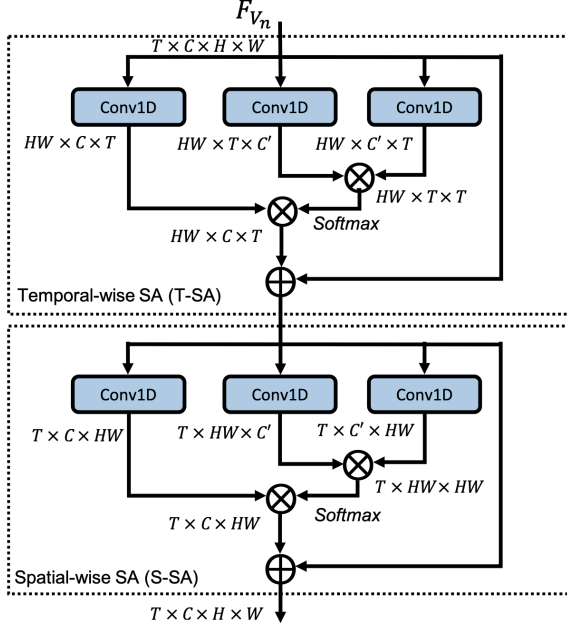


Figure 4: **Factorized spatio-temporal Self-Attention (F-SA).**

arated regions. However, computing correlation between each position with all the others in a 3D spatio-temporal feature map is very computationally expensive, particularly if it is applied on higher feature maps in G . Therefore, we propose a novel *factorized spatio-temporal self-attention*, namely F-SA, as shown in Figure 4. F-SA consists of a Temporal-wise SA (T-SA), followed by a Spatial-wise SA (S-SA). Such factorization reduces the computational complexity, allowing for application of the F-SA on larger feature maps.

In our G^3AN , we apply F-SA on the output of the G_3^3 in the G_V stream, which achieve best video quality. We report related evaluation results of applying F-SA at various hierarchy-layers of the G^3AN in Section 4.

3.2. Discriminator

Towards improving both video and frame quality, similar to MoCoGAN, we use a two-stream *Discriminator* architecture, containing a video stream D_V and an image stream D_I . During training, D_V accepts a full video as input, whereas D_I takes randomly sampled frames from videos.

3.3. Training

Given our two-stream Discriminator architecture, G^3AN simultaneously optimizes D_V and D_I . Both losses use the GAN loss function proposed in DCGAN [28]. The objective functions of G^3AN can be expressed as

$$G^* = \min_G \max_{D_I, D_V} L(G, D_I, D_V), \quad (1)$$

$$L(G, D_I, D_V) = L_I(G, D_I) + L_V(G, D_V), \quad (2)$$

where L_I denotes the loss function related to D_I , L_V represents the loss function related to D_V .

$$L_I = \mathbb{E}_{x' \sim p_{data}} [\log(D_I(x'))] + \mathbb{E}_{z_a \sim p_{z_a}, z_m \sim p_{z_m}} [\log(1 - D_I(G(z_a, z_m))')], \quad (3)$$

$$L_V = \mathbb{E}_{x \sim p_{data}} [\log(D_V(x))] + \mathbb{E}_{z_a \sim p_{z_a}, z_m \sim p_{z_m}} [\log(1 - D_V(G(z_a, z_m))')], \quad (4)$$

G attempts to generate videos from z_a and z_m , while D_I and D_V aim to distinguish between generated samples and real samples. $(\cdot)'$ characterizes that T frames are being sampled from real and generated videos.

4. Experiments

4.1. Implementation details

The entire network is implemented using PyTorch. We employ ADAM optimizer [19] with $\beta_1=0.5$ and $\beta_2=0.999$. Learning rate is set to $2e^{-4}$ for both G and D . Dimensions of latent representations constitute 128 for z_a and 10 for z_m . We set $N = 5$ in order to generate videos of 16 frames with spatial scale 64×64 . We randomly sample $T = 1$ frame from each video as input of D_I . More implementation details can be found in SM.

4.2. Datasets

We evaluate our method on following four datasets.

Facial expression datasets. The **MUG** Facial Expression dataset [1] contains 1254 videos of 86 subjects, performing 6 facial expressions, namely *happy*, *sad*, *surprise*, *anger*, *disgust* and *fear*. The **UvA-NEMO** Smile dataset [9] comprises 1240 video sequences of 400 smiling individuals, with 1 or 2 videos per subject. We crop faces in each frame based on detected landmarks using [6]².

Action recognition datasets. The **Weizmann** Action dataset [11] consists of videos of 9 subjects, performing 10 actions such as *wave* and *bend*. We augment it by horizontally flipping the existing videos. The **UCF101** dataset [34] contains 13,320 videos of 101 human action classes. Similarly to TGAN [30], we scale each frame to 85×64 and crop the central 64×64 regions.

In all our experiments, we sample video frames with a random time step ranging between 1 and 4 for data augmentation.

4.3. Experimental Results

We test our method both quantitatively and qualitatively, providing results of four experiments. Specifically, firstly we evaluate and compare videos generated from G^3AN , VGAN, TGAN and MoCoGAN, quantitatively and qualitatively on all four datasets. Next, we test *conditional* and *unconditional* video generation, where we aim to demonstrate the effectiveness of the proposed decomposition method.

²<https://github.com/ladrianb/face-alignment>

Then, we manipulate the latent representation, providing insight into each dimension of the two representations. We proceed to add appearance vectors and study the latent representation. Finally, we conduct an ablation study, verifying the effectiveness of our proposed architecture.

4.3.1 Quantitative Evaluation

We compare G³AN with three state-of-the-art methods, namely VGAN, TGAN, as well as MoCoGAN. We report two evaluation metrics on the above four datasets. In particular, we use the extension of two most commonly used metrics in image generation, the Inception Score (IS) [31] and Fréchet Inception Distance (FID) [13], in video level by using a pre-trained 3D CNN [12] as feature extractor, similar to Wang *et al.* [39].

The video **FID** grasps both visual quality and temporal consistency of generated videos. It is calculated as $\|\mu - \tilde{\mu}\|^2 + Tr(\Sigma + \tilde{\Sigma} - 2\sqrt{\Sigma\tilde{\Sigma}})$, where μ and Σ represent the mean and covariance matrix, computed from real feature vectors, respectively, and $\tilde{\mu}$, and $\tilde{\Sigma}$ are computed from generated data. Lower FID scores indicate a superior quality of generated videos.

The video **IS** captures the quality and diversity of generated videos. It is calculated as $\exp(\mathbb{E}_{x \sim p_g} D_{KL}(p(y|x)||p(y)))$, where $p(y|x)$ and $p(y)$ denote conditional class distribution and marginal class distribution, respectively. A higher IS indicates better model performance.

We report FID on MUG, UVA-Nemo and Weizmann datasets, and both FID and IS on UCF101. Since IS can only be reported, when GAN and feature extractor are trained on the same dataset, we do not report it on other datasets.

Comparison results among different methods are presented in Table 1. Our method consistently achieves the lowest FID on all datasets, suggesting that videos generated by G³AN entail both, best temporal consistency and visual quality. At the same time, the obtained highest IS on UCF101 indicates that our method is able to provide the most diverse samples among all compared methods. Such evaluation results show that proposed decomposition method allows for controlling the generated samples, and additionally facilitates the spatio-temporal learning of generating better quality videos. Generated samples are illustrated in Figure 5.

| | MUG | UvA | Weizmann | UCF101 | |
|-------------------|--------------|---------------|--------------|--------------|-------------|
| | FID ↓ | FID ↓ | FID ↓ | FID ↓ | IS ↑ |
| VGAN | 160.76 | 235.01 | 158.04 | 115.06 | 2.94 |
| TGAN | 97.07 | 216.41 | 99.85 | 110.58 | 2.74 |
| MoCoGAN | 87.11 | 197.32 | 92.18 | 104.14 | 3.06 |
| G ³ AN | 67.12 | 119.22 | 86.01 | 91.21 | 3.62 |

Table 1: Comparison with the state-of-the-art on four datasets w.r.t. FID and IS.

In addition, we conduct a subjective analysis, where we asked 27 human raters to pairwise compare videos of pertaining to the same expression/action, displayed side by side. Raters selected one video per video-pair. We randomized the order of displayed pairs. We had an equal amount of pairs for each studied case (e.g. G³AN / Real videos). The posed question was "Which video clip is more **realistic**?". We report the mean user preference in Table 2.

We observe that human raters express a strong preference for the proposed framework G³AN over MoCoGAN (84.26% vs. 15.74%), TGAN (87.31% vs. 12.69%) and VGAN (90.24% vs. 9.76%), which is consistent with the above listed quantitative results. Further, we compare real videos from all datasets with the generated video sequences from our method. The human raters ranked 25.71% of videos from our G³AN as more realistic than real videos, which we find highly encouraging.

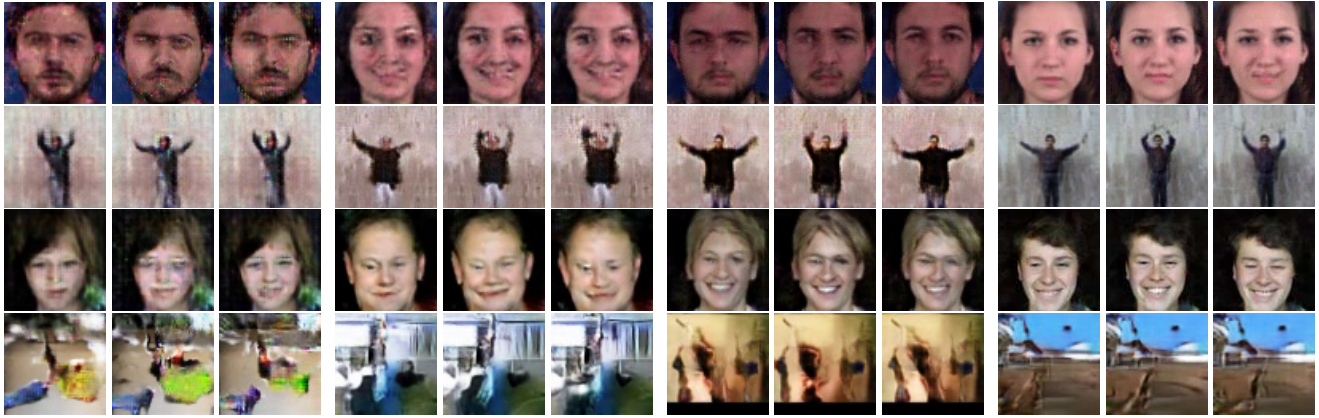
| Methods | Rater preference (%) |
|---------------------------------|----------------------|
| G ³ AN / MoCoGAN | 84.26 / 15.74 |
| G ³ AN / TGAN | 87.31 / 12.69 |
| G ³ AN / VGAN | 90.24 / 9.76 |
| G ³ AN / Real videos | 25.71 / 74.29 |

Table 2: Mean user preference of human raters comparing videos generated by the respective algorithms, originated from all datasets.

4.3.2 Qualitative Evaluation

We conduct an **unconditional generation** experiment utilizing the Uva-NEMO dataset, where we fix z_a and proceed to randomly vary motion, z_m . Associated generated samples from G³AN and MoCoGAN are shown in Figure 6. While we observe the varying motion in the video sequences generated by G³AN, the appearance remains coherent. Hence, our model is able to successfully preserve facial appearance, while *altering* the motion. Therefore, this suggests that our three-stream design allows for manipulation of appearance and motion separately. On the contrary, video sequences generated by MoCoGAN experience constant motion, despite of altering z_m .

Further, we leverage on labels of the MUG and Weizmann datasets, in order to analyze **conditional video generation**. Towards this, we here concatenate a one-hot category vector and motion noise z_m , feeding it into G_T . We note that the inputs of G_S and G_V remain the same as in the setting of unconditional generation. Related results show that when varying motion-categories, while having a fixed appearance, G³AN correctly generates an identical facial appearance, with appropriate category-based motion (facial expressions and body actions), see Figure 7. Further, we note that appearance is very well preserved in different videos and is not affected by category-alterations. In addition, in the same conditional setting, we note that when varying the noise z_m , G³AN is able to generate *the same*



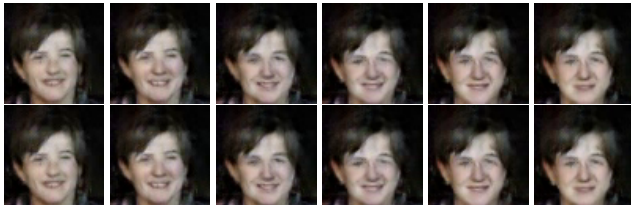
(a) VGAN

(b) TGAN

(c) MoCoGAN

(d) G³AN

Figure 5: **Comparison with the state-of-the-art** on MUG (top-left), Weizmann (top-right), UvA-NEMO (bottom-left) and UCF101 (bottom-right). More samples are presented in SM.

(a) G³AN

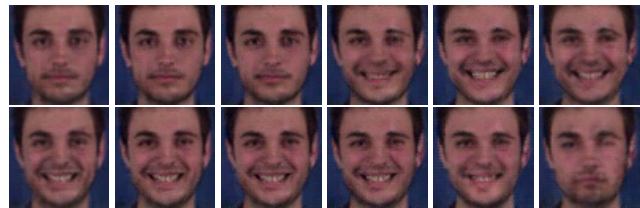
(b) MoCoGAN

Figure 6: **Unconditional video generation** of G³AN and MoCoGAN on Uva-Nemo. For each model, we fix z_a , while testing two z_m instances (top and bottom lines). See SM for more samples.

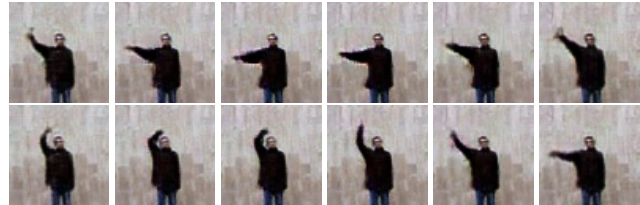
category-motion in different ways. This indicates that z_m enables an intra-class diversity.

In videos generated by MoCoGAN, we observe a correctly generated motion according to given categories, however we note that the category also affects the appearance. In other words, MoCoGAN lacks a complete disentanglement of appearance and motion in the latent representation, see Figure 8. This might be due to a simple motion and content decomposition in the latent space, which after a set of convolutions can be totally ignored in deeper layers. It is notable that G³AN effectively prevents such cases, ensured by our decomposition that occurs in both, latent and feature spaces.

Latent representation manipulation. While there is currently no general method for quantifying the degree of learnt disentanglement [14], we proceed to illustrate the ability of our model to learn latent representations by ma-



(a) MUG: Happiness

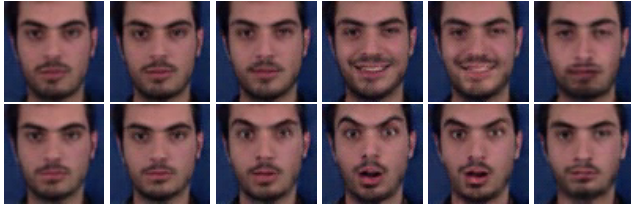


(b) Weizmann: One hand waving

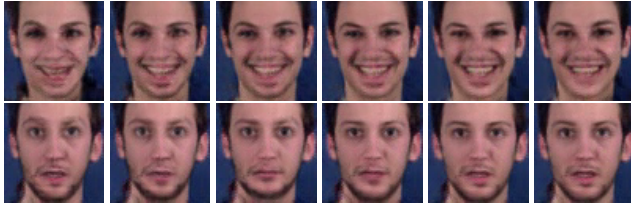
Figure 7: **Conditional video generation** on MUG and Weizmann. For both datasets, each line is generated with random z_m . We observe that same category (*smile* and *one hand waving*) is performed in a different manner, which indicates that our method is able to produce *intra-class* generation. See SM for more samples.

nipulating each dimension in the appearance representation. We show that by changing *values* of different dimensions in the *appearance representation*, we are able to cause a modification of specific appearance factors, see Figure 9. Interestingly such factors can be related to semantics, *e.g.* facial view point in Figure 9a, mustache in Figure 9b, and color of pants in Figure 9c. We note that motion is not affected by altering the appearance representation. Similarly, when altering *values* of different dimensions in the *motion representation*, we observe that factors such as starting position, motion intensity and moving trajectory are being affected, see Figure 10. Such observations show that our method learns to interpolate between different data points in motion- and appearance-latent spaces, respectively.

Addition of appearance representations. We here *add* appearance vectors, aiming to analyze the resulting latent



(a) G^3AN



(b) MoCoGAN

Figure 8: **Comparison between G^3AN and MoCoGAN.** Given fixed z_a and z_m , as well as two condition-labels *smile* and *surprise*, G^3AN and MoCoGAN generate correct facial expressions. However, while G^3AN preserves the appearance between rows, MoCoGAN alters the subject’s appearance.



(a) Manipulation of *third* dimension on UvA-NEMO



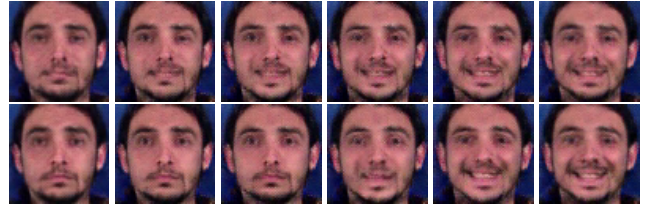
(b) Manipulation of *third* dimension on MUG



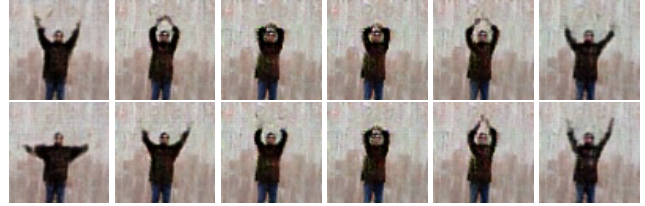
(c) Manipulation of *second* dimension on Weizmann

Figure 9: **Latent appearance representation manipulation.** For each dataset, each row shares the same motion representation, whereas from top to bottom values in one dimension of appearance representation are increased. See SM for more samples.

representations. Towards this, we generate two videos V_a and V_b by randomly sampling two sets of noises, (z_{a_0}, z_{m_0}) and (z_{a_1}, z_{m_1}) . Next, we add z_{a_0} and z_{a_1} , obtaining a new appearance z_{a_2} . When combining (z_{a_2}, z_{m_0}) and (z_{a_2}, z_{m_1}) , we observe in the two new resulting videos a *sum-*



(a) Manipulation of *sixth* dimension of MUG



(b) Manipulation of *second* dimension on Weizmann

Figure 10: **Latent motion representation manipulation.** For each dataset, each row shares the same appearance representation, whereas from top to bottom values in one dimension of the motion representation are increased. See SM for more results.



(a) z_{a_0}, z_{m_0} (top) and z_{a_2}, z_{m_0} (bottom)



(b) z_{a_1}, z_{m_1} (top) and z_{a_2}, z_{m_1} (bottom)

Figure 11: **Addition of appearance representations.** We add the appearance vectors of two samples (top rows of (a) and (b)), and obtain the sum-appearance in each bottom row. We inject motion pertained to each top appearance of (a) and (b) and are able to show same motion within lines of (a) and (b).

mary appearance pertaining to z_{a_0} and z_{a_1} , with identical motion as z_{m_0} and z_{m_1} , see Figure 11.

4.3.3 Ablation Study

We here seek to study the effectiveness of proposed G^3AN architecture, as well as the effectiveness related to each component in the proposed Generator. Towards this, we firstly generate videos by removing G_S and G_T , respectively, in order to verify their ability of controlling motion and appearance. We observe that when removing G_T , the model is able generate different subjects, however for each person the facial movement is constant, see Figure 12 (*top*). Similarly, when G_S is removed, changing motion will af-

fect subject’s identity, whereas the appearance vector loses its efficacy, see Figure 12 (*middle*). When removing both, G_T and G_S , appearance and motion are entangled and they affect each other, see Figure 12 (*bottom*). This demonstrates the effective disentanglement brought to the fore by the streams G_S and G_T .

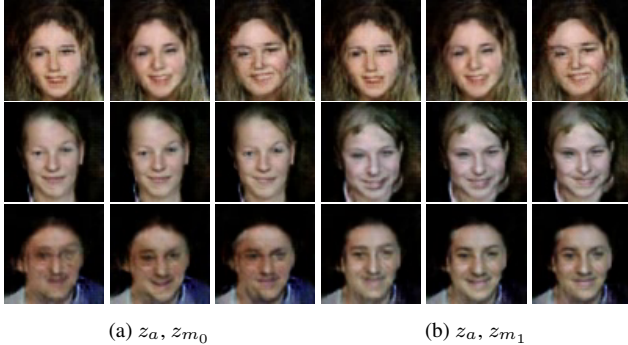


Figure 12: **Ablation study.** Generated videos obtained by removing G_T (*top row*), removing G_S (*middle*), and both (*bottom row*).

We proceed to demonstrate the contribution of G_S , G_T and F-SA in the Generator w.r.t. video quality. In this context, we remove each component individually and report results on the four datasets in Table 3. The results show that after removing all three components, video quality is the poorest, which proves that all of them contribute to the final results. Individually, G_S plays the most pertinent role, as removing it, decreases FID most profoundly for all datasets. This indicates that generating appearance features separately can be instrumental for good quality videos. Moreover, our results confirm the necessity of F-SA in our approach.

| Architecture | MUG FID ↓ | UvA FID ↓ | Weizmann FID ↓ | UCF101 FID ↓ | IS ↑ |
|----------------------|--------------|--------------|-------------------|-----------------|------|
| w/o $G_S, G_T, F-SA$ | 117.10 | 164.04 | 252.97 | 127.09 | 2.78 |
| w/o G_S, G_T | 113.44 | 159.54 | 176.73 | 120.17 | 3.16 |
| w/o G_S | 109.87 | 129.84 | 141.06 | 117.19 | 3.05 |
| w/o F-SA | 85.11 | 128.14 | 97.54 | 98.37 | 3.44 |
| w/o G_T | 82.07 | 121.87 | 94.64 | 96.47 | 3.16 |
| G^3AN | 67.12 | 119.22 | 86.01 | 91.21 | 3.62 |

Table 3: **Contribution of main components in G .**

Transposed Convolutions. Then, we compare the proposed factorized transposed spatio-temporal (1+2)D convolution, standard transposed 3D convolution, and transposed (2+1)D convolution, when used in G_V w.r.t. video quality. We carefully set the number of kernels, allowing for the three networks to have nearly same training parameters. We report the results of the quantitative evaluation in Table 4. Both convolution types, (1+2)D and (2+1)D outperform standard 3D kernels w.r.t. generated video quality. (1+2)D is slightly better than (2+1)D, and the reason might be that the (1+2)D kernel uses more 1×1 kernels to refine tempo-

ral information, which we believe to be important in video generation tasks.

| | MUG FID ↓ | UvA FID ↓ | Weizmann FID ↓ | UCF101 FID ↓ | IS ↑ |
|--------|--------------|---------------|-------------------|-----------------|-------------|
| 3D | 93.51 | 149.98 | 154.21 | 117.61 | 2.88 |
| (2+1)D | 73.08 | 141.35 | 95.01 | 98.70 | 3.36 |
| (1+2)D | 69.42 | 140.42 | 87.04 | 96.79 | 3.07 |

Table 4: **Comparison of various convolution types in G .**

Where to insert self-attention? Finally, we proceed to explore at which level of the Generator, F-SA is the most effective. We summarize performance rates in Table 5. Inserting F-SA after the G_3^3 module provides the best results, which indicates that *middle level feature maps* contribute predominantly to video quality. As shown in GAN Dissection [3], *mid-level features* represent semantic information, e.g., object parts while *high-level features* represent local pixel patterns, e.g., edges, light and colors and *low-level features* do not contain clear semantic information, which could be the reason, why F-SA achieves the best result in G_3^3 module.

| | MUG FID ↓ | UvA FID ↓ | Weizmann FID ↓ | UCF101 FID ↓ | IS ↑ |
|---------|--------------|---------------|-------------------|-----------------|-------------|
| G_0^3 | 83.01 | 188.60 | 96.38 | 100.37 | 3.09 |
| G_1^3 | 72.54 | 178.64 | 99.66 | 126.12 | 2.74 |
| G_2^3 | 69.02 | 160.12 | 97.53 | 112.36 | 3.03 |
| G_3^3 | 67.12 | 119.22 | 86.01 | 91.21 | 3.62 |

Table 5: **Comparison of inserting F-SA at different hierarchical levels of G^3AN .**

5. Conclusions

We have presented the novel video generative architecture G^3AN , which leverages among others on (i) a three-stream Generator that models appearance and motion in disentangled manner, as well as (ii) a novel spatio-temporal fusion method. We have performed an extensive evaluation of our approach on four datasets, outperforming quantitatively and qualitatively the state-of-the-art video generation methods VGAN, TGAN and MoCoGAN. Further, we have shown the ability of G^3AN to disentangle appearance and motion, and hence manipulate them individually.

Future work involves the design of GAN models for video generation of high resolution videos.

Acknowledgment. This work is supported by the French Government (National Research Agency, ANR), under Grant ANR-17-CE39-0002.

References

- [1] Niki Aifanti, Christos Papachristou, and Anastasios Delpoulos. The mug facial expression database. In *Work-*

- shop on Image analysis for multimedia interactive services (WIAMIS)*, pages 1–4. IEEE, 2010. 4
- [2] Badour AlBahar and Jia-Bin Huang. Guided image-to-image translation with bi-directional feature transformation. In *ICCV*, 2019. 1
- [3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *ICLR*, 2019. 8
- [4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828, 2013. 1, 2
- [5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 1
- [6] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*, 2017. 4
- [7] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, 2019. 1, 2
- [8] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016. 2
- [9] Hamdi Dibeklioglu, Albert Ali Salah, and Theo Gevers. Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *ECCV*, 2012. 4
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2
- [11] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *TPAMI*, 29(12):2247–2253, December 2007. 4
- [12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *CVPR*, 2018. 5
- [13] 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 *NIPS*. 2017. 5
- [14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017. 6
- [15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017. 1
- [16] Yunseok Jang, Gunhee Kim, and Yale Song. Video Prediction with Appearance and Motion Conditions. In *ICML*, 2018. 2
- [17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1
- [18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4
- [20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2
- [21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1
- [22] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 2
- [23] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *ECCV*, 2018. 1, 2
- [24] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *CVPR*, 2018. 1, 2
- [25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 1
- [26] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. *arXiv preprint arXiv:1903.04480*, 2019. 2
- [27] Yunchen Pu, Shuyang Dai, Zhe Gan, Weiyao Wang, Guoyin Wang, Yizhe Zhang, Ricardo Henao, and Lawrence Carin. Jointgan: Multi-domain joint distribution learning with generative adversarial nets. *arXiv preprint arXiv:1806.02978*, 2018. 1
- [28] Alec Radford, Luke Metz, and Soumith Chintala. Un-supervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4
- [29] Andrés Romero, Pablo Arbeláez, Luc Van Gool, and Radu Timofte. Smit: Stochastic multi-label image-to-image translation. In *ICCV Workshops*, 2019. 1
- [30] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017. 2, 4
- [31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training GANs. In *NIPS*. 2016. 5
- [32] Md Mahfuzur Rahman Siddiquee, Zongwei Zhou, Nima Tajbakhsh, Ruibin Feng, Michael B Gotway, Yoshua Bengio, and Jianming Liang. Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization. In *ICCV*, 2019. 1
- [33] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, 2019. 2
- [34] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild. Technical report, CRCV-TR-12-01, November 2012. 4
- [35] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018. 2

- [36] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 2
- [37] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 2
- [38] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017. 2
- [39] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 2, 5
- [40] Yaohui Wang, Piotr Bilinski, Francois F Bremond, and Antitza Dantcheva. ImaGINator: Conditional Spatio-Temporal GAN for Video Generation. In *WACV*, 2020. 3
- [41] 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*, 2018. 1
- [42] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *ECCV*, 2018. 1, 2
- [43] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. *arXiv preprint arXiv:1805.08318*, 2018. 3
- [44] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 1
- [45] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*, 2018. 2